Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Link scheduling algorithm for industrial wireless networks applied to factory automation ⋆

Gustavo P. Cainelli ∗ Max Feldman ∗ Gustavo Künzel ∗ Ivan Müller ∗ Carlos E. Pereira ∗ João C. Netto ∗

∗ *Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil (gustavo.cainelli@ufrgs.br, max.feldman@ufrgs.br, gustavo.kunzel@ufrgs.br, ivan.muller@ufrgs.br, cpereira@ufrgs.br, netto@inf.ufrgs.br)*

Abstract: Industrial Wireless Networks are an alternative to wired networks for process automation and factory automation. In this type of network, the network manager is responsible for creating and maintaining the network. One of the tasks of the network manager is the scheduling process. It is desirable that this process be carried out as fast as possible. In factory automation applications, this process becomes even more critical as cycle times are much shorter and the network topology changes more often compared to process automation applications. Therefore, one metric that should be considered when evaluating scheduling algorithms is the expected execution time under certain network conditions. This paper proposes a method that perform a pre-scheduling before the network start operating, in order to reduce the processing time. When new devices join to the network they receive pre-scheduled timeslots, making the search for available slots faster. We also compared the method proposed with scheduling algorithms applied in Industrial Wireless Networks and results show that the technique of pre-scheduling may be appropriate when it is necessary to execute the scheduling process in a fast manner.

Keywords: Industrial Wireless Networks, Links Scheduling, Multiple Superframes TSCH.

1. INTRODUCTION

Compared to wired technologies, Industrial Wireless Networks (IWN) have advantages that make them more suitable for Industrial Internet of Things (IIoT) and Industry 4.0 operations, such as flexibility and low installation cost, and facilitate communicating with robots and mobile devices. Typical IWN applications require deterministic, reliable, low latency communications, see Liu et al. (2019).

There are several scenarios for the application of IWN, with very diverse requirements. Some areas, such as Process Automation (PA), may differ substantially from others, such as Factory Automation (FA). PA focuses on the monitoring and control of chemical, biological or other processes in a plant, involving a wide variety of different sensors, which measure for example temperature, pressure and flow, and actuators, for example valves or heaters. Protocols like *Wireless*HART, ISA SP100.11a, and WIA-PA aim to meet PA requirements. These protocols have slow behavior during network formation and configuration, with times ranging from seconds to minutes, see Rauchhaupt and Meier (2013) and Zand et al. (2014).

Factory automation involves for example, assembly lines, robot motion control and mobile devices. Performance requirements vary with use cases. In general, FA applications have more stringent requirements in terms of low latency, reliability and determinism. For example, motion control of robotic devices may require cycle times less than 1 ms whereas in PA applications may tolerate longer cycle times since due to the nature of the variables commonly measured in PA. On the other hand, a network in FA, has fewer nodes (typically up to 100) compared to IWN applied in PA with potentially hundreds of distributed sensors, see Luvisotto et al. (2017). To address critical communication requirements required by FA applications, some solutions are found in the literature such as WISA/PNO WSAN, see Scheible et al. (2007), besides recent proposals like WIA-FA, WirelessHP and 5G, see Liang et al. (2019), Luvisotto et al. (2019) and 5G-ACIA (2018).

To achieve determinism in communications some protocols uses Time Division Multiple Access (TDMA) as medium access method where time is divided into timeslots. During a timeslot, a pair of devices communicates while the others remain idle to prevent collisions. The length of timeslots depends on the protocol used, where is common for PA timeslots of 10 ms (WH, ISA 100, WIA-PA) and FA timeslots less than 1 ms (WIA-FA, WirelessHP). Some protocols also use multiple channels to carry out communications such as IEEE 802.15.4 based protocols which divide the frequency spectrum into 15 channels. This technique is called Time Slotted Channel Hopping (TSCH) and operates based on TDMA with channel hopping to prevent communications from always happening on the same chan-

⋆ The authors thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) and the Federal Institute of Education, Science and Technology of Rio Grande do Sul - IFRS for financial support.

nel by decreasing the influence of external interference, see Luvisotto et al. (2019).

Typical centralized IWN are usually composed of the following elements: Network Manager (NM), Gateway, Access Point (AP), and Field Devices (FD). The NM is responsible for tasks such as provisioning new devices, routing, scheduling and optimizing, as well as adapting to dynamic network issues, see Künzel et al. (2019). The applications involve cyclic data exchange between a central controller (gateway) and a set of distributed sensors and actuators (field devices). To organize these communications, the NM must perform the scheduling process that consists of reserving network resources so that each device has enough time to carry out its transmissions. Field devices ask the NM for resources so that they can publish their data. The NM checks for available timeslots, and if so, allocates links to these timeslots so that the device can exchange data with the gateway. Data publishing rates depend on the application to which the device is inserted and may be in the range of seconds to minutes in PA and microseconds to seconds in FA, see Luvisotto et al. (2017).

The use of centralized management brings advantages such as simplification of field device hardware and firmware as the device does not have to perform management tasks, only communication, see Künzel et al. (2019). On the other hand, since the NM performs all the mentioned management tasks, it may happen that the management algorithms execution time significantly influences the decision making by the NM. The use of routing, scheduling and optimization algorithms that perform tasks quickly is desirable to allow IWN to meet the demands of FA applications and to reduce NM hardware complexity. The time it takes for the manager to execute its scheduling algorithms should be evaluated and minimized so that the manager can respond quickly to changes that occur during network operation, particularly in networks applied to FA.

Although link scheduling in IWN is a widely discussed subject in the literature, few studies discuss the execution time of the scheduling algorithms. In this context, the main contribution of this work is the proposal of a scheduling algorithm that aims to reduce the execution time of the scheduling process. The remainder of this article is organized as follows. Section II presents the literature review. Section III describes the system under consideration. Section IV presents the method proposed and the algorithms used in comparison, Section V shows the performance evaluation of the algorithms. Finally, Section VI presents the conclusions and future work.

2. LITERATURE REVIEW

Nobre et al. (2015) and Teles Hermeto et al. (2017) present a literature review on the main scheduling and routing solutions used for *Wireless*HART and also for IEEE 802.15.4 based protocolos. The approaches are evaluated by its objectives and some of the main metrics used in the evaluation of scheduling algorithms are link allocation success rate, average delay and superframe occupancy rate.

The Han et al. (2011) algorithm gives the scheduling priority for the fastest publish rate devices. The algorithm consists of two parts: the first determines which regions of the superframes will be allocated the links of a device and the second is responsible for finding a timeslot within the given region for the link. This strategy performs the scheduling FD by FD, that is, all links belonging to a FD i are allocated and only then allocate the links of FD $i+1$. The technique assigns four links to each device. The S_i superframe for a given flow Fi is divided into four parts, where each part will have a link. The length of superframe S_i is P_i, where P_i is the period which device update its variable and send it to the gateway. The algorithm looks for a single available timeslot for each link in the flow. For the first link allocation, the algorithm searches for a available timeslot in range 0 to $P_i/4$, for the second link in the range of $P_i/4$ to $P_i/2$ and so on. This technique uses multiple data superframes, that is, each publication period has its own superframe but it is not concerned with the execution time of the algorithms, which can be high for the application in factory automation if the number of devices is large.

Min et al. (2016) proposes a centralized scheduling algorithm with support for multiple superframes. The work presents two main procedures: Calculate the set of *timeslots* available and allocate the communication according to the data traffic requirement. The authors compare single superframe with multiple superframes and conclude that there is a reduction in energy consumption with the use of multiple superframes which consequently, increases the operating time of the field device. However, this work does not evaluate the execution time of the algorithms.

In Modekurthy et al. (2019) are presented decentralized scheduling techniques. The essence of these strategies is to adopt local scheduling (at the device level), allowing each device to schedule its transmissions using a local, real-time scheduling policy. This method avoids the need to create and disseminate a global scale that, according to the authors, reduces significantly the use of resources. The authors show that, compared to centralized strategies, there are gains in execution of the algorithms. Here they evaluate the execution time of scheduling algorithm but they use a decentralized technique and does not use multiple superframes.

One of the gaps found in the reviewed articles is that, in general, centralized scheduling algorithms with support for multiple superframes are not primarily aimed at reducing execution time, which is a relevant metric for factory automation applications.

3. SYSTEM DESCRIPTION

3.1 Network Manager Tasks

The NM is responsible for several tasks, like network formation, maintenance and optimization. The tasks can be described in 6 main steps. The moment the NM starts operating, communication is established with the gateway and provisioning of the AP (Step 1) occurs. Once initial configurations are done, the AP begins publishing advertise packets for FDs to join the network (Step 2). During the FD join process, the routing algorithms build graphs (Step 3). Graphs are constructed taking into account the network topology. For star topology networks where path redundancy does not exist, the routing process is not be required. Then the scheduling algorithms allocate links in timeslots (Step 4) taking into account application requirements and rules. To avoid sending redundant settings, the new settings generated are compared with the old ones, if exists, (Step 5). Finally the settings are then converted to a sequence of commands and sent to the devices (Step 6). To perform the reconfiguration, new links must be sent to devices and only then the old links will be removed, see Zand et al. (2014) and Künzel et al. (2019). The NM monitors the network frequently and if any topology changes occur, i.e. a FD joins network, the routing and scheduling algorithms are re-executed.

3.2 Scheduling process

As the objective of this work is to evaluate the scheduling step, it was considered a star topology, in order to simplify the application, since it does not require routing techniques.

This system can be modeled as a graph $G = (V, E)$ where nodes $v \in V$ represent FDs, E is the set of edges that represent the l_{ij} communication links between the FDs and the AP. Fig. 1 presents i field devices connected to the AP.

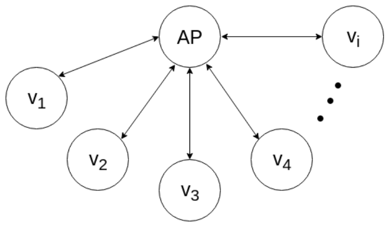

Figure 1. Star topology with i FDs conected to AP.

Each of the end-to-end deliveries between the gateway and the devices is defined as a $F_i \in F$ flow, where $F = F_0, F_1, ..., F_n$. Each flow F_i periodically generates a packet with period P_i. Flows follow harmonics periods according to the expression a^q, where a is a constant numbers and q is any natural number, where all periods can be divided together. Examples of harmonic periods are $1, 2, 4, 8, 16$ and $3, 6, 12, 24$. For the set F, T is the hyper-period which is the least common multiple between the flow periods, see Nobre et al. (2015).

A collection of consecutive timeslots is a superframe S_i which repeats cyclically. A network can contain multiple superframes of various sizes arranged in parallel in time. Multiple superframes can be used to determine how different device groups will communicate. These groups can be defined by the P_i period of data publication. The NM generates a superframe for every P_i present in the network. The l_{ij} links from nodes with period P_i must be allocated to S_i. In addition to the superframes the NM stores a M scheduling matrix where the columns refer to timeslots and the rows refer to channel offset. The M matrix size is $N_{channels}$ x T.

The $|S|$ superframes generated must coexist on the network without causing conflicts. That is, a link used in one superframe should not conflict with a link from another superframe. For this, all superframes must be organized within the M matrix so that they do not occupy the same timeslots. The superframes are multiples of each other and thus repeat within the M matrix, for example if the longest publishing period of network devices is 16 timeslots, then M has 16 timeslots and an 8 timeslots superframe repeats itself 2 times within M. Fig. 2 presents an example where the devices v_1, v_2 and v_3 from Fig. 1 periodically publish their data every 4, 8 and 16 timeslots respectively. The figure shows the superframes for each of the periods mentioned in the example. In each of the superframes an uplink link is allocated. Note that the superframes share the M scaling matrix and that S_1 and S_2 repeat more than once within M. Once the superframes are cyclically repeated it is sufficient to find a schedule for packet transmissions generated up to the T timeslot (in this example $T = 16$).

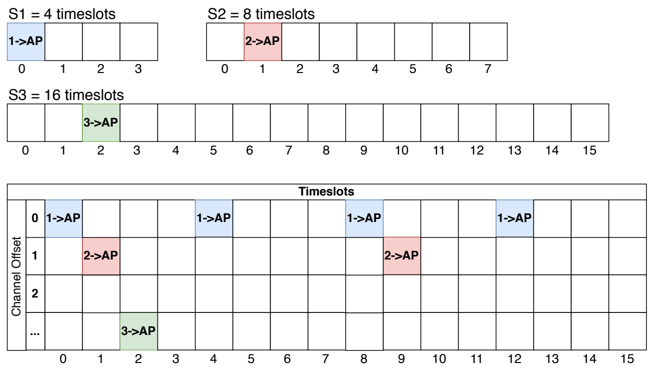

Figure 2. Scheduling matrix M.

For simplicity, in this work it was determined that 4 links will be allocated for each device: transmission and retransmission of uplink (from device to gateway) and downlink (from gateway to device). In a real application, different amounts of uplink and downlink links could be distributed between sensors and actuators. For example, an actuator does not necessarily need to publish data (uplink) periodically as often as it receives from the controller (downlink), so the number of uplink links could differ from downlink. In addition, frequency reuse is not considered, meaning that two transmissions cannot occur on the same timeslot, even on different channels.

4. SCHEDULING ALGORITHMS

To perform a comparison and analysis of the execution times of the scheduling algorithms, different link scheduling techniques were implemented, as well as a proposed technique. The algorithms implemented are Deadline Monotonic (DM), Earliest Deadline First (EDF), Han and an adaptation of the Han algorithm using concept of macro operations called Han-MO that consists of allocating links simultaneously. The difference from the original Han algorithm and Han-MO is that adapted method consists of allocating all links from flows of a v_i device simultaneously to the M matrix reducing the number of iterations of the algorithm.

4.1 Scheduling Algorithm Proposal

The technique called Link Scheduling Using Pre-Allocated structures (LSPA) consists of two phases: the first, nammed Pre-Scheduling is executed in the moment the NM starts operating, the second, called scheduling is performed during network operation and uses information generated in the first phase.

Pre-Scheduling: This process takes into account the number of links that will be allocated to each device and also the publishing periods supported by the network. The NM generates a list of structures that represent the timeslots where devices will be allocated. These structures consist of a group of k timeslots, where k is the amount links that will be allocated to each device (in this work $k = 4$). Since the NM is aware of possible device publishing periods and also the number of links that will be allocated to each one, it is possible to predict the number of devices that can be allocated. For example, if all devices publish their data within 8 timeslots, 2 devices can be allocated since each device will have 4 timeslots for exclusive use. This way the NM can predict what are the timeslots in which the links of a given device will be allocated. Fig. 3 exemplifies this case. The four links of each device are equally spaced within the superframe. The numbers indicate which device is occupying each timeslot. Arrows indicate whether the link is from uplink (↑) or downlink (↓).

1↑	2↑	1↑	2↑	1↓	2↓	1↓	2↓
0	1	2	3	4	5	6	7

Figure 3. FDs 1 and 2 scheduled in superframe S_i with size 8.

Although it is possible to predict which slots a device of a certain period will use, it is not known how many devices of each period will enter the network and the order in which they will enter, so it is not possible to pre-schedule the links of the devices. Thus, the pre-allocation of structures is a way to reserve timeslots without necessarily knowing which links will occupy them. For each publishing period that the network suports, a Ci set of structures is generated. Each Ci set has $\frac{P_i}{4}$ structures called Ci_n where n are defined by (1).

$$0 < n < \left(\frac{P_i}{4}\right) - 1; \qquad (1)$$

Each structure Ci_n has j timeslots where $0 \leq j \geq 3$ calculated by (2).

$$timeslots\ of\ Ci_n = j * \frac{P_i}{4} + n \qquad (2)$$

For example, Figure 4 presents the pre-allocation of structures for periods of 8 and 16 timeslots. The NM generates these array in the pre-scheduling process (Step 1). Instead of assigning links to timeslots, at first the indexes of each structure are assigned. This information is used in the scheduling process (Step 4).

Some of the structures have conflicting timeslots, so at the moment that one structure is used, all the others that conflict with it can not be used.

$P_i = 8$

$C8_0$	$C8_1$	$C8_0$	$C8_1$	$C8_0$	$C8_1$	$C8_0$	$C8_1$
0	1	2	3	4	5	6	7

$P_i = 16$

$C16_0$	$C16_1$	$C16_2$	$C16_3$	$C16_0$	$C16_1$	$C16_2$	$C16_3$	$C16_0$	$C16_1$	$C16_2$	$C16_3$	$C16_0$	$C16_1$	$C16_2$	$C16_3$
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 4. Pre-Scheduling of structures of set $C8$ and $C16$.

Scheduling: The scheduling process consists of searching for pre-allocated structures. For each v device the algorithm searches for an available Ci_n structure that meets the P_i publishing period of the device and do not conflit with an used structure $u \in U$, where U is the set of used structures. Algorithm 1 presents how this process works:

Algorithm 1: LSPA

```
    Input: V//Field devices set    C//Structures set
    Output: M//Scheduling matrix
1   foreach v ∈ V do
2       i = period of v;
3       foreach Ci_n ∈ C_i do
4           if Ci_n == unavailable then
5               continue;
6           end
7           //U used structure set.
8           else if Ci_n conflict with u ∈ U then
9               Ci_n = unavailable;
10              continue;
11          end
12          //Available structure found.
13          Allocate the v links in M using the Ci_n
             slots.
14          Ci_n = unavailable;
15      end
16  end
```

By the time the algorithm finds the structure available for the given period (line 10) it is already possible to know the four timeslots to which the device links will be allocated. This way the allocation of the links in the matrix M is made using the slots of the found structure. From then on, the structure becomes unavailable.

5. PERFORMANCE EVALUATION

A simulation environment in C language was developed to evaluate the approach in two general study cases. Case A is a complete network rescheduling and Case B is scheduling of a device that is joining an already formed network. The step evaluated in this study was Step 4 of section 3.1.

5.1 Metrics

The metrics used for performance analysis were:

(1) Runtime: is the total time required to generate the scheduling. 3.1.
(2) Success Rate: The number of times the algorithm has succeeded in scheduling a device that tries to join the network over the total number of experiments. The moment a device attempts to join an already formed

network, the scheduling algorithm succeeds if it can find a feasible scheduling.

5.2 Case of study

The experiments performed seek to represent possible cases of IWN applications.

Case A: In the first case it was considered the complete rescheduling of the network where the NM runs the scheduling process for all devices on the network. There are two different publishing periods between devices. Six groups were evaluated (A-F). The total number of devices ranged from 10 to 150. table 1 presents the parameters of this scenario.

Table 1. Case A parameters

	Number of devices					
P_i(slots)	A	B	C	D	E	F
400	5	10	20	30	40	50
800	5	15	30	45	60	100
Total	10	25	50	75	100	150

Case B: The Case B refers to the second case, which is the scheduling of a device that is entering a network already formed. For this scenario, the scheduling of a device group was initially generated. The chosen group has 30 devices with publish rate of 200 timeslots, 30 with publish rate of 400 and 20 with publish rate of 800. Then, the occupancy percentage of the scheduling matrix is chosen. From the chosen occupation, links from the scheduling matrix are randomly deleted until the desired occupation is reached. After obtaining the scheduling matrix with the desired occupancy, the Han, Han-MO and LSPA algorithms are executed to perform the scheduling of a device v that has a publish period of 200 timeslots. In this scenario it is analyzed if the algorithms can scale the device in the partially occupied scheduling matrix. It is also evaluated the execution time of each algorithm.

5.3 Results

Below are results for the proposed scenarios. The tests were performed on a 2.4 GHz Intel Core 2 Duo processor machine with Ubuntu 18.04 operating system.

Case A: The Fig. 5 presents the result of the experiment performed. In this scenario EDF presented the slowest execution time among the algorithms. Han's algorithm had a runtime longer than DM for 150 devices. The Han-MO method had shorter execution times than the last three ones. In this scenario the LSPA method outperformed the other algorithms and for 150 devices the execution time was 40% of the time spent by Han-MO.

Case B: The Fig. 6 presents the result of the experiment performed. The bandwidth was varied from 25% to 99% and evaluated whether the scheduling algorithm was able to schedule the device that wants to be part of the network. The figure shows that for 25% and 50% of occupancy of the scheduling matrix the three evaluated algorithms were able in all repetitions to scale the device. Starting at 95% Han-MO algorithms and LSPA begin to decrease success rate. Han's algorithm gets the best results

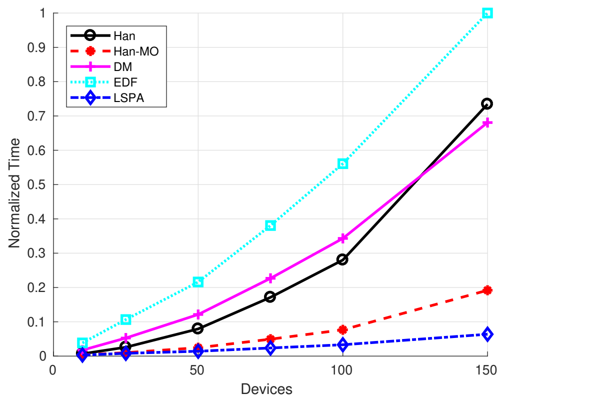

Figure 5. Execution time of scheduling algorithms considering two different publishing periods between devices.

in this analysis, and this is because the search for timeslots is done individually for each link making scaling more flexible. Han-MO and LSPA algorithms maintain a fixed distance between links at the time of scheduling, making it less likely to find a feasible scheduling. Although it is a strategy that makes the scheduling process faster, it decreases the success rate in scheduling because the algorithms are less flexible than Han. The DM and EDF algorithms were not compared in this scenario as they consider priority between links to allocate, so they are not suitable for scheduling a single device in an already formed network.

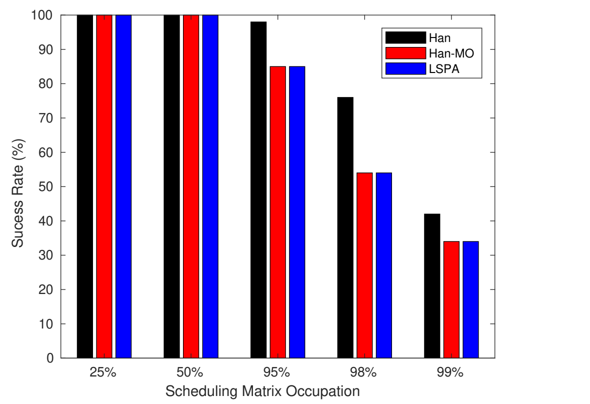

Figure 6. Algorithm success rate in 100 different scenarios for each occupancy of matrix M.

The Fig. 7 presents the results of the algorithm executions for the different band occupations. This time the distributions of the results are presented because each execution of the algorithms is performed considering a different scheduling matrix configuration, for example, in 25% bandwidth tests are performed with 100 different scheduling matrix configurations. It can be observed that the HAN-MO algorithms and the remained with low dispersion, with their values close to the average. Han's algorithm, despite having a higher success rate, had the worst results for cases where the scaling matrix is almost complete, but had a high dispersion of values for cases of 95%, 98% and 99%. This is because the Han algorithm performs link allocation individually and the search for each link is performed in a specific area of the superframe. Thus, for scheduling matrix configurations where available slots are in the first positions the algorithms will need a few iterations to find a scheduling for the device. However, if the available slots are far from the first positions, the algorithm will need several iterations to find a scheduling, these repetitions are multiplied by the number of links.

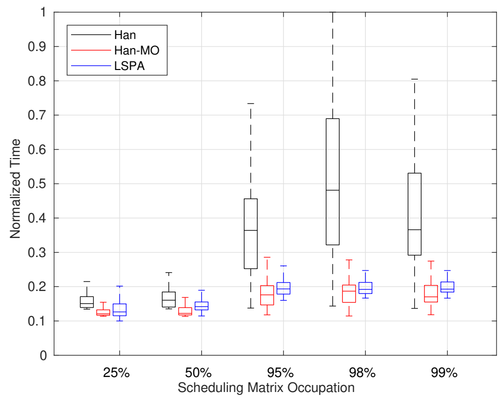

Figure 7. Algorithm execution time for scaling a device with different bandwidth rates.

6. CONCLUSION

This paper presented an centralized algorithm with multiple superframe suport to reduce the time required to perform the scheduling process. LSPA, use a pre-scheduling technique to reserve timeslots to devices. A comparison between existed link scheduling algorithms was made. The results show that LSPA can generate scheduling in less time than the compared algorithms, being a good alternative to the scheduling algorithm in an IWN applied to the FA, where the manager needs to respond quickly to changes in the network. In the experiments, DM and EDF techniques have the worst results. This is because each iteration of the algorithm the list of links is sorted according to the priority of each one. Han and Han-MO algorithms allocate all links from a set of device. However, Han searches for available timeslots for each link individually and Han-MO searches for all links simultaneously, making the search faster. The LSPA does not search for available timeslots like the others, but instead searches for a structure of available timeslots.

In factory automation applications where timeslots are less than 1ms the processing time of these steps can represent several cycles of the superframes causing a slow manager response to network changes. If optimization algorithms are applied to IWN, successive executions of routing and scheduling algorithms may occur. Successive execution of the scheduling algorithms can make management a slow process that may eventually not meet the high dynamic requirements of an network applied to FA.

Future works are intended to improve the pre-scheduling technique to meet more complex topologies such as tree and mesh. Pre-scheduling adaptation in these types of topology can make room for frequency reuse, where two or more transmissions could occur at the same timeslot on different channels. Moreover, practical tests are necessary to validate the proposed technique in a real IWN.

REFERENCES

5G-ACIA (2018). 5G for Connected Industries and Automation. Technical report, ZVEI “German Electrical and Electronic Manufacturers" Association. URL `www.5g-acia.org`. Access in: 15 Jul 2019.

Han, S., Zhu, X., Mok, A.K., Chen, D., and Nixon, M. (2011). Reliable and real-time communication in industrial wireless mesh networks. In *17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 3–12. IEEE Computer Society, Massachusetts (EUA).

Künzel, G., Indrusiak, L., and Pereira, C.E. (2019). Latency and lifetime enhancements in iwsn: a q-learning approach for graph routing. *IEEE Transactions on Industrial Informatics*.

Liang, W., Zheng, M., Zhang, J., Shi, H., Yu, H., Yang, Y., Liu, S., Yang, W., and Zhao, X. (2019). WIA-FA and its applications to digital factory: A wireless network solution for factory automation. *Proceedings of the IEEE*, 107(6), 1–21. doi:10.1109/JPROC.2019.2897627.

Liu, Y., Kashef, M., Lee, K.B., Benmohamed, L., and Candell, R. (2019). Wireless Network Design for Emerging IIoT Applications: Reference Framework and Use Cases. *Proceedings of the IEEE*, 107, 1166–1192.

Luvisotto, M., Pang, Z., and Dzung, D. (2017). Ultra High Performance Wireless Control for Critical Applications: Challenges and Directions. *IEEE Transactions on Industrial Informatics*, 13, 1448–1459.

Luvisotto, M., Pang, Z., and Dzung, D. (2019). High-Performance Wireless Networks for Industrial Control Applications: New Targets and Feasibility. *Proceedings of the IEEE*, 107, 1074–1093.

Min, M., Yang, Z., Zhang, Y., Wang, Y., and Li, Z. (2016). Traffic aware multiple slotframes scheduling algorithm in industrial IoT applications using IEEE802.15.4e TSCH. *International Conference on Communication Technology Proceedings, ICCT*, 608–614.

Modekurthy, V.P., Saifullah, A., and Madria, S. (2019). Distributedhart: A distributed real-time scheduling system for wirelesshart networks. In *25 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 216–227. IEEE, Montreal (CanadÃ¡).

Nobre, M., Silva, I., and Guedes, L.A. (2015). Routing and scheduling algorithms for wirelesshart networks: A survey. *Sensors (Switzerland)*, 15, 9703–9740.

Rauchhaupt, L. and Meier, U. (2013). Performance classes for industrial wireless application profiles and its determination. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 1–8.

Scheible, G., Dzung, D., Endresen, J., and Frey, J.E. (2007). Unplugged but connected - Design and implementation of a truly wireless real-time sensor/actuator interface. *IEEE Industrial Electronics Magazine*, 1, 25–34.

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.

Zand, P., Mathews, E., Havinga, P., Stojanovski, S., Sisinni, E., and Ferrari, P. (2014). Implementation of wirelesshart in the ns-2 simulator and validation of its correctness. *Sensors (Basel, Switzerland)*, 14, 8633–68.