



# Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey

Rodrigo Teles Hermeto, Antoine Gallais, Fabrice Theoleyre

## ► To cite this version:

Rodrigo Teles Hermeto, Antoine Gallais, Fabrice Theoleyre. Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey. Computer Communications, 2017, 114, pp.84-105. 10.1016/j.comcom.2017.10.004 . hal-02565958

**HAL Id: hal-02565958**

**<https://hal.science/hal-02565958>**

Submitted on 6 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey

Rodrigo Teles Hermeto<sup>a,\*</sup>, Antoine Gallais<sup>a</sup>, Fabrice Theoleyre<sup>a,\*</sup>

<sup>a</sup>*ICube Laboratory, CNRS / University of Strasbourg, Boulevard Sebastien Brant, 67412 Illkirch Cedex, France*

---

## Abstract

The so-called Industrial Internet of Things (IIoT) is expected to transform our world, and in depth modernize very different domains such as manufacturing, energy, agriculture, construction industry, and other industrial sectors. The need for low power radio networks first led to low duty cycle approaches where nodes turn off their radio chipset most of the time to save energy. The medium access control (MAC) has thus been largely investigated over the last fifteen years. Unfortunately, classical contention access methods use a random access and are unable to provide guarantees. In the meantime, some dedicated standards have emerged (e.g. IEEE 802.15.4-2006, IEEE 802.15.4-2015), combining Time Division Multiple Access (TDMA) with slow channel hopping in order to enable reliability and energy efficiency. Slow channel hopping allows each node to use different channels for a frame and its possible retransmissions with a low-cost hardware. To provide high-reliability, these protocols rely on a common schedule in order to prevent simultaneously interfering transmissions. In this context, we clearly observe a strong growth of the number of proposals in the last years, denoting a strong interest of the research community for deterministic slow channel hopping scheduling for the IIoT. We categorize here the numerous existing solutions according to their objectives (e.g. high-reliability, mobility support) and approaches. We also identify some open challenges, expected to attract much attention over the next few years.

**Keywords:** Industrial Internet of Things (IIoT); Wireless networks; Scheduled Medium Access Control (MAC); Slow Channel Hopping; Dedicated Transmission Opportunities; Deterministic MAC

---

## 1. Introduction

The wide adoption of low-power embedded devices has attracted much attention to the Internet of Things, which enables the interconnection of all these small smart objects to the Internet. While Cisco has predicted 50 billions small IoT devices by 2020, Huawei even expects the deployment of 100 billions by 2025. Very high densities of smart objects expect to help the smart cities to liquefy the vehicular traffic, save energy, reduce the pollution, etc. [1].

While the early adopters bought small best-effort devices for leisure, modern applications are more and more demanding in terms of responsive communications with high reliability. In particular, radio

---

\*Corresponding author

Email addresses: [teleshermeto@unistra.fr](mailto:teleshermeto@unistra.fr) (Rodrigo Teles Hermeto), [gallais@unistra.fr](mailto:gallais@unistra.fr) (Antoine Gallais), [theoleyre@unistra.fr](mailto:theoleyre@unistra.fr) (Fabrice Theoleyre)

transmissions tend to replace reliable wired connections. However, sensors and actuators have now to interact in real-time: a reliable delivery of the command packets before a given deadline is thus expected.

We expect a very large adoption of the Industrial Internet of Things (IIoT), in various key areas. Smart agriculture would exploit a radio infrastructure to monitor in real-life a greenhouse, or a field [2]. Industry 4.0 expects to leverage on the Internet of Things to make the industrial chain more malleable [3]. It aims to transform all the devices for the supply and manufacturing chain into autonomous radio embedded devices. Smart buildings expect to incorporate many radio sensors and actuator for home automation, in particular for energy savings [4]. The ETSI has detailed the requirements (delay, reliability, volume of traffic) for different applications in Smart Cities [5]. To this purpose, real-time systems are needed, encompassing the operating system, the application and the communication protocols. End-to-end performance and guarantees are the alpha and the omega.

Networking stacks for low power radio networks try to implement low duty cycle approaches: a node has to turn off its radio chipset most of the time to save energy. The medium access control (MAC) is in charge of deciding when a node is authorized to transmit to avoid both deafness (the receiver is not awake) and collisions (two transmissions overlap). S-MAC [6] was one of the first algorithm to schedule transmissions at the link layer. Neighboring devices have to send their expected wake-up times to be able to exchange packets. Later, preamble sampling approaches proposed to reduce the overhead by forcing the transmitters to announce, via a *preamble*, their upcoming transmissions [7]. When a node detects a preamble, it must stay awake to receive the upcoming frame. However, these approaches suffer from the hidden terminal problem, and rely on classical contention access methods, thus being unable to provide strict guarantees for medium access.

The industry has then tried to push interoperability by creating standards. IEEE 802.15.4-2006 [8] proposed a time-synchronized approach, mixing random access during the first part, and dedicated timeslots for real time transmissions during the last part. Unfortunately, bursts of collisions keep on occurring during the random access part, and impact very negatively both latency and reliability [9]. More recently, the amendments of this standard [10] focused on improving the reliability and the energy efficiency. These approaches rely for most of them on Time Division Multiple Access (TDMA) paired with a synchronization mechanism.

Besides, slow channel hopping has been proved to combat efficiently narrow band noise, very common in industrial environments [11]. A frame is transmitted through a single physical channel, but the other packets (and the possible retransmissions) use a different channel, following a pseudo-random sequence. While these MAC protocols focus on high reliability and low latency, they all impose to schedule the transmissions. Interfering transmitters should not be allowed to transmit simultaneously, to eliminate collisions. Similarly, the schedule has to be properly computed in order to optimize the end-to-end latency, a key performance indicator for most industrial networks [12].

In this article, we present an overview of the different existing approaches to construct an accurate schedule. We investigate the various constraints that must be considered when scheduling transmission opportunities in a low-power industrial wireless network. We focus in this paper on the solutions dedicated to the Frequency-Time Division Multiple Access (FTDMA) approaches. More precisely, we consider the industrial networks where a time-frequency schedule has to be installed, so that different transmitters receive a dedicated access, without creating collisions. We envision increasingly complex scenarios of utilization in IIoT, and categorize the existing algorithms depending on their key characteristics. We also describe for each of them the features they offer (e.g. high reliability, mobility support, delay minimization).

Figure 1 references the scheduling algorithms that we detail hereafter. We identified for each reference its date, and the technology it relies on. First scheduling algorithms targeted the organization of the dedicated timeslots of IEEE 802.15.4-2006 [8], which has stopped attracting attention for a few years. Inversely, most of recent contributions are based on the IEEE 802.15.4-2015 amendment [10], with a supremacy of the TSCH in the last 2 years. Besides, we clearly observe a strong growth of the number of contributions in the last years, denoting a strong interest of the research community for deterministic and schedule-based slow channel hopping scheduling for the IIoT.

This survey aims to provide a comprehensive view of the different scheduling approaches for slow channel hopping, time-division based MAC, whatever the actual standard is. We present first a classification of the

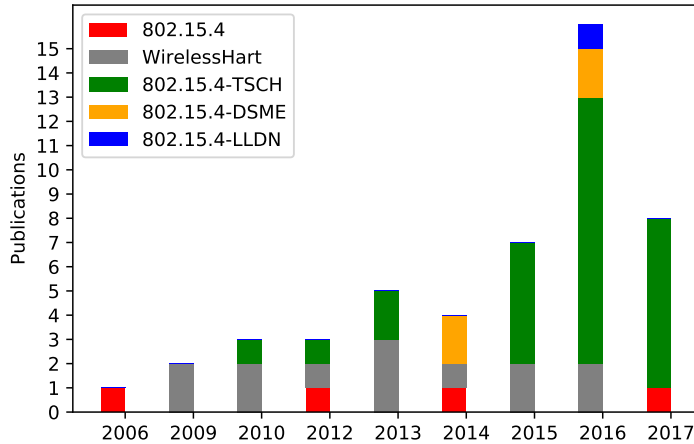


Figure 1: Number of papers and its target technologies over the years.

different proposals, and then detail each family, to discuss also about their limitations. This surveys aims to serve as a central point in the large collection of scheduling algorithms existing in the literature.

Guglielmo *et al.* [13] presented a very accurate survey, describing in depth the main IEEE 802.15.4e mechanisms. In particular, they detail the different novel modes (TSCH, LLDN and DSME) proposed in this amendment. While this survey addresses also the scheduling problem in TSCH, it represents only one of the many aspects addressed in the paper. To our mind, the present survey is complementary: we aim to focus uniquely on the scheduling problem in TSCH, detailing the different families of approaches, their limits, and pointing the open challenges which remain to be tackled in this research area.

This survey presents the following contributions:

1. we present the existing standards and protocols which rely on slow channel hopping and time-division for their transmissions;
2. we detail the requirements of a scheduling algorithm used for this kind of MAC protocol;
3. we provide a comprehensive view of the existing scheduling algorithms. For each of them, we specify their ideal cases of utilization, and their respective limits;
4. we finally identify some remaining open challenges in this research field.

The paper is organized as follows. Section 2 describes the fundamental mechanisms of the IEEE 802.15.4-TSCH standard since it corresponds to the most recent and now used standard for wireless low power industrial networks. In Section 3, we expose the required preliminaries to understand the ins and outs of scheduling in slow channel hopping MAC for low power industrial wireless networks. We then expose the characteristics of a scheduling algorithm for slow channel hopping in Section 4 where we also propose a taxonomy of the existing approaches. We then detail existing solutions for centralized schedules relying on e.g., a Path Computation Element (Section 5). Then, we detail the distributed algorithms, able to schedule the transmissions in each node, while minimizing the collisions in the union of the local schedules (Section 6). We also present mobility aware scheduling techniques (Section 7), which require specific approaches to deal with highly dynamic topologies. Finally, we identify some open challenges that remain in the field (Section 8.1), and give some concluding remarks in Section 8.

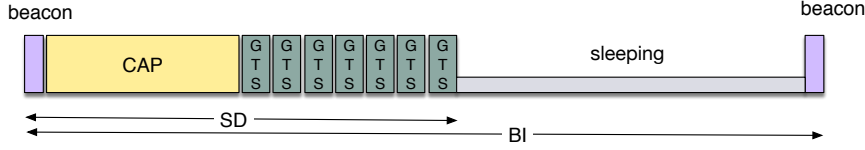


Figure 2: Superframe of IEEE 802.15.4-2006.

## 2. Mechanisms of the IEEE 802.15.4 TimeSlotted Channel Hopping (TSCH) mode

The Industrial Internet of Things requires to respect strict guarantees. A best-effort solution is not anymore acceptable: high reliability must be ensured (e.g.  $> 99\%$ ), as well as a low latency, with an upper bound. The Medium Access Control represents a key element in the networking stack to achieve such objective. It needs to share the radio bandwidth among the different contenders while minimizing the number of collisions and limiting the transmission delay.

Random Access is particularly efficient to deal with a variable traffic and number of transmitters. In particular, random techniques such as Slotted Aloha or CSMA provide a very low latency with negligible packet losses when the traffic is very low in regards to the capacity [14]. However, fairness, reliability and reactivity decrease very significantly as soon as the load exceeds a threshold. Thus, pure random access methods seem inaccurate for the industrial applications.

The IEEE802.15.4-2006 standard [8] mixes a Time Division Multiple Access (TDMA) and a random access to organize the transmissions. In the beacon-enabled mode, a coordinator maintains its neighbor synchronized by transmitting periodically a **beacon** (Fig. 2). After the reception of the beacon, any node can transmit a frame, adopting a random access approach until the end of the Contention Access Period (CAP). Only a few transmission opportunities may be reserved for a contention-free access – the Guaranteed TimeSlots. Finally, all the nodes can switch their radio off to save energy.

To cope efficiently with multihop networks, the beacons of the sleeping periods have to be carefully scheduled [15]. Some extensions propose to use a multichannel approach, where each coordinator receives a portion of time and a physical channel for its superframe [16]. However, the standard keeps on suffering from a burst of collisions at the beginning of the superframe [9], impacting very negatively the reliability. This standard is thus inaccurate for wireless industrial networks, but has given birth to the TimeSlotted Channel Hopping (TSCH) mode, much more relevant to cope with these situations.

IEEE 802.15.4-2015 has proposed the TSCH mode for industrial wireless sensor networks [10]. It is mainly built upon the previous WirelessHART [17] and ISA100.11a [18] standards, and the Time Synchronized Mesh Protocol [19]. For instance, Wireless HART and the TSCH mode differ only in their packet format. Since most of the concepts are very closely related, we will uniquely concentrate our explanations on the IEEE802.15.4-TSCH standard.

### 2.1. Organization of the transmissions

IEEE 802.15.4-TSCH relies on a strict organization of the transmissions to make the protocol deterministic, and to remove most of the collisions. TSCH relies on a slotframe, comprising a fixed number of timeslots (i.e. the slotframe length). This slotframe is cyclic and can be represented with a scheduling matrix (Fig. 3). A cell is defined by a pair of timeslot and channel offset denoting respectively its instant of transmission in the slotframe and the frequency it has to use for the transmission. A set of cells is allocated to each active link which has to forward a certain quantity of traffic. The slotframe is repetitive: the same cell is allocated in each slotframe to the same set of radio links.

Depending on the traffic forwarded through the link, two different types of cells are defined in TSCH:

- a **dedicated** cell is contention-free. A transmitter has the right to send its packets without contention in all its dedicated cells. Thus, the same dedicated cell should not be allocated to a set of interfering transmitters.

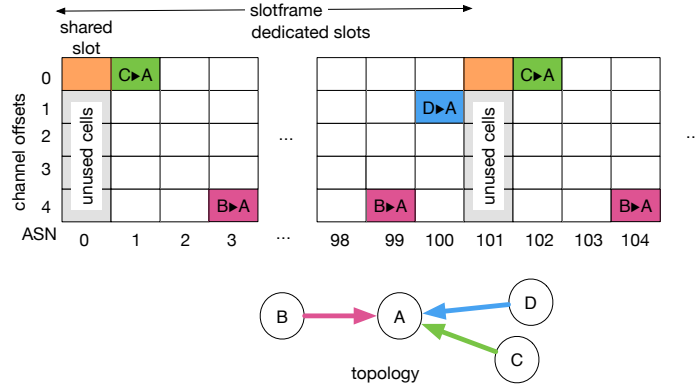


Figure 3: Structure of the slotframe in IEEE802.15.4-TSCH.

a **shared** cell implements a slotted Aloha approach. TSCH adopts an optimistic strategy: the packet is transmitted without contention for its first transmission. If the packet has to be acknowledged while nothing is received by the transmitter, a collision is assumed to have taken place. In this case, the transmitter selects randomly a backoff value between 0 and  $2^{BE} - 1$ , BE denoting the Backoff Exponent [10]. Then, the transmitter must defer the retransmission during this random number of shared cells (in the same or in different slotframes).

The Absolute Sequence Number (ASN) denotes the number of timeslots since the network started. We have consequently a time reference, global to all the nodes.

To support efficiently broadcast transmissions, all the nodes have to stay awake during a given cell. In that case, the same shared cell should be allocated to all the nodes, typically during the first timeslot of the slotframe. In that case, such shared cells are used for control packets when a node joins the network.

By allocating real-time traffic to dedicated cells, IEEE 802.15.4 2015-TSCH is consequently able to provide a deterministic medium access. Only the control (less sensitive) packets may suffer from collisions.

## 2.2. Slow Channel Hopping for High-Reliability

Slow channel hopping techniques have been proved to perform very well in industrial environments [11]. A frame is transmitted using a single channel, with a low-cost radio interface. However, the subsequent transmissions follow a pseudo-random sequence: when a transmission fails, its retransmission will use another physical channel, decreasing the probability to create another collision. Thus, a slow channel hopping technique helps to improve the reliability in presence of external interference. and to combat fading in indoor environments [20].

As explained previously, a cell is defined by a pair of timeslot and channel offset. To support slow channel hopping, a node derives the actual frequency to use at the beginning of the timeslot with the following equation:

$$channel = F[(chOffset + ASN) \bmod NbChannels] \quad (1)$$

where  $chOffset$  denotes the channel offset,  $ASN$  counts the number of timeslots since the network started,  $NbChannels$  the number of physical channels (by default 16 in TSCH), and  $\bmod$  the modulo operator.  $F[]$  is a bijective function mapping an integer comprised between 0 and 15 into a physical channel.

If the slotframe length and the number of physical channels are mutually prime, this means that the same cell will be mapped to a different physical frequency in different slotframes. Using the whole diversity of the radio spectrum helps to improve the reliability.

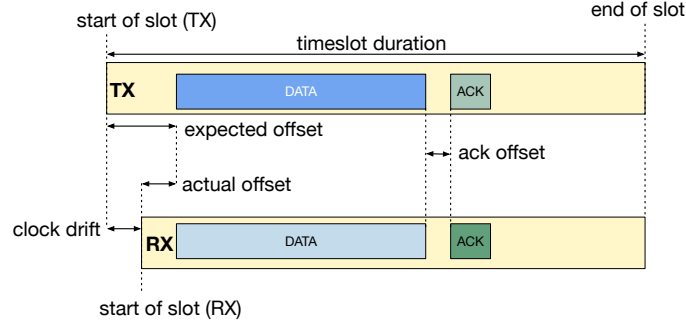


Figure 4: Resynchronization with contention-free timeslots – the receiver here realigns its clock with those of the transmitter.

### 2.3. Time Synchronization

To exploit efficiently a schedule and to avoid the collisions, the network has to be globally synchronized. Thus, each node selects a parent to create globally a synchronization tree, rooted at the PAN coordinator. Then, a node readjusts continuously its clock when it exchanges packets with its parent.

More precisely, the transmitter sends its frames after a fixed offset from the beginning of the timeslot (Fig. 4). Thus, the receiver is able to compute the time difference between the expected and the actual time of reception. Since this difference corresponds to the clock difference, the receiver can readjust its clock after each reception. If no frame is received during a long time, keep-alive (KA) packets have to be transmitted to maintain the two nodes synchronized. The guard-time and the maximum clock drifts define the maximum period for these KA packets.

We can note that the backoff value denotes the number of shared cells to count before transmitting the enqueued frame. Thus, the offset for the beginning of the transmission is fixed even for shared cells: the receiver can readjust its clock after *any* reception.

### 2.4. Scheduling

All these approaches rely on a global scheduling matrix. In particular, a scheduler has to assign enough links for each transmitter, and its actual implementation was left unspecified. This authorizes to construct schedules with different objectives. For instance, a network may balance the load among the different nodes while another one would minimize the end-to-end latency.

Some solutions assume a schedule may be pre-computed, and installed in each device. However, pre-configuring the devices may constitute a challenging task. Alternatively, the schedule may be changed dynamically, with an approach inspired from the Path Computation Element (PCE) [21]: new cells are reserved when a flow is inserted in the network. 6TiSCH [22] currently works on providing the mechanisms to change on-the-fly the schedule in each device, adopting either a centralized or a distributed approach.

To improve the end-to-end reliability, we can also schedule redundant paths. *Duocast* [18] helps to achieve such objective: a packet is transmitted, and may be acknowledged by any of the two different receivers. Thus, a single transmission is sufficient to *cover* both receivers, improving de facto the reliability. This technique is efficient only if the packet losses are uncorrelated for both receivers.

To support different qualities of service, the scheduler may select different paths for each type of flow. For instance, the source assigns a Graph ID for each of its packets in WirelessHART [17]. By configuring a different next hop for each Graph ID, a packet is forced to use a given route with some specific QoS criteria (e.g. latency vs. energy efficiency). 6TiSCH introduces the concept of *track*, representing a set of cells along a route from the source to the destination [23]. Thus, the scheduler is able to allocate a given bandwidth (amount of cells) for a set of end-to-end flows.



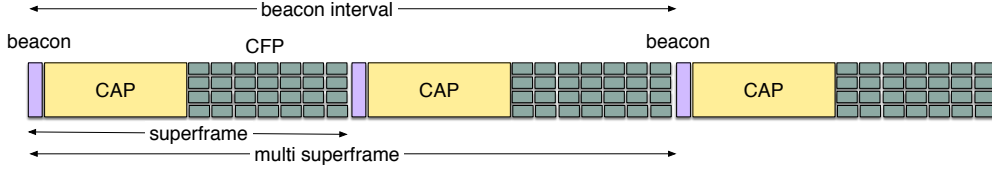


Figure 5: Superframe of IEEE 802.15.4-DSME.

### 2.5. Other standards

While we focused here on the description of IEEE 802.15.4-TSCH, other protocols exist which share the same scheduling concepts. To a certain extent, most of the scheduling approaches and algorithms presented in this survey can be adapted to different standards relying on the same principles.

The Deterministic and Synchronous Multi-channel Extension (DSME) mode extends IEEE 802.15.4-2006 which provided only seven reserved timeslots per superframe for real time traffic. DSME adopts a multisuperframe structure (cf. Figure 5): the superframe starts with a beacon so that all the receivers can re-adjust their clock. Then, the Contention Access Period (CAP) can be used by any transmitter, implementing a random access method, with a random backoff value before the transmission. Finally, the Contention Free Period (CFP) is located at the end of the superframe: a pair of nodes has a dedicated channel access and can transmit there its real-time packets.

Beacons of all the coordinators have to be scheduled to avoid collisions [24]. By selecting different instants for the beacons of interfering coordinators, a node can discover fastly a coordinator to join the network [25].

However, the transmissions have to be carefully scheduled during the CFP: each transmitter must receive enough transmission opportunities to respect a certain level of quality of service. Only the slow channel hopping feature is not available in DSME, but this characteristic does not impact the design of the scheduling algorithm.

However, a DSME slot is unidirectional: it does not include the acknowledgement part. Thus, the scheduling algorithm has to consider only interference from the transmitters to the receivers, and not among transmitters and receivers. Most of the algorithms presented in this survey are adaptable, by modifying the interference model (for centralized approaches) or considering the direction of the transmissions (for distributed algorithms).

While WirelessHART shares most of its principles with TSCH, it presents minor differences [26]. In particular, the schedule length is fixed and contains 100 slots, of 10ms. Besides, WirelessHART was not designed for distributed scheduling, and supports only a centralized *Network Manager*. Finally, it allows only 15 different channels, i.e. the channel 26 is excluded. However, the centralized scheduling algorithms designed for TSCH can be easily adapted to these characteristics.

## 3. Preliminaries

The Industrial Internet of Things has enabled a large set of applications. While traditional wireless sensor network applications have been assumed as delay-tolerant, real-time requirements and optimal scheduling are of utmost importance in industrial (wireless) networks. In these environments, some specific requirements arise, especially regarding real-time communications [27]. Indeed, sensor data may trigger chain reactions from the industrial facility wherein it has been generated. Thus, delayed sensing data would be either useless or detrimental to the decisions made upon this monitoring system. Industrial applications impose often deadlines to respect.

Unfortunately, the environment presents characteristics (e.g. dust, heat, water) that make those operations challenging. Wireless links are also impacted by surrounding electromagnetic interference from other wireless devices [28]. Scheduling policies have consequently to be designed under various scenarios (e.g. unreliable fading channels, rate adaptation, dynamic topologies). Such operations require to take various



aspects of wireless real-time communications into account (e.g. per-packet delay bound, timely-throughput requirement of each flow, heterogeneity of wireless channels).

Real-time communications would require both long-term and short-term guarantees. Guaranteeing a long-term fairness is for instance not enough: some constraints have to be respected on a per-packet basis.

Furthermore, most existing solutions rely on centralized decisions made by an omniscient access point, responsible to determine the communication schedules. We envision more complex scenarios, with multiple colocated deployments or fully distributed decisions, made at each node depending on its local environment and surrounding nodes.

The Industrial Internet of Things is expected to comprise a large set of nodes in a vast geographical areas. To deal with large densities, the frequency reuse must be optimized by enabling multihop technologies while ensuring a network wide connectivity [29]. Even in such complex topologies, we have to guarantee scalability (e.g. number of nodes, number of communication flows) and viability (e.g. limited computational and storage resources).

To reduce the deployment costs, several applications / services may cohabit on top of the same topology [12]. However, each flow has its own requirements, and its performance should not to be impacted by the rest of the traffic. The schedule may be constructed to respect perfectly a flow isolation, assigning bandwidth to each flow independently, taking care of the buffer space in each forwarding node, etc. Alternatively, some scheduler may exploit statistical multiplexing to guarantee the minimum requirements for each flow, while re-using the same slots for different flows.

### 3.1. Constraints & Feasibility

Real-time wireless communications need to respect strict guarantees, concerning in particular a minimum of packets delivered during a given time interval [30]. Hou *et al.* define necessary and sufficient conditions to verify that a set of requirements is *feasible*, i.e. a schedule exists, which achieves the expected minimum delivery ratio, over a set of radio links with a given reliability.

Karnik *et al.* [31] studied the scheduling problem in wireless multihop networks. They exploited the graph of conflict, i.e. the set of interfering radio links, to derive the maximum capacity. In particular, they proved that constructing a schedule for the most loaded clique is a necessary and sufficient condition to construct a feasible global schedule. In particular, the clique around the sink is the most important constraint with a convergecast traffic pattern. Prabh *et al.* provide bounds on the schedule for a convergecast traffic pattern [32]. However, they only address hexagonal mesh topologies, with a 1-hop interference model – a pair of transmitter/receiver *blocks* all its radio neighbors.

Soua *et al.* [33] construct an upper bound on the delivery delay, which can be applied to any connected topology. For a network deployed with a sink with  $k$  interfaces, the minimum number of timeslots required to complete a convergecast is  $\left\lceil \frac{N-1}{\min(k, n_c, n_{channel})} \right\rceil$ , where  $n_c$  is the number of children of the sink and  $n_{channel}$  the number of channels.

DISCA [34] also provides some bounds on the schedule, considering variable parameters such as the number of radio interfaces per node, and the number of channels. For instance, when the number of channels exceeds the number of children of the sink, the constraint relies in the subtree rooted at each child, and not anymore in the sink. The authors also proved that the schedule with and without acknowledgement has the same size, when considering a 1-hop interference model.

### 3.2. End-To-End Delay and Buffer Overflows Constraints

Many industrial applications require that the packets are delivered before a given deadline [27]. In wireless multihop networks, each forwarding node has to listen to the medium during the incoming cells, and can only transmit its packets during the outgoing cells. Thus, a forwarding node has to buffer a packet between each incoming and outgoing cells.

Let consider in Figure 6 the schedule of a forwarding node (B) in a simple 3 node-topology. If the node B receives a packet during the first incoming cell, it has to wait two cells for the next outgoing cells (delay 1). Similarly, it has to wait 7 cells if it receives a packet during its second incoming cell – the slotframe is cyclic, repeated indefinitely.

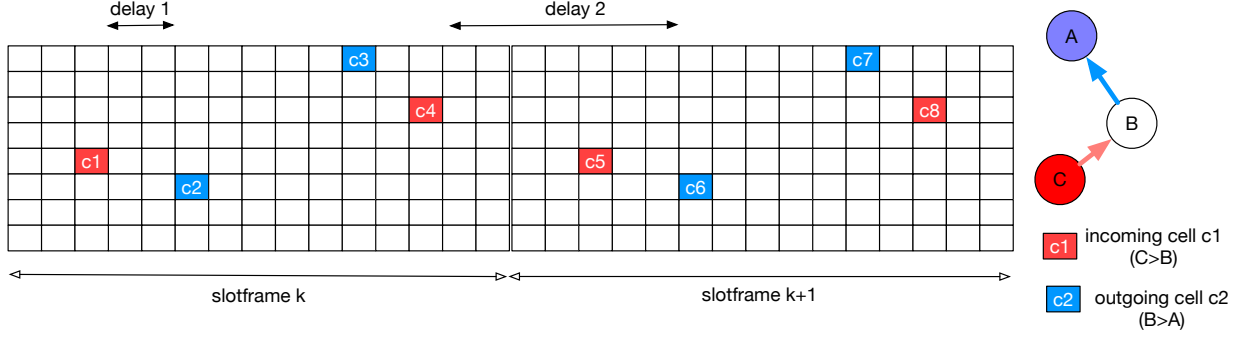


Figure 6: Schedule of the node B with 2 outgoing cells and 2 incoming cells.

The minimum bound on the number of timeslots required to transmit all the data from the nodes to the sink is crucial to respect the strict deadlines imposed by industrial applications. Incel *et al.* [35] shown that the minimum schedule length in a tree topology is  $\max(2n_k - 1, N)$ , where  $n_k$  is the maximum number of node in any subtree and  $N$  is the number of sources node. Additionally, the scheduler has to ensure a proper slots allocation order for each flow and to consider different link qualities along its path to the sink. Otherwise, a node might be inadequately scheduled to transmit before it receives the data to forward.

The buffer occupancy is also strongly correlated with this buffering delay: a longer delay increases the probability to create a buffer overflow in the forwarding nodes. Let consider the topology and the schedule in Figure 6. If the node B receives a packet during two consecutive incoming cells (C4 and C5), the node has to store two packets in its buffer before the next outgoing cells (C6 and C7). If the buffer is too small, some of the packets have to be dropped: even if transmissions are perfectly reliable, the system as a whole is unreliable.

### 3.3. Traffic Characteristics

To be energy efficient, the schedule should be traffic adaptive. We make the following distinction among the different types of traffic:

**Uplink:** a packet is generated by a node, and forwarded through a path to the sink;

**Downlink:** the sink has to send a specific packet to one device;

**Any2Any:** a packet is generated for another device, and is forwarded along a path which may not pass through the sink. This type of communication involves typically a sensor which notifies an actuator;

**Multicast:** a packet is generated for a set of destinations, which all have to receive it;

**Broadcast:** a packet has to be delivered by all the logical neighbors. A broadcast is typically used to announce its presence via a **beacon**.

Most of the scheduling solutions address the convergecast traffic pattern, where all the traffic is generated by the nodes and transmitted to the sink (uplink). Han *et al.* [36] propose to construct three different routing graphs (uplink/downlink/broadcast) guaranteeing the required end-to-end reliability independently to each type of transmission.

The traffic intensity has obviously a strong impact on the schedule construction. Because the typical MTU is very small in the IIoT (e.g. 128 bytes), an application payload may be split (i.e. *fragmented*) into several Layer-2 frames. Thus, the scheduling algorithm has to provide guarantees by considering the packets in groups (of fragments). For instance, to achieve a given end-to-end delay, the *last* fragment must be received by the sink before the deadline. Similarly, respecting a 99% reliability for each fragment

independently is not sufficient to achieve a 99% per flow reliability (losing one single fragment makes the reconstruction impossible, even if all the other ones have been received correctly).

We identify two main types of traffic:

**Constant Bit Rate (CBR):** each source generates its packets, and maintains the inter-packet time constant. For instance a network used in Structural Health Monitoring (SHM) applications collects data continuously at constant data rate [37]. With CBR traffic, the scheduler knows exactly when each packet is generated by each source node;

**Event-Based:** the application generates a packet when it detects an event (e.g. a measure above a threshold, a physical stimulus). For instance, a fire alarm application may generate a bursty traffic when it detects smoke [38]. This type of traffic is much more complicated to handle because the schedule has to be constructed for the worst case scenario.

### 3.4. Radio Link Characteristics

Low power radio networks are known to be lossy [39]: packet losses are frequent. In particular, industrial environments comprise many obstacles, and multipath fading is the rule. Expected Transmission Count (ETX) [40] was one of the first metrics to estimate the link quality, counting the expected number of packets to transmit before receiving an acknowledgement. ETX reflects both the reliability, through the Packet Delivery Ratio (PDR), and the energy efficiency of a radio link. However, ETX does not take into account the available throughput of the link and it may perform badly under heavy traffic load [41].

Radio links may also be asymmetrical, because of e.g. different transmission powers or noise floors [42]. Srinivasan *et al.* [43] consider a link as asymmetrical when the PDR in both directions differs by more than 40%. A scheduling algorithm may consider easily this asymmetry by handling a directed graph, with different radio link qualities in both directions.

External interferences are very common in industrial environments. To combat narrow-band noise, slow channel hopping has been proved to perform very well in industrial environments [11].

Unfortunately the radio link quality is time-variant [44]. While very good quality links tend to be very stable, the other medium quality links may exhibit significant long-term variations. Constructing prediction models such as [45] may help to construct an accurate schedule, tailored for the worst case scenario.

Packet losses do not exhibit a perfectly random property. Some time periods exist during which all the packets are dropped in sequence: a link oscillates between the *good* and *bad* states. This so-called *link burstiness* is a consequence of obstacles or external interference [46], and has a strong impact on the reliability: the scheduling algorithm has to provision enough retransmission opportunities during this *bad period*.

To provide high-reliability and low latency, the scheduling algorithms have to take into account the link quality. For instance, retransmissions have to be provisioned for unreliable links. Alternatively, another path constituted with only good links should be identified.

## 4. Existing approaches at a glance

Slow channel hopping technologies with TDMA rely on constructing a common schedule: a transmitter receives a list of *tx cells* (timeslot and channel) during which it has the right to transmit a packet for a given neighbor. Symmetrically, the algorithm also allocates the *rx cells* during which the nodes have to listen for incoming packets.

While the scheduling algorithm is left unspecified in the standard, it largely impacts the performance of the system. Most of these algorithms can be *adapted* to deal with any of these standards / protocols (i.e. DSME, TSCH, Wireless HART). In this survey, we aim to provide a comprehensive view of the existing scheduling algorithms, regardless the standard used in their performance evaluation.

The different scheduling algorithms can be classified in the following categories:

**Centralized:** a sink has the full knowledge of the network characteristics (traffic, interference, etc.) and allocates a set of transmission opportunities to each radio link. For instance, in Path Computation Element (PCE) approaches, a central controller allocates the resource;

**Hierarchical:** the algorithm is executed on top of a pre-established hierarchical (tree) topology. For instance, the routing protocol is executed first to construct the best routes to the sink. Then, the scheduling algorithm recursively allocates bandwidth to the links from each node to its next hop.

**Distributed:** nodes negotiate only with their *neighbors* (i.e. nodes at a maximum hop or euclidean distance) to decide which channel/timeslot to use.

#### 4.1. Constraints for the Scheduling Algorithm

With slow channel hopping techniques, the algorithms have to respect the following constraints:

**Half-duplex constraints:** the nodes are equipped with radio interfaces that can either transmit or receive, but not at the same time. A node cannot be scheduled in the same timeslot with different channel offsets;

**Interference constraint:** two transmitters which would create a collision in at least one of the two receivers cannot be scheduled in the same timeslot and channel offset;

**pre-existing routes:** the scheduler may be executed on top of a routing protocol. In this case, the scheduling algorithm just allocates bandwidth along the different pre-existing paths. In any other case, the scheduler has to decide both which route to be used and how much bandwidth to be reserved for each flow.

In multipath routing, the scheduler can distribute the load among different paths;

**application requirements:** the schedule has to respect end-to-end constraints, such as the delay and the reliability [47]. For instance, some retransmitting slots have to be scheduled to improve end-to-end reliability, while respecting the end-to-end delay constraint.

**buffer length constraint:** nodes can often store a limited number of packets in their buffer. To avoid dropping a packet because of a buffer overflow, the schedule has to consider this memory constraint;

**hardware constraint:** if the node has to directly execute the scheduling algorithm, the limited hardware capabilities (CPU, memory) should also be considered.

#### 4.2. Taxonomy

We will detail here the classification we adopted to present the different approaches. We first make the distinction between the approaches which rely on a central controller, such as a Path Computation Element (PCE), and the distributed algorithms directly executed by the node themselves.

Indeed, these two families rely on very different approaches, and are tailored for different scenarios. The PCE-based approaches are well-suited for very stable topologies, where the central controller can have a precise view of the traffic and radio characteristics. Inversely, distributed solutions tend to be more robust to changes without making any a priori assumption neither on the radio topology, nor on the volume of traffic to transmit.

To our mind, distributed solutions cannot be classified according to their optimization objective since all the nodes miss a precise, global view. They rather aim to construct a valid schedule able to deliver all the packets. Thus, we split the distributed family according to its characteristics, into the solutions dealing with a stationary topology, and those tailored for mobility support. Indeed, mobility support requires to implement specific and aggressive approaches, such that a mobile node keeps on reserving enough bandwidth along a valid path, even when radio links change (i.e. a kind of multihop handoff with resource reservation).

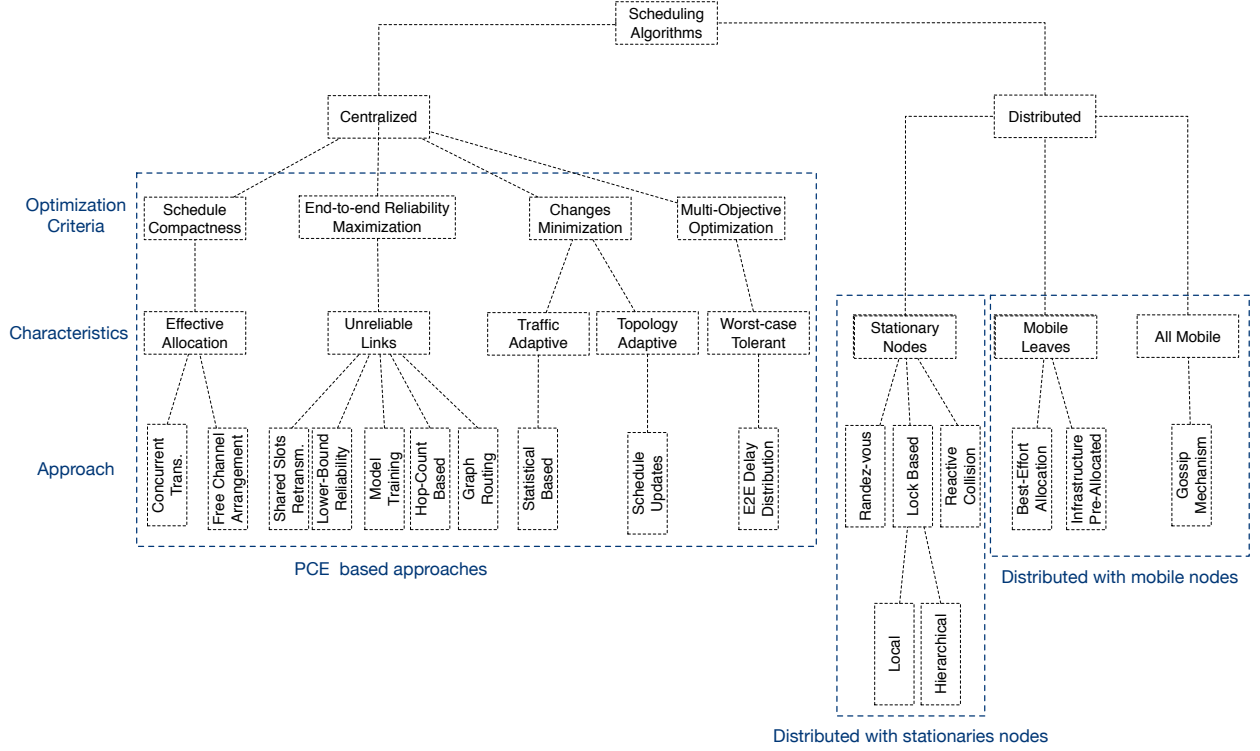


Figure 7: Taxonomy of slow channel hopping scheduling algorithms.

#### 4.2.1. PCE based Approaches

Centralized schedule algorithms take advantage of an omniscient entity that has full knowledge about the current network status. Some algorithms implicitly assume that this knowledge is gained before the network deployment while some others rely on the collection of statistics after the network initialization. In this case, the scheduling algorithm corresponds to an optimization problem, trying to minimize (or equivalently maximize) a given objective.

We consequently chose to classify these algorithms according to their objective function: what criteria does each algorithm try to optimize?

**Schedule’s Compactness.** A compact schedule means that the matrix should be as compact as possible: the number of channel offsets and timeslots not used by any transmitter must be maximized. A compact schedule may be convenient to admit later new flows: enough space is available to allocate more bandwidth.

To construct a compact schedule, the algorithm has to allocate non-interfering links during the same cell. We can choose to minimize the number of timeslots required, and to exploit the maximum number of channels (Figure 8a) or inversely to minimize the number of channels (Figure 8b). Exploiting an *horizontal* schedule may be efficient when blacklisting techniques are required because many channels perform badly [48].

**End-to-End Reliability Maximization.** Because many industrial applications require high-reliability [36], we must be able to guarantee a certain end-to-end Packet Delivery Ratio (PDR) for all the flows. Unfortunately, the radio links are known to be unstable, and unreliable [44]: the scheduling algorithm has to allocate several transmission opportunities to deal with retransmissions.

The challenge consists here in allocating a sufficient number of slots for each radio link, while still limiting the overprovisioning which negatively impacts the energy efficiency. A pool of cells can be allocated for all the retransmissions, each node being able to pick one of them when one of its transmission fails. Inversely, the cells may be dedicated to maintain the network deterministic and fully predictable.

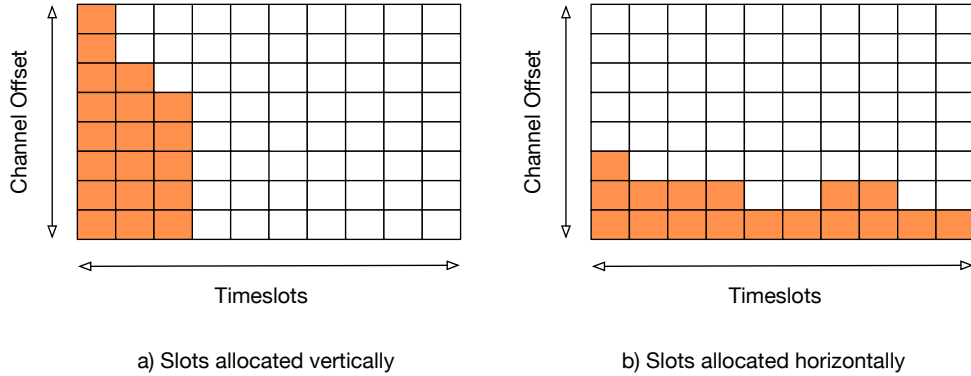


Figure 8: Vertically and horizontally filling pattern.

Because the radio links exhibit very different properties, it may be relevant to *learn* the key characteristics of the radio environment. These techniques need to construct some models, parameterized thanks to a pre-deployment, assumed to be *representative* of the final topology.

Finally, reliability may also be improved by exploiting several different, but possibly partially overlapping paths. The backup routes provide a certain diversity, and the network is able to recover after a temporary or permanent fault.

**Changes Minimization.** Several deployments should be sufficiently flexible to support for instance multiple applications, or variable conditions [49]. Unfortunately, most of the scheduling algorithms do not assume a pre-existing, operational schedule: they recompute from scratch a novel schedule.

To reduce the cost of re-installing a novel schedule, a scheduling algorithm should minimize the number of changes between the previous and new acceptable schedule for the actual conditions (traffic, interference). Call admission may alternatively help to avoid reconfigurations: each source has to send a request to the PCE, which rejects the request if not enough bandwidth is still available (i.e. no schedule reconfiguration is in this case authorized).

**Multi-Objective Optimization.** Some algorithms try to optimize several criteria. For instance, many industrial applications require strict delay bounds: a packet has to be delivered before a given deadline. Thus, it may be interesting to compute the Pareto-optimal schedules concerning both the end-to-end delay and the energy consumption. It is therefore possible to investigate the achievable tradeoff between latency and energy efficiency. For such approach, the reliability is considered as a constraint (each flow must respect a certain minimum end-to-end reliability).

#### 4.2.2. Distributed Approaches – Stationary Nodes

With distributed algorithms, each node has to exchange information with its neighbors to construct a consistent schedule. In particular, the same cell should not be allocated to two interfering transmitters. Besides, the schedule must respect some PDR and delay requirements. In such scenarios, the nodes adjust their schedule dynamically negotiating bandwidth with their neighborhood nodes. The schedule is tuned when the traffic demand increases/decreases or when the topology changes. We focus here on the stationary nodes scenario, since mobility support requires specific approaches.

Rendez-vous based methods allow each node to construct its schedule without exchanging any packet. By selecting accurately its schedule, any pair of nodes has probabilistically at least one transmission opportunity to exchange data packets. The challenge consists in determining a relevant pseudo-random function to construct an uniform quorum system [50].

The reservation based solutions need to send control packet to agree locally on a consistent, collision-free schedule. Thus, a node has to *lock* a given cell either locally in its neighborhood, or hierarchically (hop-by-hop toward the sink). The reservations are often assumed to be terminated sufficiently fast to avoid conflicts.

Self-healing approaches adopt the opposite method: a node can use any cell, without reservation (and the associated control packets). Since collisions may arise, the concerned nodes must detect and solve them probabilistically. Such methods need a stable environment, with low traffic conditions (i.e. enough bandwidth is available to resolve probabilistically the collisions).

#### 4.2.3. Distributed Approaches – Mobile Topologies

Some distributed approaches target networks where some or all the nodes may move around the environment. They rely on mechanisms that minimize the attachment delay, making the joining time faster and reducing the latency of messages sent by the mobile nodes. We identified two scenarios for mobile networks: (i) the network relies on fixed wireless routers (only end-devices are mobile), (ii) all nodes may move freely. Besides, we identified the following approaches.

Most of the solutions provide a 2-level schedule. The system allocates a set of dedicated cells to the infrastructure nodes to provide them a real-time access, and enough bandwidth to forward their packets. On the other hand, the cells for the mobile nodes are either dedicated or shared among all the mobile nodes. Multiplexing statistically the transmissions of mobile nodes are more energy efficient since it reduces the over-provisioning, but has a negative impact on the reliability, since the system is not anymore deterministic.

In highly mobile scenarios, the reservations rely on Gossip mechanisms to disseminate the cells selected by each node for its transmissions. The redundancy of the transmissions helps to reserve quickly a cell, and to detect quickly inconsistencies in the schedule to solve the collisions.

#### 4.3. Supported Characteristics

Table 1 summarizes all the papers detailed in this survey, classified according to their key characteristics. These characteristics were identified after extensive literature review of all selected works. We identified the common characteristics shared by most of them. We believe that these are the most important characteristics of scheduling solutions that target applications with high-reliability and real-time delivery requirements. Moreover, network designers can take advantage of this comparative study to select the best algorithm for their application, according to its specific requirements.

We detail now the selected characteristics:

1. Variable traffic: refers to the scheduler ability to adapt itself when the traffic increases or decreases;
2. Fixed topology: some algorithms achieve higher performance when the network topology does not change;
3. Faulty links: the algorithms propose mechanisms to deal with time-invariant transmissions failures;
4. Link burstiness: the scheduler considers that the links pass through a period of instability where the packets are dropped in a burst;
5. Buffer overflow: the algorithm presents some mechanisms to deal with a limited amount of memory;
6. Multiple interfaces: the devices (sink and/or nodes) allow multi-radio communications;
7. Evaluation: the algorithm has been evaluated analytically, in simulation and/or in testbed;
8. Interference: it refers to the interference assumption used for evaluating the algorithm. The interference can be known before the network deployment, based on either the number of hops or distance. Some algorithms also avoid to schedule concurrent transmissions on the same channel;
9. Goal of the algorithm: it highlights the main objective of the algorithm.

Many approaches rely on a centralized scheduling algorithm to construct globally an efficient schedule. While some of them try to adapt the schedule to complex situations (e.g. burstiness), most of these



[illegible]

approaches implicitly target simple and ideal scenarios: networks with fixed topology and traffic rate. Furthermore, few centralized approaches evaluated their performance in real testbeds, which makes simulation the most widely used tool for evaluation.

Most of the mobile-aware solutions are distributed, combining static and mobile nodes with constant traffic rate. While much attention has been given to minimize the joining time, they do not propose mechanisms to improve the end-to-end reliability. Furthermore, few works were tested in real testbeds and some of them do not consider any internal or external interference.

Finally, distributed solutions target networks with dynamic traffic rate and topology. The distributed approaches are highlighted in the literature as the best option to deal with radio link quality / traffic variations. However, they have not been designed for mobile scenarios and require adaptations to deal with such dynamic situations.

## 5. PCE based Scheduling

We will describe here the scheduling algorithms that are built and maintained by a central controller. This kind of deployment is very common in industrial scenarios [98, 99]. A network has to provide deterministic performance and fulfill real-time communication requirements. As the controller has a global view of the network, its job consists in constructing an optimal schedule (concerning a criteria), while respecting a set of constraints (e.g. traffic, routes). Creating this schedule while considering all these aspects represents a NP-complete problem [100].

As pointed out by Tsitsiklis *et al.* [101], the centralized architecture is ideal in static networks where every property is known precisely. Indeed, a distributed approach can only use local information, and consequently may make sub-optimal decisions.

### 5.1. Schedule's Compactness

Some algorithms prioritize to allocate simultaneous transmissions at the same timeslot, while using different frequencies for conflicting nodes. Inversely, some other algorithms compute the schedule to be channel-optimal: they aim to allocate the minimum number of channels. Independently of the approach, both implicitly consider a previous deployment knowledge (e.g. topology, traffic).

#### 5.1.1. Concurrent Transmissions

The following approaches maximize the network throughput by allocating the same cell (channel offset and timeslot) to conflict-less nodes, while allocating different channels when two interfering nodes use the same timeslot. They transform the construction of a conflict-free schedule in an optimization problem, solving it by using graphs or meta-heuristics algorithms.

TASA [51] [52] is a multihop centralized scheduling algorithm for slow channel hopping. It considers a hierarchical network, where every node has only one parent and the traffic flows from the leaves to the sink. TASA aims to find a compact schedule (see Figure 9), minimizing the slot number of the last used cell in the slotframe. The algorithm gives a larger priority to nodes with more traffic to forward. This way, it tries to allocate bandwidth first to the most constrained nodes, using different channels for conflicting links. For this purpose, it uses a combination of matching and coloring algorithms to build an appropriate scheduling.

The authors of TASA evaluated the efficiency of their algorithm by simulation, showing good performance using four channels at most. Sempere-Pay *et al.* [57] evaluated TASA experimentally in a multihop network with 10 OpenMote-CC2538 motes and a slotframe of 100 slots. They observed high packets losses even for nodes that were one hop distant from the sink. For the shortest slotframe of 15 slots (i.e. 150ms), and when a packet is generated by each source every 10 seconds, most of the cells are unused in the slotframe. This configuration provides highest reliability, but also the highest energy consumption since it exploits a very high duty cycle ratio. Inversely, TASA provides a low PDR with a slotframe of 100 slots (1 second), and a traffic of 1 pps. Because no cell is over-provisioned for retransmissions, most of the packets are dropped.

A similar approach is adopted by Ojo and Giodano [55]. They exploit a graph based approach, and maximize the network throughput through a *maximization problem*. To reduce the problem complexity,

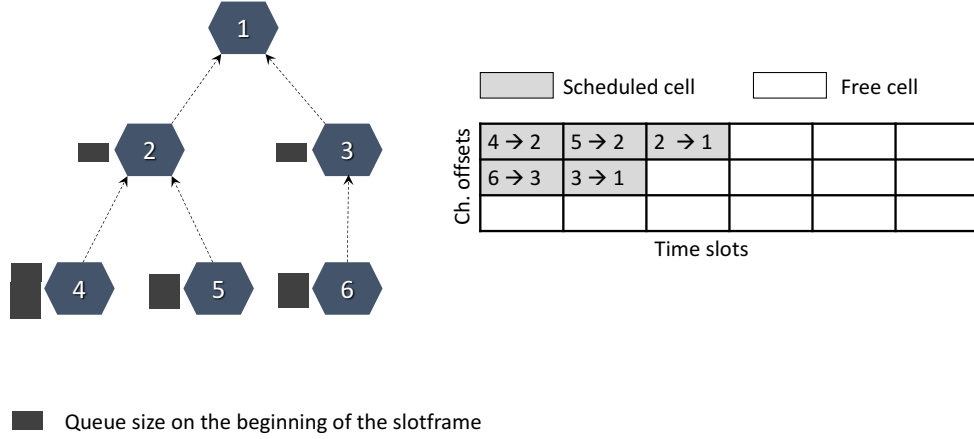


Figure 9: Example of a compact scheduling built by TASA with 3 channels offsets and 6 time slots. In this example, the transmissions are scheduled as soon as possible.

they use a matching to transform the scheduling into a maximum weighted bipartite matching (MWBM) problem. Afterwards, they use the Hungarian algorithm to solve it in polynomial time.

MODESA [33] deals with a sink equipped with several radio interfaces. Thereby, it allocates less slots and achieves higher throughput. Having multiple interfaces helps to improve the performance and the scalability [102]. Chen *et al.* [53] also adopt a greedy approach to schedule non interfering links in the same cell.

Lee *et al.* [56] adopt a meta-heuristic approach to create a conflict-free schedule that satisfies an end-to-end delay bound. They compared the delay bound achieved by the Simulated Annealing (SA) and the Participle Swarm Optimization (PSO) techniques. Their results indicate that PSO converges more slowly, but after convergence, the maximum end-to-end delay is approximately half the one achieved by SA.

MABO-TSCH [54] combines a centralized scheduling and a local blacklisting negotiation. The (centralized) scheduler exploits a coloring problem where the nodes are sorted according to their degree in the graph. To be reactive, a pair of nodes decides locally the physical channels to blacklist. To maintain a consistent blacklist among the receiver and the transmitter, they piggyback the blacklist into the ACK frame, combined with a sequence number. To estimate the channel quality, MABO-TSCH exploits the Stochastic Multi-Armed Bandits with 16 arms (one for each available channel).

### 5.1.2. Free Channel Arrangement Approaches with Perfect Links

The following approaches assume the channel frequency is a scarce resource and they compute the schedules minimizing the number of allocated channels.

In [58], a latency-optimum scheduling algorithm for linear networks is proposed, the sink being located at one extremity of the line (e.g. a production line). For every timeslot, the algorithm executes a forward and backward scheduling. The first step considers iteratively the nodes, ranked according to their distance to the sink. The algorithm allocates a timeslot to a node if it has data to send, and has a neighbor able to receive it. Next, the algorithm considers the nodes backwardly, from the farthest to the closest one. It considers the half-duplex condition to allocate the channel offsets, avoiding a node to be involved in several transmissions. The algorithm aims to minimize the number of used channels and prioritizes the allocation of timeslots to complete a convergecast. The authors also propose an extended version for multi-line [60], where the sink has multiple children (lines), and tree topologies [59]. Both works assign higher priority to lines or subtrees with a larger number of nodes.

Li *et al.* [61] exploit data aggregation techniques to reduce the number of transmissions. In this approach, the nodes with a common parent are scheduled sequentially, and prior to the parent in charge of aggregating and forwarding the packets. This aggregation mitigates the delay compared with an approach where the children transmissions are scheduled anarchically.

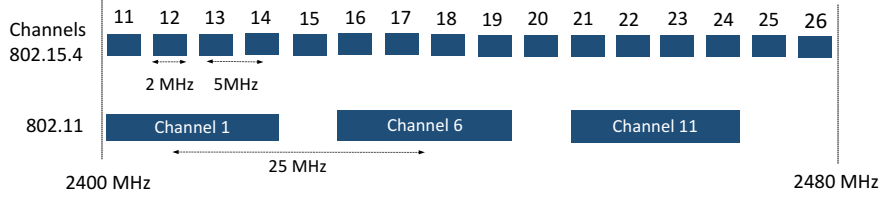


Figure 10: Overlapping channels between IEEE 802.11 and IEEE 802.15.4 standards.

### 5.1.3. Limitation

Most of the scheduling algorithms presented here do not consider packets losses. In real world industrial applications, a network may share the same unlicensed band with other networks. Musaloiu *et al.* [103] showed that a IEEE 802.11 network may increase the packet losses of an IEEE 802.15.4 network up to 58%. While slow channel hopping tends to increase the reliability, packet losses keep on occurring [104]. As shown in Figure 10, both IEEE 802.11 and IEEE 802.15.4 standards have overlapping channels.

Furthermore, the transmission power of IEEE 802.11 devices (30 dBm) is higher than IEEE 802.15.4 devices (0 dBm). This asymmetry impacts the performance and the reliability of IEEE 802.15.4 networks [105].

## 5.2. End-to-end Reliability Maximization

While packets may be lost at the link layer, some applications still require high end-to-end reliability. For instance, an application that monitors people health conditions [106] requires very high delivery rate, with tight delay constraints. Thus, this category of algorithms aims to find a feasible schedule while respecting the minimum end-to-end reliability. Most of the solutions focus on this reliability constraint without trying to optimize any given objective function: constructing a feasible schedule represents a sufficiently challenging objective.

### 5.2.1. Shared Slots Retransmissions

A set of shared cells can be used for retransmission attempts to increase the reliability. The shared cells are placed in different locations of the slotframe in order to minimize the number of collisions and to reduce the bandwidth consumption.

Yang *et al.* [28] propose to mutualize the retransmissions: some spare slots are reserved, used by *any* child which did not receive any acknowledgement for its previous transmissions. Thus, the algorithm has to take care of the precedence: all the receiving cells (from the children) have to be scheduled before the transmitting cells (to the parent). The algorithm also reschedules the timeslots when a link or a node fails. For this purpose, it uses the concept of free nodes, i.e. neighbors of the sink without any descendant. These free nodes can be rescheduled in any timeslot to let their place to other nodes with a new time constraint, created by the reconfiguration.

Figure 11 shows an example of this rescheduling feature. When the node 3 fails, the node 4 selects node 6 as parent. To ensure that the node 6 will receive all the packets before its retransmission slots, the new transmission from 4 to 6 should be scheduled at the beginning of the slotframe. However, because retransmissions use shared slots, collision free transmissions are not guaranteed, thus endangering end-to-end reliability.

Elsts *et al.* [62] propose also to deal with unreliable links by using shared cells. The schedule is constructed such that each node receives a *column* of the schedule, each line being allocated to a possible next hop. Some shared cells are also present to retransmit the unacknowledged packets: this feature is particularly efficient when the different links present very different PDR. Finally, a node can *steal* the cells of its descendants when they are sufficiently far in the tree.

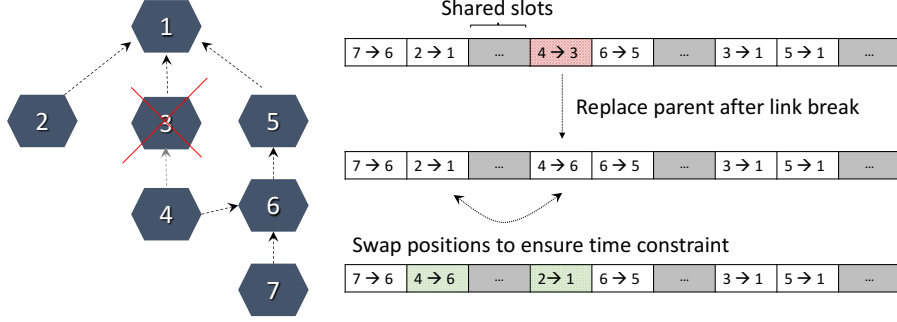


Figure 11: Reschedule transmissions to ensure proper transmission order. Ideally, each transmission group should have at least one free node to simplify the rescheduling [28].

### 5.2.2. Lower-Bound Reliability

These approaches use link quality metrics such as the Packet Delivery Rate (PDR) to compute the number of additional cells to insert in the schedule to respect a minimum end-to-end reliability.

SchedEx [63] fortifies the schedule, by allocating additional slots to lower-bound the end-to-end reliability. The number of extra timeslots is calculated in function of the packet reception rate of every link. However, SchedEx was initially designed for single channel scenarios. In slow channel hopping, we have also to consider the half-duplex constraint, which makes the problem more challenging.

TASA<sub>rtx</sub> [64] extends TASA to deal with non-perfect radio links (see Section 5.1). TASA<sub>rtx</sub> calculates how many additional cells are needed in the schedule to satisfy the expected end-to-end PDR. TASA<sub>rtx</sub> also considers the fragmentation of packets in order to transmit long messages over several slots. The algorithm helps to improve the reliability with lossy radio links.

Kausa [65] constructs a schedule which respects Service Level Agreement (SLA). It allocates slots considering different Key Performance Indicators (KPI) for each communication flow. The number of allocated cells for every link is directly related to its packet error rate (PER). Furthermore, transmissions are scheduled along the path to respect both the reliability and the latency constraints. Because it adopts a greedy approach, the algorithm uses a backtracking procedure where the most loaded and most vulnerable links are blacklisted, and it reiterates to find a valid schedule.

### 5.2.3. Model Training

The algorithms in this family pass through a training phase where the statistics of packet losses are computed for every link. The training phase should be long enough to model the links behavior correctly and to improve its accuracy. Based on analysis of the trained model, the algorithms compute the number of additional slots.

Munir *et al.* [67] calculate for every flow the number of slots to take into account the bursts of packet losses along the path to the sink. Figure 12 shows an example of timeslot allocation for the link  $2 \rightarrow 1$ . Its training phase consists in measuring the amount of consecutive failed transmissions ( $B_{\max}$ ) for each link. This information is crucial to determine the link burstiness level of each radio link. Munir *et al.* [67] trained their system over a 21 day period, which appeared as a minimum duration to obtain accurate measures (e.g. longer training required for stationary nodes with high  $B_{\max}$ ). Then, the scheduler avoids the links that exhibit a link burstiness level  $B_{\max}$  larger than a given threshold.

Pöttner *et al.* [68] consider also the transmission power to reduce globally the interference level. It keeps on requiring to collect data regarding the link burstiness before the network actually starts to operate. The authors investigated how much time was required to collect enough data to calculate the link burstiness. In the first scenario (a oil refinery), 21 hours were required to obtain consistent measures, while in an indoor

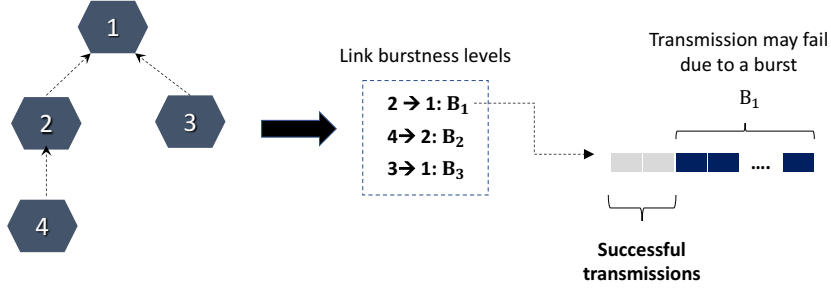


Figure 12: Example of cell allocation considering link burstiness.

environment, almost 142 hours were required. Indeed, numerous sources of interference tend to increase the convergence delay in indoor. Furthermore, throughout a 308 hour performance monitoring, the network never violated the requirement of 99% packet delivery rate. However, the training phase has to be restarted from scratch each time the radio link characteristics change.

#### 5.2.4. Node Position Based

In [66], authors assume that nodes closer to the sink have more traffic to forward, because of the funneling effect. Thus, these nodes are more subject to packet losses and they need to allocate more transmission slots to guarantee the end-to-end reliability. In this solution, Jin *et al.* consider the node data rate and its distance to the sink to compute the number of backup slots. The algorithm allocates first backup slots to the nodes that are 1-hop away from the sink. Next, the algorithm considers the next hop using the remaining free slots. This procedure continues until the last hop to allocate the remaining cells. However, the algorithm does not take into account the link PDR and may allocate unnecessary slots to nodes with good link quality. Consequently, a further node with bad link may not be well covered.

#### 5.2.5. Graph Routing

In a graph routing approach, each node has ideally at least two neighbors able to forward its packets toward the sink. The scheduling algorithm tries to allocate the cells on redundant paths to increase the robustness.

Dang *et al.* [69] allocate additional timeslots based on graph routing to implement hop level retransmissions. When the communication with a neighbor fails, the node tries to retransmit on a different channel. If the transmission still fails, the node tries to transmit to the other parent. The scheduling algorithm considers all these cases when allocating the cells to each radio link. However, the authors did only present a theoretical analysis, without evaluating the algorithm with any simulation or experiment.

A similar approach is adopted by Zhang *et al.* [70]. The authors aim to minimize the delay by allocating retransmissions slots only to the nodes on the optimal route. The optimal route is given by the algorithm ORMGR [107], a routing algorithm for WirelessHART networks that constructs a connected graph from the source to the destination, by using statistics of packet losses. As in [69], the authors evaluated the algorithm only analytically.

Huynh *et al.* consider opportunistic anycast scheduling, i.e. a single transmission is received by a collection of next hops which decide opportunistically which one will forward the packet [71]. They demonstrated that opportunistic scheduling helps to increase the network capacity, even exploiting a Gilbert-Elliot channel model. To schedule accurately the transmissions, their Approximate Dynamic Programming (ADP) approach relies on the knowledge of the channel state, complicated to obtain in practice.

Lee *et al.* [72] schedule dedicated links for a redundant communication path. They exploit the max-min algorithm to optimize the number of dedicated slots necessary to establish an alternative route once the link fails. The algorithm properly allocates the slots in order to meet the delay bounds after the route switching.

Similarly, Wu and Lee [73] reserve sections of the superframe to allocate the transmissions for an alternative path when the link between two nodes fails. Each section is used by a different set of nodes that are equidistant to the sink. The length of these sections (number of reserved slots) depends of the traffic rate and the number of nodes that may use them.

#### 5.2.6. Limitation

Most of the algorithms presented here assume that the link characteristics are time-invariant, which is not the case in most practical scenarios (see section 3.4). Thus, they may fail to provide end-to-end reliability, when e.g. packets are dropped in bursts [108]. Exploiting the average PDR for each radio link may not be sufficient to compute an efficient schedule. More specifically, PDR overestimates the link quality and it can still yield high values during small periods of instabilities [109].

On the other hand, the link burstiness aware approaches must compute the number of successful and failed transmissions. To the best of our knowledge, no method is able to predict systematically how long the training phase should last in order to collect statistically relevant measures to calculate the link burstiness level. Thus, tuning this training phase may be practically complicated.

While these algorithms successfully compute schedules with end-to-end guarantees and respecting latency bounds, they assume implicitly stable conditions. If the deployment area changes (e.g. new obstacles) or new external interference arise, the previously calculated link burstiness level may change and a new dataset is necessary. Collecting the measures during the training phase is often incompatible with a normal behavior of the industrial network: the manufacturing process has to be stopped, which seems unrealistic in many situations. To the best of our knowledge, an online measurement method, cohabiting with the normal data schedule, is yet to be proposed.

### 5.3. Schedules Changes Minimization

Some centralized scheduling algorithms are traffic or topology adaptive: they do not rebuild the schedule when these characteristics change. These approaches make a tradeoff between the loss of optimality, and the cost to update the schedule.

#### 5.3.1. Statistics Based

The traffic pattern in industrial deployments tends to be time-invariant. In [74], i-MAC takes advantage of this predictable behavior, to detect and solve the collisions. It constructs a centralized schedule to deal with bursty traffic. The sink is aware of the nodes traffic pattern and the probability of collisions. To compute this probability, i-MAC identifies the time period during which a burst of traffic is detected. Then, the algorithm schedules during the same timeslot the nodes which have a low probability of collision. On the contrary, colliding traffic is scheduled in different timeslots. i-MAC implements a *frequency-transceiver* hopping: when collisions occur in sequence, the transmitter and the receiver select another channel for the next transmissions. All the nodes follow a common channel mapping function that selects a different channel to transmit on.

#### 5.3.2. Schedule Updates

Because the topology may change (e.g. new nodes, parent swithingm, etc.), some centralized algorithms update the schedule dynamically according to the new state of the network.

Montero *et al.* [76] investigated the impact of the utilization of a centralized scheduling in an industrial scenario where nodes use passive discovery method. In order to join (or rejoin) the network, the mobile node must be in receive mode to listen for advertisements sent by the static nodes. The advertisements include which timeslots the mobile node should use to send a joining request, forwarded by the static nodes to the sink. Finally, the sink replies with the list of allocated slots to the mobile device, valid until it changes its point of attachment. As highlighted in the experiments, mobility requires a short slotframe, and the handoff has still a strong impact on the performance, dropping packets until a new cell is allocated.

In Centralized Link Scheduling (CLS) [75], a new node has to reserve bandwidth before joining the network. The joining node transmits a **join-request**, forwarded along a tree topology toward the sink. Then,



the sink allocates bandwidth for each intermediary hop, and *installs* the schedule during the *join-reply*. When a node changes its parent, it sends a removal request to the sink, deallocating the cells in each intermediary hop.

### 5.3.3. Limitation

To validate a schedule using a centralized architecture is expensive: the sink has to verify that no colliding pairs of cells exist in the schedule. Unfortunately, such detection requires to collect a large volume of statistics, expensive to send to the sink, especially for the most distant nodes. Reducing this communication overhead requires efficient mechanisms to estimate the link quality levels at the sink.

## 5.4. Multi-Objective Optimization

Some industrial applications require to consider multiple, possibly antagonist objectives. To investigate the mutual impact of these criteria, Pareto-optimal schedules are derived.

### 5.4.1. E2E Delay and Energy Minimization

Because many industrial applications require to respect strict guarantees, Wang *et al.* [77] proposed a framework to construct Pareto-optimal schedules. They define a relaying matrix to determine the probability for each node to forward a packet received from a given neighbor. To be formulated with a Markov-based model, each node is allowed to transmit at most once each frame it receives, i.e. retransmissions are not considered. To cope with unreliable links, multiple forwarders may be selected for each hop: the link-layer transmissions are considered opportunistic, transmitted in anycast to any registered next hop node. With the Markov model, the authors derive worst case delay bounds, and are able to construct a schedule which minimizes such end-to-end delay.

### 5.4.2. Limitation

The computed schedule is fixed and needs to know precisely the amount of packets to forward and the PER of each radio link. Since the algorithm assigns a probability to forward a packet, the network is not anymore deterministic. Assigning the cells is consequently more complicated: a transmission opportunity has to be allocated even if a node has a very small probability to forward a packet. Finally, this approach considers only low traffic conditions: a packet is delivered to the border router before any other packet is generated in the network.

## 5.5. Conclusions and Open Problems

The approaches presented here were designed to work on centralized architectures with a sink aware of every characteristic of the network. The policies discussed here aim to create a communication plan to provide (i) high throughput, (ii) low delay and (iii) end-to-end guarantees, all three being requirements of most time critical applications. However, some challenges have yet to be addressed:

**Accuracy of information:** some works assume that the sink exploits directly an interference model (e.g. any pair of nodes less than  $k$  hop far away are assumed to interfere). In real applications, each node should collect the link quality of all its neighbors and send them to the sink to estimate the interference level. This increases the communication overhead, as more control packets will be transmitted.

To stay efficient, the sink should update its schedule when the network conditions change. Collecting the statistics generates a large overhead in the network. Besides, online measurement methods, cohabiting with data transmissions, have still to be proposed.

**Schedule changes** require to push the new schedule to all involved nodes, generating a large overhead. Livolant *et al.* [110] already investigated the cost of installing and updating a schedule using a central controller in a 6TiSCH network, by exploiting respectively a custom or a standard-based protocol. They highlighted the large overhead generated by this process, and proposed some optimizations to reduce the signaling cost.

Some work still remains to be done, particularly to maintain a consistent schedule even with packet losses. In particular, losses of some control packet may lead to a globally inconsistent schedule (e.g. the transmitter uses the previous schedule while the receiver uses the new one), possibly leading to network disconnections. Similarly, a scheduling algorithm which minimizes the amount of control packets to transmit has yet to be addressed.

**Lack of experimental evaluations:** most of the works presented here evaluate the scheduling algorithms only in simulated environment. Simulations provide a fast way to test an hypothesis before a time consuming implementation. However, simulations heavily depend on the accuracy of models [111], and tend to under-estimate the problems which may arise in practical scenarios [112].

The simulation results may diverge from the experimental results obtained in a testbed. For instance, Sempere *et al.* [57] evaluated experimentally TASA, and proved that packet losses are much more frequent than expected, leading to a poor end-to-end reliability.

## 6. Distributed Scheduling with Stationary Nodes

Because the network often changes continuously, we will here focus on algorithms that dynamically adjust the schedule to fulfill new communication requirements. The approaches are expected to deal efficiently with dynamic topologies and traffic. In most of the following approaches, each node is responsible to create a communication plan using only local information. Using a distributed architecture reduces communication overhead, as messages are transmitted only by the involved nodes. Guglielmo *et al.* [13] argue that distributed algorithms improve the scalability and the energy efficiency of the IIoT.

### 6.1. Stationary Nodes

The following algorithms are mainly designed to work on networks where all its nodes are static. In such scenarios, they achieve higher end-to-end reliability and low latency.

### 6.2. Rendez-Vous based Solutions

Some algorithms rely on a pseudo-random sequence to define autonomously a *rendez-vous* for any pair of nodes. A rendez-vous consists in a pair of instant and channel, derived from a pseudo-random sequence, so that the transmitter knows when and through which channel it can send its packets. The deafness problem (i.e. the receiver either sleeps or listens to another channel) must be avoided. The pseudo-random sequence initiates from a seed shared by both the receiver and the transmitter, so that no exchange of any explicit packet is needed to agree on a rendez-vous. Typically, many approaches use the MAC address of the receiver and/or the transmitter [113, 114, 78].

SSCH exploits a pseudo-random hopping sequence [113] so that any pair of nodes has overlapping channels to exchange data packets.

More recently, Chao *et al.* [114] proposed a similar approach, exploiting the concept of quorums, i.e. a collection of channel hopping sequences such that any pair of sequences share a non empty subset of common cells. These *Rendez-Vous* based MAC have been designed for IEEE802.11 mesh networks. Thus, they focused on improving the global network throughput by multiplexing the transmissions across orthogonal channels. We present here the solutions adapted for industrial low power networks.

Orchestra [78] was an adaptation of this rendez-vous based approach for low power networks. The algorithm relies on TSCH, and defines three types of slots:

**receiver oriented:** the slots are dedicated to receive packets. Since several neighbors may exist, they correspond to shared slots;

**transmitter oriented:** the slots are dedicated for its transmissions. The node considers itself as the unique transmitter, and identifies the slot as dedicated;

**broadcast:** common shared slots are in charge of transmitting broadcast traffic.

Orchestra proposes to use a hash of the MAC address of the node to determine which cells should be dedicated for transmission/reception in each slot. Since any neighbor can apply the same hash function, the neighboring schedule can be directly computed without exchanging any additional control information.

#### 6.2.1. Limitations

These approaches do not rely on specific signaling to decide which slots, and how many slots, must be used. Thus, the number of transmission opportunities does not depend on the volume of traffic forwarded through a given radio link. Since the different nodes often do not have the same amount of packets to forward, this approach both wastes energy (some nodes do not use all their cells) and suffers from a poor network capacity (the most loaded nodes may not have enough cells).

In addition, the slots are not allocated with the aim of minimizing the latency. Indeed, the slots are allocated pseudo-randomly, and a packet has to be buffered until a slot exists toward the next hop node. It both increases the end-to-end delay and the probability to have a buffer overflow.

Besides, these Rendez-Vous based solutions keep on relying on random access. Orchestra [78] defines transmitter-oriented slots, but since the cell is defined according to an hash of the address, collisions may still occur during *some* of the frames. The behavior is not deterministic, which could be prejudicial for industrial networks.

### 6.3. Local Lock-Based Algorithms

In this category, the protocols reserve a given cell in their vicinity. When a transmitter has to increase its bandwidth, it has to select a cell not used by any interfering node, and to flood locally its reservation. This cell is then *locked* for its personal usage.

DRAND [80] was originally designed to allocate the slots in Z-MAC [115]. More precisely, a node has to take the ownership of a timeslot in a distributed manner. The authors assume a 2-hop interference level, i.e. interference may arise only among transmitters at most 2 hops apart. The node sends a request to its 1-hop neighbors, starting a *round*, and the receivers have to acknowledge this reservation by replying with the list of timeslots already reserved by at least one neighbor. If one reservation is already pending, the request is rejected. Else, the node selects one timeslot and broadcasts its choice so that its neighbors *lock* the corresponding slot for the next reservations. Unfortunately, inconsistencies (collisions) may be present if the topology changes, or if interferences are under-estimated. Besides, the authors assume that a broadcast is reliable: inconsistent locks may arise if the request or the reply is not received by all the neighbors.

DiSCA *et al.* [34] is a distributed algorithm that tries to minimize the number of timeslots required to deliver all the packets to the sink. The algorithm proceeds iteratively, allocating at the step  $i$  the  $i^{th}$  transmission of each node. A node which has more packets to transmit is scheduled first in its neighborhood. Basically, the transmitter picks the first available pair of timeslot and channel offset and notifies all its conflicting nodes. Differently of DiSCA, a node executing DeAMON [79] overhears the control packets exchanged by its conflicting neighbors to adjust its own schedule.

#### 6.3.1. Limitation

Notifying the reservations to the interfering nodes is practically a complicated task. First, the set of conflicting nodes has to be established precisely, and estimating the interfering nodes is still an open challenge. Besides, no mechanism tries further to detect and solve the inconsistencies (collisions). Finally, a reservation in multicast is still an open problem in low power lossy networks. Dealing with unreliable links increases both the convergence delay, and the probability to create inconsistent schedules.

### 6.4. Hierarchical Lock-Based Algorithms

This family of protocols also reserves some cells for the personal usage of some nodes. However, to solve the conflicts, they use a hierarchical topology, so that nodes merge recursively the schedules of their children. Conflicts are then easier to solve, a node arbitrates the conflicts in the micro-schedule of its subtree.

DeTAS [81] is the decentralized version of TASA [51]. At first, each node requests bandwidth from its parent. The parent node computes the amount of packets that it will receive from its children and its

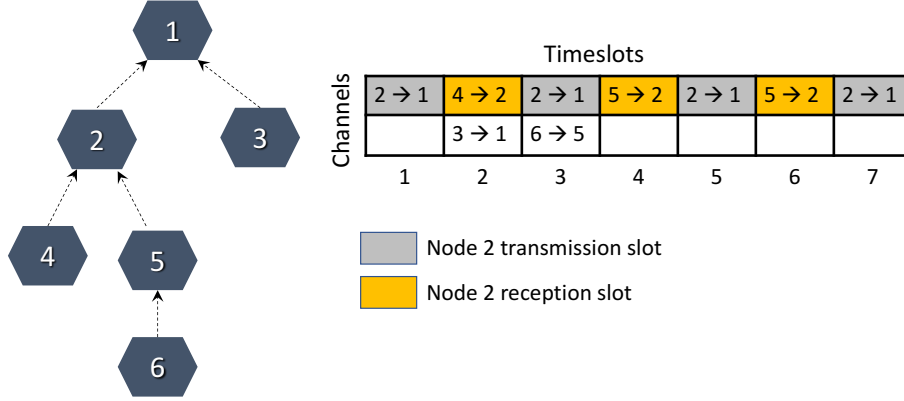


Figure 13: Alternating schedule built by DeTAS [81]. In this example, node 2 synchronizes its reception slots with the transmissions slots of the nodes 4 and 5.

own traffic requirements and forwards this information recursively until it reaches the sink. The sink starts the allocation by scheduling the timeslots to receive the aggregated traffic from each child. To reduce the end-to-end delay, and the buffer overflows, DeTAS schedules alternatively the reception/transmission slots along the path to the sink. In other words, a packet received during a slot, is transmitted to the parent in the next slot. Figure 13 shows an example of an alternating schedule in DeTAS.

In this example, each node (except node 1) generates one packet at the beginning of the slotframe. In the timeslot 2, node 2 receives the data from node 4, which forwards it to node 1 in timeslot 3. Next, node 2 receives from node 5 and forwards again on the timeslot 5. This alternating procedure continues until all the nodes in the network send their messages. Another alternating scheduling approach is proposed by Lee and Cho [116], using the tree depth as parameter to determine the alternating order.

Morell *et al.* [82] consider a scheduling algorithm for hierarchical networks, the Completely Fair Distributed Scheduler (CFDS). The leader of each cluster sorts its children according to their traffic demand. Each node reserves bandwidth using the Resource Reservation Protocol (RSVP) and Generalized Multi-Protocol Label Switching (GMPLS). CFDS works based on transmission cycles, where the most demanding node of each cluster is selected first by its parent. Each cycle begins after each leader schedules the transmission of the node selected in the previous cycle. The leader nodes also have their own transmission cycles, where they apply the same procedure with their own leaders. CFDS constructs the schedule alternating the transmission cycles similarly as DeTAS. To avoid interference from different clusters, each cluster selects randomly a channel offset to transmit their data on.

DIS\_TSCH [83] is another algorithm for hierarchical networks aiming to minimize the end-to-end delay. In a network executing DIS\_TSCH, each node knows its logic position in the tree topology. This information is obtained during the network tree building. The logic position includes the depth of the nodes and their relative positions in the cluster. Basing on this identification, the nodes construct their schedules independently. The algorithm schedules the leaves first, allocating consecutive timeslots for each routing path.

#### 6.4.1. Limitation

Because the solutions use an alternating mechanism (rx followed by a tx), if a packet is lost due to a poor radio link quality, all the subsequent slots scheduled on the parents nodes to forward this packet are wasted. This problem is even exacerbated when an over-provisioning strategy is used. Consequently, the parent nodes will unnecessarily turn their radio on looking for the start-of-frame that will be received during the next transmission attempts only.

Additionally, the solutions present the same limitation as the centralized approaches. When the network topology or traffic changes, the schedule procedure has to start over to still match the latency requirements.

The scheduler does not know a priori the traffic generated by each device. This information is collected in a distributed manner, aggregated along the tree rooted at the sink. However, considering a dynamic topology or traffic is not addressed in the original proposals. More specifically, mechanisms to notify the sink and to restart the schedule construction have yet to be proposed.

### 6.5. Reactive Collision Detection Approaches

Some solutions try to allocate randomly the cells, using only local information. The algorithm has then to detect the collisions reactively, when the same cell is allocated to two interfering radio links. Finally, it re-locates the colliding cells, selecting another cell which seems available for both the transmitter and the receiver. These solutions are also traffic reactive: new bandwidth is allocated on-demand, when more packets have to be forwarded.

Phung *et al.* [87] exploit a Reinforcement Learning approach to create a traffic-adaptive schedule. All the nodes act as an independent autonomous agent that receives feedbacks, as a result from its action, and learns from it. For every timeslot, each node executes a random action (transmit, receive or sleep) and receives a feedback. The feedback is positive when no collision occurs and when the node receives an acknowledgment. Upon positive feedback, the node will repeat this action during the slot in the next slotframe. Inversely, another action is selected randomly when the feedback is negative. This process continues until all the nodes are able to send their data successfully without idle slots. Intuitively, the algorithm can accommodate new demands, since it keeps on executing the corresponding actions. However, the convergence is not guaranteed.

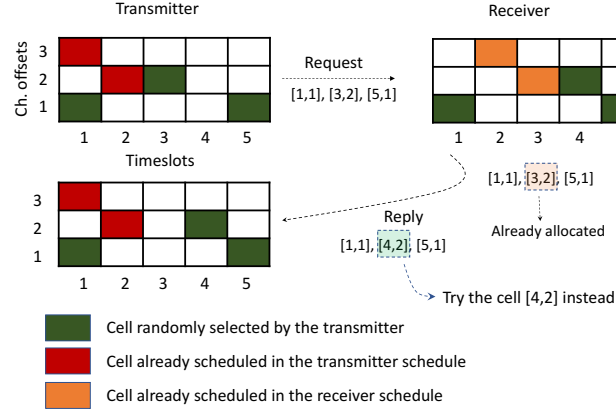
The Scheduling Function Zero (SF0) [117] is being standardized by the 6TiSCH Working Group. SF0 inherits functionalities from OTF [84] and adapts them slightly. The bandwidth estimation algorithm estimates the number of cells collecting the bandwidth requirement from each neighbor. It maintains the actual number of cells reserved with each neighbor equal to the amount of traffic to receive/transmit. When both quantities differ, SF0 engages a new reservation. SF0 also uses a hysteresis function to avoid oscillations in the allocation. In addition, SF0 keeps monitoring the used cells to detect when the packet delivery ratio drops below a given threshold, denoting a colliding cell. In such situation, the cells are reallocated randomly in the schedule table. Another policy to select which cells to allocate was proposed by Chang *et al.* [85]. The scheduling policy daisy-chains timeslots to accommodate new communication requirements along a multihop path.

SF0 allocates the cells randomly, without knowing which cells are used in the neighborhood. Municio *et al.* [86] demonstrated that this strategy works well for star technologies, but the number of collisions increases quickly in multihop topologies. They propose the DeBraS approach, where each node has to piggyback its local schedule when broadcasting its Enhanced Beacons. This knowledge helps to reduce the amount of collisions when a node has to select a new cell for a flow.

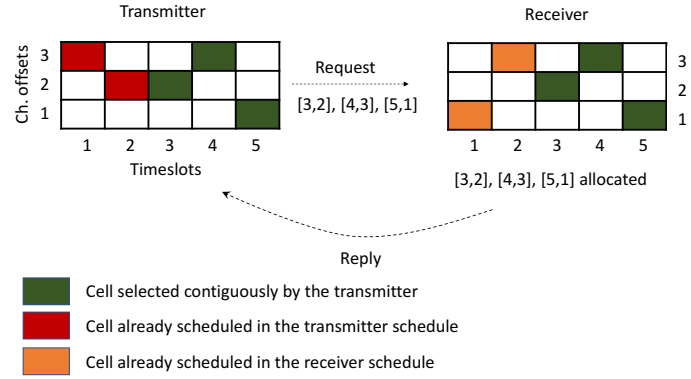
Hosni *et al.* focuses on the end-to-end delay constraint, to guarantee that a packet is delivered before the end of the slotframe, even if it has to be forwarded by several nodes [89]. The network is divided in strata, based on the sink's distance. Then, different portions of the schedule are allocated to each stratum. These portions are organized consecutively so that a node is sure to receive the packets (and their retransmissions) from its children before forwarding them to its parent.

Duy [90] *et al.* adopts a similar approach by dividing the slotframe into portions, but to balance more efficiently the load in the slotframe. Each portion is characterized by a density value, that yields the proportion of used slots. The Best Portion Selection (BPS) algorithm mitigates the probability of collisions by considering only the portion with the lowest density. The algorithms select a channel offset randomly, making sure that interfering nodes do not select the same value.

Theoleyre and Papadopoulos [88] proposed a distributed scheduling for 6TiSCH networks. This approach allocates bandwidth while guaranteeing flow isolation: each application has its own bandwidth reserved along the whole path. The bandwidth estimation algorithm works similarly to SF0, but it directly uses the current queue length of the node, and the Expected Transmission Count (ETX) of each allocated cell to compute the number of cells to allocate/deallocate. The authors compared two scheduling policies: either the timeslot and channel offsets are selected randomly (Figure 14a) or it tries to reduce the end-to-end delay (Figure 14b). In the latter case, the source tries to select the first available slots after the reception slots corresponding to the same flow.



(a) Random cells



(b) Contiguous cells

Figure 14: Example of channel/timeslot selection when scheduling new cells.

#### 6.5.1. Limitations

The works discussed here often ignore experiments with very dense topologies: very few large scale deployments tried to evaluate the scalability. Large densities reduce the number of available slots since many cells have already been reserved so far by interfering nodes. This increases the communication overhead, as the negotiation process may last longer until nodes agree on the cells to use, available for both sides.

#### 6.6. Conclusions and Open Problems

These approaches try to allocate distributively the cells, with only a local knowledge. When the traffic or the conditions change, the schedule is naturally modified, allocating more bandwidth with the next hop. Taking local decisions tends to improve the reactivity. However, we identified the following open problems:

**Centralized vs Distributed:** centralized vs. distributed solutions should be compared quantitatively in terms of performance, reliability and adaption to changes. In particular, that would allow to identify which schedule approach to use in which situation. Such quantitative comparison is still missing to *guide* the choice of a particular solution;

**No deterministic approach:** because these solutions are dynamic, it is quite hard to predict how they will perform in a given situation. There is no proof of convergence or stability. Using these approaches in real-time applications should be investigated very carefully.

## 7. Distributed Scheduling with Mobility Support

The deployment of industrial networks assumes static nodes and mobility is often not considered [118]. In fact, WirelessHart, ISA 100.11a and 802.15.4-TSCH standards do not give any special attention to mobility. However, mobility plays an increasingly important role for many applications [119].

Dealing with dynamic topologies is not sufficient: mobile devices create many topology changes, and the schedule becomes quickly suboptimal and inefficient. Similarly, a mobile device should discover a coordinator very fast to acquire the list of scheduled cells and avoid colliding with existing traffic. A mobile device must also *join* the network before being able to exchange packets with its neighbors. A Fast Attachment (FastA) has been proposed as an amendment for IEEE 802.15.4e [120], to reduce the number of control messages when a mobile device has to attach with the network.

To discover a valid neighbor, the two following strategies are available:

**active:** the node must send a solicitation in broadcast, and the receivers may reply with an Enhanced Beacon (EB). Unfortunately, the enquirer is not yet associated, and does not know the current schedule. Thus, its solicitation may create collisions with regular data traffic. Besides, the neighbors may listen to another channel (or may have turned their radio off), and may be deaf to its transmissions. Thus, the active method is not common for the low power slow channel hopping methods;

**passive:** the node has to wait for the reception of an Enhanced Beacon (EB). Because of the channel hopping property, the enquirer has to probe different channels at different moments to discover an existing node. To reduce the discovery delay, the node has to carefully select the channels [121].

Dealing with mobility in centralized architectures could be a very complex and challenging task, as the central manager should rebuild or update the schedule in every association or disassociation occurrence. Furthermore, the schedule updates have to be delivered reliably to each concerned device, before the mobile device changes its point of attachment, and consequently, the local schedule. WirelessHART and ISA 100.11a have been designed to support only centralized scheduling algorithms.

Distributed approaches tend rather to impact only locally the network when a node has to be inserted: the incriminated nodes have to negotiate locally for the new transmission opportunities. IEEE 802.15.4-TSCH lets the actual scheduling process unspecified, making it a favorable candidate for scenarios where mobility is a requirement.

### 7.1. Mobile Leaves

To reduce the cost, an operator may deploy a multihop topology of static low-power wireless routers, so that a collection of mobile devices may attach to one of these routers to exchange traffic. For instance, this infrastructure may be used to monitor a fleet of smart bikes [122]: a bike has to periodically compute its geographical location, and to push this information in the wireless infrastructure. Thus, only the last hop of the topology may comprise mobile devices.

#### 7.1.1. Best-Effort Allocation for Leaves

We can implement a differentiated medium access for the leaf and the relay nodes. While the relay nodes have dedicated cells for their transmissions, the mobile nodes can use some common shared cells to transmit their data frames. No allocation process is required, a mobile node has just to change the identity of the next hop to push its packets to the sink through another path.

6TiSCH minimal [123] relies on shared cells common to all the nodes, at the beginning of the slotframe. By default, 6TiSCH minimal recommends the usage of 1 shared cell, this parameter being tunable in the Enhanced Beacons. The shared cells are inserted in the schedule as soon as the node receives an EB and joins the network. Since a node may use the scheduled cell to transmit/receive all types of link-layer frames, any mobile node can send its data packets through this shared common cell.

Nidawi *et al.* [95] focused on the association procedure, engaged by a mobile node after it received an Enhanced Beacon. To reduce this collision probability, Nidawi *et al.* [95] modify the Enhanced Beacon to let the receiver know when the source of the EB will be available to receive a request (through a receiver-oriented dedicated cell). The same mechanism may be adapted also for data packet.



### 7.1.2. Limitations

Using shared cells for data transmission does not guarantee a minimum end-to-end reliability and does not offer any predictable performance. This indeterministic behavior restricts their adoption to simple scenarios with limited number of mobile nodes.

### 7.1.3. Infrastructure Pre-Allocated

The scheduler reserves bandwidth to all the static nodes to minimize the joining time of the mobile ones. When a mobile node changes its point of attachment, it uses this pre-allocated infrastructure to keep sending its packets to the sink with low latency.

Dezfouli *et al.* [91] tackle the real-time communication problem in a centralized architecture. In order to respect the time constraint, the algorithm reserves bandwidth to the mobile node itself and for each static node in all communications paths towards the sink. This avoids to rebuild the schedule when the mobile node changes its point of attachment and eliminates the rejoin process: the schedule has just to update the last hops. The authors also propose a reallocating policy for mobile nodes [92], to avoid blocked nodes. A node is blocked when it can not be scheduled to forward the flows from other nodes: it is already scheduled to transmit or receive a packet for a given flow.

To support real-time communications in mobile networks, the scheduling process can consider all the possible locations of each mobile device when the mobility can be accurately predicted or controlled [93]. The scheduling algorithm constructs a valid path for each mobile device along its whole trajectory. Since a single path is used at a time, the algorithm can merge the different paths in the schedule to avoid over-consuming the radio bandwidth.

Lee and Chung [94] proposed a scheduling algorithm for IEEE802.15.4-DSME networks. The work takes advantage of the hierarchical topology of DSME networks, so that each coordinator assigns the slots in its *cluster*. All the nodes must exchange control packets to maintain a knowledge about their queue length, the slots assigned, etc. When a node is mobile, it must acquire this knowledge from a neighboring stationary node. In particular, a node which has many packets to transmit, will advertise a longer queue length, and will have the right to *steal* some cells.

### 7.1.4. Limitations

In a multihop scenario, the stationary nodes forward the data from other stationary nodes. When a mobile node changes its point of attachment, the new communication demand must be re-allocated for each hop until the sink. Unfortunately, depending of the speed of the mobile node, the node may change its point of attachment before receiving a response from the infrastructure.

The stationary node must also ensure that the allocation of the new node will not impact the performance of existing flows. Otherwise, the scheduler would need to relocate some cells, and to push the new schedule. This process may take a long time, impacting the global performance.

If we consider a scenario where several nodes are moving, ensuring real-time guarantees will be even more complex. Preallocating bandwidth as proposed by Dezfouli *et al.* [91], may mitigate this problem. However, this approach reduces the amount of available slots to use for data transmissions, while keeping several idle slots. Furthermore, this preallocation is tuned for a limited number of mobile nodes.

## 7.2. All Mobile

A second and more complex scenario comprises only mobile nodes, without any fixed infrastructure. It corresponds to a mobile ad hoc network (MANET), with low power constraints. Bellavista *et al.* [124] envision a convergence between MANET and the IoT, for instance in vehicular networks.

### 7.2.1. Gossip Mechanism

The nodes continuously broadcast their schedule to spread their transmission times and of all its communicating neighbors. These messages instruct new nodes entering into the neighborhood for the first which slots to use for intra-neighborhood communications.

Tinka *et al.* [96] proposed a fully distributed scheduling algorithm for 802.15.4-TSCH networks to support a large number of mobile nodes. Nodes periodically broadcast advertisements specifying timeslots and channels that a neighbor node may use to transmit packets. In addition, the algorithm includes a gossip mechanism that disseminates the schedule to other nodes to make them aware about their 2-hop neighbors. As soon as a mobile node leaves the network, the corresponding slots are released for others. Mathew and Manuel [97] enhanced this algorithm to improve its reliability. They applied wavelet packet analysis to estimate the channel quality to avoid the most congested ones.

### 7.2.2. Limitations

Surprisingly, establishing a schedule for mobile networks has received little attention so far. The few existing works targeting fully mobile networks implicitly consider only simple scenarios without any real-time requirements (e.g. maximum delay, guaranteed delivery, etc.). Attending real-time requirements while dealing with frequent changes of topology represents a very challenging task. Mobile nodes create very dynamic topologies, and inconsistencies may quickly arise in different local schedules.

### 7.3. Conclusions and Open Problems

Several mechanisms have been proposed to enable mobility support in industrial applications. Most of the presented works rely on distributed approaches that combine mobile and stationary nodes. The mobile nodes use the infrastructure formed by stationary nodes to send their requests to the sink.

We identified the following open problems:

**Reliability:** the approaches do not focus on ensuring reliable delivery guarantees. To provide high-reliability while the nodes move and change their PoA constitutes a very challenging task. Efficient mechanisms have to be proposed to relocate some cells, minimizing the schedule updates to accept a new flow / handoff.

**radio link quality:** a mobile node has to estimate very quickly the link quality with the coordinator. In particular, external interferences may impact locally negatively the PER. A fast estimation of the link quality and the level of external interferences (for blacklisting) is still an open problem.

## 8. Conclusion

In this survey, we exposed the required preliminaries to understand the ins and outs of scheduling in slow channel hopping MAC for low power industrial wireless networks. We highlighted that real-time communication requirements, traffic models and radio link characteristics are key criteria to consider for scheduling algorithms in the IIoT.

During our literature review, we distinguished two categories of solutions, namely centralized (PCE-based) and distributed schedules. We classified them according to the key characteristics of the scenarios they are targeting (e.g. dynamic vs. static traffic, lossy versus ideal links, mobile vs. static topologies). The anticipated interest for frequency and time division multiple access is indeed confirmed by the vast number of papers that fall in this topic. Our prime objective was then to survey most of them while identifying some application scenarios and associated algorithms or guidelines to deploy them safely. We also listed systematically the characteristics supported by each scheduling algorithm (e.g. bursty links, multiradio, interference model) to guide the choice of the engineers / researchers in this field.

### 8.1. Open Challenges

While scheduling transmissions in slow channel hopping systems has attracted much attention in the last years, we still have to tackle several challenges, both for the scheduling algorithms themselves, and for the protocols (i.e. mechanisms) set-up to make such networks efficient.

### 8.1.1. Software-Defined Networking (SDN)

This approach seems promising to simplify the deployment of a large-scale network while centralizing the decisions to achieve better performances [125]. In a multichannel slow channel hopping network, it consists in creating a sink executing one of the aforementioned centralized algorithm. Unfortunately, we have still to tackle the following challenges:

**Interference Models:** most approaches assume that two nodes more than  $k$  hops apart do not interfere. A small  $k$  value under-estimates the level of interference, leading to collisions; inversely, a large  $k$  means a lower spectrum re-use. We must propose efficient mechanisms to detect the level of interference dynamically and to avoid the PCE to schedule interfering links simultaneously;

**Fault detection:** a radio link may be faulty, e.g. one node crashes, the link quality suddenly decreases. Thus, we still have to propose mechanisms to monitor the network [126], to identify the faulty nodes and links, and to modify the schedule accordingly;

**Scalability:** a single sink taking all the decisions may create a bottleneck in higher network densities. We must propose mechanisms to reduce the overload on a single entity. Yeganeh *et al.* [127] consider to deploy multiple central entities or to make the intermediate devices to take simple decisions.

### 8.1.2. Hybrid (Centralized and Distributed) Strategy

We are convinced that both approaches are complementary: the centralized scheduling algorithm allocates bandwidth to a given geographic area, while the distributed part tries to allocate the given timeslots locally. For instance, 6TiSCH introduces the concept of *chunks* for this purpose [22]: the sink allocates a chunk to a node which can then use this time-frequency block for its own purpose. However, we still have to address the following problems:

**Frequency re-use:** because a chunk is allocated to a node (and not a link), we have to over-estimate the interfering area: a timeslot can be used with *any* neighbor. Could we propose localized mechanisms to detect collisions and to re-allocate another cell of the chunk dynamically?

**Chunk allocation:** the scheduler has to allocate a set of chunks for each node so that they can manage locally their bandwidth. For this purpose, the scheduler has to define the size of each chunk and to designate the responsible nodes (the bandwidth for a radio link can be assigned by any of its end-node). It depends on the volume of traffic it has to forward, the radio link quality with its parent / children, the level of local external interference. To the best of our knowledge, this problem has not been addressed in the literature;

**Stability and Reactiveness:** a routing reconfiguration may change suddenly the volume of traffic to forward: the chunk of the concerned nodes may become insufficient. Reactive solutions must detect the situation, and patch the schedule to quickly converge to a legal state. A centralized approach would require a large delay (a chunk being allocated by the previous node, to be re-allocated to another one). Inversely, a distributed solution may transfer a part of a chunk from one node to a neighbor.

### 8.1.3. Efficiency in Realistic Conditions

Deployment environments of IIoT networks present some harsh conditions for radio embedded devices. Physical obstacles and interferences from concurrent transmissions negatively impact signal propagation and network reliability. Link quality therefore becomes unpredictable and may endanger the stability of upper layers (e.g. routing). In addition, foreseen scenarios in IIoT will require continuous updates of the built schedules while fault-tolerance will be expected from proposed solutions.

**Stability:** first, as detailed in Section 3.4, radio links are unstable. When a distributed routing protocol is used (e.g. RPL [128]), route changes may become frequent [129], disturbing the convergence of the scheduling algorithm. The induced challenge of ensuring stability would impose a careful cooperation

or co-design of both the schedule and the routing protocol, without compromising layer independency. For instance, a flow could be switched to another path only when enough bandwidth would have been pre-reserved. Stable performance should be guaranteed on top of a varying schedule, routing and radio topology;

**Iterative Continuous Optimization:** a network where all the nodes start to send data simultaneously is very unlikely. The scheduling algorithm must consequently allocate gradually the bandwidth, flow by flow. We may face to several challenges:

1. a gradual allocation may become quickly suboptimal, and would need to re-allocate some flows to e.g. optimize the frequency re-use. We have to explore the tradeoff between sub-optimality (colliding or unused cells), and the overhead for changing the schedule in all the nodes, which represents practically a challenging task [110].
2. scheduling algorithms should be able to *predict* the evolution in the requests, so that enough bandwidth is reserved for additional traffic in each node.

**Fault tolerant scheduling:** continuous updates will also be required upon node or link failures, while carefully considering some of the previously mentioned requirements (e.g. robustness to changes, dynamic bandwidth allocation). The capacity of scheduling algorithms to cope with node leaves and arrivals may become a matter of prime importance as more and more wireless devices will be deployed in increasingly hostile surroundings.

**Centralized versus distributed:** the employment of the scheduling families are quite orthogonal, which reinforces the common sense that the centralized algorithms performs worse then the distributed ones in a dynamic scenario and vice-versa for static scenarios. Although, no work comparing quantitatively both families in realistic conditions has been made so far.

#### 8.1.4. Mobility Support

The Internet of Mobile Things is now emerging [130], where smart objects can be moved or can move independently. Smart cities now integrate more and more mobile devices, for e.g. transport and logistic applications [131]. Similarly, the healthcare industry has to support users able to move independently while assuming ultra-reliable networks. Slow channel hopping schedules may help to provide such guarantees.

**Mobile devices:** in an infrastructure based topology, a collection of wireless routers forwards the traffic for and from mobile devices. While a strong attention has been given to the attachment delay, we have now to propose accurate scheduling methods to deal both with static and mobile nodes. To provide ultra-high reliability, cells have to be pre-reserved for the new flows. Besides, we should make the distinction between the relay nodes, and the mobile devices. While some of the cells are dedicated to relay the traffic among the wireless routers, the other ones can only be used from a mobile device to a wireless router. This differentiation would limit the collisions while making the system more scalable;

**Mobile sink:** a mobile sink helps to increase the network lifetime for delay tolerant data [132]. However, no scheduling algorithm focuses on allocating bandwidth to a collection of mobile sinks. If each sink follows a predictable trajectory, the algorithm should be able to provision and route data in advance, to increase the network capacity while optimizing energy savings.

## 9. References

### References

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of Things for Smart Cities, IEEE Internet of Things Journal 1 (1) (2014) 22–32. doi:10.1109/JIOT.2014.2306328.
- [2] J. Ye, B. Chen, Q. Liu, Y. Fang, A precision agriculture management system based on internet of things and webgis, in: International Conference on Geoinformatics, IEEE, 2013, pp. 1–5. doi:10.1109/Geoinformatics.2013.6626173.

- [3] M. Hermann, T. Pentek, B. Otto, Design Principles for Industrie 4.0 Scenarios, in: Proceedings of 49th Hawaii International Conference on System Sciences (HICSS), 2016, pp. 3928–3937. doi:[10.1109/HICSS.2016.488](https://doi.org/10.1109/HICSS.2016.488).
- [4] I. Khajenasiri, A. Estebasari, M. Verhelst, G. Gielen, A review on internet of things solutions for intelligent energy control in buildings for smart city applications, *Energy Procedia* 111 (2017) 770 – 779. doi:[10.1016/j.egypro.2017.03.239](https://doi.org/10.1016/j.egypro.2017.03.239).
- [5] Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications, ETSI TC ERM, TR 103 055, v1.1.1.
- [6] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, 2002, pp. 1567–1576 vol.3. doi:[10.1109/INFCOM.2002.1019408](https://doi.org/10.1109/INFCOM.2002.1019408).
- [7] J. Polastre, J. Hill, D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks, in: Proceedings of ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 95–107. doi:[10.1145/1031495.1031508](https://doi.org/10.1145/1031495.1031508).
- [8] IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4-2006 (2006).
- [9] N. Abdeddaim, F. Theoleyre, M. Heusse, A. Duda, Adaptive IEEE 802.15.4 MAC for Throughput and Energy Optimization, in: Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 2013, pp. 223–230. doi:[10.1109/DCOSS.2013.44](https://doi.org/10.1109/DCOSS.2013.44).
- [10] IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011) (April 2016).
- [11] T. Watteyne, A. Mehta, K. Pister, Reliability through frequency diversity: Why channel hopping makes sense, in: Proceedings of ACM Symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN), 2009, pp. 116–123.
- [12] G. Gaillard, D. Barthel, F. Theoleyre, F. Valois, Service Level Agreements for Wireless Sensor Networks: A WSN operator’s point of view, in: Proceedings of IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1–8. doi:[10.1109/NOMS.2014.6838261](https://doi.org/10.1109/NOMS.2014.6838261).
- [13] D. D. Guglielmo, S. Brienza, G. Anastasi, IEEE 802.15.4e: A survey, *Elsevier Computer Communications* 88. doi:[http://dx.doi.org/10.1016/j.comcom.2016.05.004](https://doi.org/http://dx.doi.org/10.1016/j.comcom.2016.05.004).
- [14] L. Kleinrock, F. Tobagi, Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics, *IEEE Transactions on Communications* 23 (12) (1975) 1400–1416. doi:[10.1109/TCOM.1975.1092768](https://doi.org/10.1109/TCOM.1975.1092768).
- [15] A. Koubaa, A. Cunha, M. Alves, A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks, in: Proceedings of 19th Euromicro Conference on Real-Time Systems (ECRTS), 2007, pp. 125–135. doi:[10.1109/ECRTS.2007.82](https://doi.org/10.1109/ECRTS.2007.82).
- [16] N. Abdeddaim, F. Theoleyre, F. Rousseau, A. Duda, Multi-Channel Cluster Tree for 802.15.4 Wireless Sensor Networks, in: Proceedings of IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2012, pp. 590–595. doi:[10.1109/PIMRC.2012.6362853](https://doi.org/10.1109/PIMRC.2012.6362853).
- [17] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control, in: Proceedings of Real-Time and Embedded Technology and Applications Symposium (RTAS), 2008, pp. 377–386. doi:[10.1109/RTAS.2008.15](https://doi.org/10.1109/RTAS.2008.15).
- [18] ANSI/ISA-100.11a-2011, Wireless systems for industrial automation: Process control and related applications, ISA (2011).
- [19] K. S. J. Pister, L. Doherty, TSMP: Time Synchronized Mesh Protocol, in: Proceedings of IASTED International Symposium on Distributed Sensor Networks (DSN), 2008, pp. 391–398.
- [20] T. Watteyne, S. Lanzisera, A. Mehta, K. S. J. Pister, Mitigating Multipath Fading through Channel Hopping in Wireless Sensor Networks, in: Proceedings of IEEE International Conference on Communications (ICC), 2010, pp. 1–5. doi:[10.1109/ICC.2010.5502548](https://doi.org/10.1109/ICC.2010.5502548).
- [21] A. Farrel, J.-P. Vasseur, J. Ash, A Path Computation Element (PCE)-Based Architecture, RFC 4665, IETF (2006).
- [22] IPv6 over the TSCH mode of IEEE 802.15.4e, <https://datatracker.ietf.org/wg/6tisch>.
- [23] P. Thubert, An architecture for ipv6 over the tsch mode of ieee 802.15.4, draft 12, IETF (August 2017).
- [24] W. y. Lee, K. i. Hwang, Y. A. Jeon, S. Choi, Distributed Fast Beacon Scheduling for Mesh Networks, in: Proceedings of IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems (MASS), 2011, pp. 727–732. doi:[10.1109/MASS.2011.79](https://doi.org/10.1109/MASS.2011.79).
- [25] M. Zou, J. Lu, F. Yang, M. Malaspina, F. Theoleyre, M.-Y. Wu, Distributed scheduling of enhanced beacons for ieee802.15.4-tsch body area networks, in: International Conference on Ad Hoc Networks and Wireless Networks (ADHOC-NOW), LNCS 9724, Springer, Lille, France, 2016.
- [26] D. Chen, M. Nixon, A. Mok, WirelessHART, Springer, 2010.
- [27] V. C. Gungor, G. P. Hancke, Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches, *IEEE Transactions on Industrial Electronics* 56 (10) (2009) 4258–4265.
- [28] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, H. Zhang, M. Gidlund, Assignment of Segmented Slots Enabling Reliable Real-Time Transmission in Industrial Wireless Sensor Networks, *IEEE Transactions on Industrial Electronics* 62 (6) (2015) 3966–3977. doi:[10.1109/TIE.2015.2402642](https://doi.org/10.1109/TIE.2015.2402642).
- [29] M. Zimmerling, P. Kumar, F. Ferrari, L. Mottola, L. Thiele, Energy-efficient Real-time Communication in Multi-hop Low-power Wireless Networks, Tech. Rep. 356, ETH Zurich, Laboratory TIK (Sep 2014).
- [30] I.-H. Hou, P. Kumar, Real-time communication over unreliable wireless links: a theory and its applications, *IEEE Wireless Communications* 19 (1) (2012) 48–59. doi:[10.1109/MWC.2012.6155876](https://doi.org/10.1109/MWC.2012.6155876).
- [31] A. Karnik, A. Iyer, C. Rosenberg, Throughput-Optimal Configuration of Fixed Wireless Networks, *IEEE/ACM Trans-*



- actions on Networking 16 (5) (2008) 1161–1174.
- [32] K. S. Prabh, T. F. Abdelzaher, On scheduling and real-time capacity of hexagonal wireless sensor networks, in: Euromicro Conference on Real-Time Systems (ECRTS), 2007, pp. 136–145. [doi:10.1109/ECRTS.2007.23](#).
  - [33] R. Soua, P. Minet, E. Livolant, MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks, in: Proceedings of IEEE 31st International Performance Computing and Communications Conference (IPCCC), 2012, pp. 91–100. [doi:10.1109/PCCC.2012.6407742](#).
  - [34] R. Soua, P. Minet, E. Livolant, DiSCA: A distributed scheduling for convergecast in multichannel wireless sensor networks, in: Proceedings of 2015 International Symposium on Integrated Network Management, 2015, pp. 156–164. [doi:10.1109/INM.2015.7140288](#).
  - [35] O. D. Incel, A. Ghosh, B. Krishnamachari, K. Chintalapudi, Fast data collection in tree-based wireless sensor networks, IEEE Transactions on Mobile Computing 11 (2012) 86–99. [doi:10.1109/TMC.2011.22](#).
  - [36] S. Han, X. Zhu, A. K. Mok, D. Chen, M. Nixon, Reliable and Real-Time Communication in Industrial Wireless Mesh Networks, in: Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011, pp. 3–12. [doi:10.1109/RTAS.2011.9](#).
  - [37] S. Wijetunge, U. Gunawardana, R. Liyanapathirana, Wireless sensor networks for structural health monitoring: Considerations for communication protocol design, in: Proceedings of 2010 International Conference on Telecommunications, 2010, pp. 694–699. [doi:10.1109/ICTEL.2010.5478798](#).
  - [38] L. Yu, N. Wang, X. Meng, Real-time forest fire detection with wireless sensor networks, in: Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing, Vol. 2, 2005, pp. 1214–1217. [doi:10.1109/WCNM.2005.1544272](#).
  - [39] A. Willig, K. Matheus, A. Wolisz, Wireless technology in industrial networks, Proceedings of the IEEE 93 (6) (2005) 1130–1151. [doi:10.1109/JPROC.2005.849717](#).
  - [40] D. S. J. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, Wirel. Netw. 11 (4) (2005) 419–434. [doi:10.1007/s11276-005-1766-z](#).  
URL <http://dx.doi.org/insis.bib.cnrs.fr/10.1007/s11276-005-1766-z>
  - [41] Z. Zaidi, T. Y. Tan, Y. Cheng, ETX could result in lower throughput, in: Proceedings of 18th International Conference on Computer Communications and Networks, 2009, pp. 1–6. [doi:10.1109/ICCCN.2009.5235384](#).
  - [42] M. Z. Zamalloa, B. Krishnamachari, An Analysis of Unreliability and Asymmetry in Low-power Wireless Links, ACM Transactions on Sensor Networks 3 (2) (2007) 1–34. [doi:10.1145/1240226.1240227](#).
  - [43] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, An Empirical Study of Low-power Wireless, ACM Transactions on Sensor Networks 6 (2) (2010) 16:1–16:49. [doi:10.1145/1689239.1689246](#).
  - [44] A. Cerpa, J. L. Wong, M. Potkonjak, D. Estrin, Temporal properties of low power wireless links: Modeling and implications on multi-hop routing, in: International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), ACM, 2005, pp. 414–425. [doi:10.1145/1062689.1062741](#).
  - [45] M. Eriksson, T. Olofsson, On long-term statistical dependences in channel gains for fixed wireless links in factories, IEEE Transactions on Communications 64 (7) (2016) 3078–3091. [doi:10.1109/TCOMM.2016.2563431](#).
  - [46] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, New York, NY, USA, 2005.
  - [47] P. D. Marco, G. Athanasiou, P.-V. Mekikis, C. Fischione, Mac-aware routing metrics for the internet of things, Computer Communications 74 (2016) 77 – 86, current and Future Architectures, Protocols, and Services for the Internet of Things. [doi:http://dx.doi.org/10.1016/j.comcom.2015.05.010](#).
  - [48] V. Kotsiou, G. Papadopoulos, P. Chatzimisios, F. Theoleyre, Label: Link-based adaptive blacklisting technique for 6tisch wireless industrial networks, in: International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), ACM, Miami, FL, US, 2017.
  - [49] A. A. K. S., K. Ovsthus, L. M. Kristensen., An industrial perspective on wireless sensor networks: A survey of requirements, protocols, and challenges, IEEE Communications Surveys Tutorials 16 (3) (2014) 1391–1412. [doi:10.1109/SURV.2014.012114.00058](#).
  - [50] Z. J. Haas, B. Liang, Ad hoc mobility management with uniform quorum systems, IEEE/ACM Transactions on Networking 7 (2) (1999) 228–240. [doi:10.1109/90.769770](#).
  - [51] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, G. Boggia, Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks, in: Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2012, pp. 327–332. [doi:10.1109/PIMRC.2012.6362805](#).
  - [52] M. Palattella, N. Accettura, On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH, IEEE Sensors Journal (2013) 1–21. [doi:10.1109/JSEN.2013.2266417](#).
  - [53] T. S. Chen, S. Y. Kuo, C. H. Kuo, Scheduling for Data Collection in Multi-hop IEEE 802.15.4e TSCH Networks, in: Proceedings of International Conference on Networking and Network Applications (NaNA), 2016, pp. 218–221. [doi:10.1109/NaNA.2016.23](#).
  - [54] P. H. Gomes, T. Watteyne, B. Krishnamachari, MABO-TSCH: Multihop and blacklist-based optimized time synchronized channel hopping, Transactions on Emerging Telecommunications Technologies. [doi:10.1002/ett.3223](#).
  - [55] M. Ojo, S. Giordano, An efficient centralized scheduling algorithm in iee 802.15.4e tsch networks, in: Conference on Standards for Communications and Networking (CSCN), IEEE, 2016. [doi:10.1109/CSCN.2016.7785164](#).
  - [56] Y. G. Kim, M. J. Lee, Scheduling multi-channel and multi-timeslot in time constrained wireless sensor networks via simulated annealing and particle swarm optimization, IEEE Communications Magazine 52 (1) (2014) 122–129. [doi:10.1109/MCOM.2014.6710073](#).
  - [57] V. Sempere-Payá, J. Silvestre-Blanes, D. Todolí, M. Valls, S. Santonja, Evaluation of TSCH scheduling implementations

- for real WSN applications, in: Proceedings of IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016, pp. 1–4. [doi:10.1109/ETFA.2016.7733642](#).
- [58] H. Zhang, P. Soldati, M. Johansson, Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks, in: Proceedings of IEEE / IFIP International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2009, pp. 1–8. [doi:10.1109/WIOPT.2009.5291641](#).
  - [59] H. Zhang, P. Soldati, M. Johansson, Performance Bounds and Latency-Optimal Scheduling for Convergecast in WirelessHART Networks, IEEE Transactions on Wireless Communications 12 (6) (2013) 2688–2696. [doi:10.1109/TWC.2013.050313.120543](#).
  - [60] H. Zhang, P. Soldati, M. Johansson, Efficient link scheduling and channel hopping for convergecast in wirelesshart networks, Tech. rep., School of Electrical Engineering, Royal Institute of Technology (KTH) (2009).
  - [61] F. Li, Z. Zhang, Z. Jia, L. Ju, Superframe Scheduling for Data Aggregation in WirelessHART Networks, in: Proceedings of IEEE International Symposium on Big Data Security on Cloud (BigDataSecurity), 2015, pp. 1540–1545. [doi:10.1109/HPCC-CSS-ICSS.2015.220](#).
  - [62] A. Elsts, X. Fafoutis, J. Pope, G. Oikonomou, R. Piechocki, I. Craddock, Scheduling high-rate unpredictable traffic in ieee 802.15.4 tsch networks, in: International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, 2017.
  - [63] F. Dobslaw, T. Zhang, M. Gidlund, End-to-End Reliability-aware Scheduling for Wireless Sensor Networks, IEEE Transactions on Industrial Informatics 12 (2) (2016) 758–767.
  - [64] G. Gaillard, D. Barthel, F. Theoleyre, F. Valois, High-Reliability Scheduling in Deterministic Wireless Multi-hop Networks, in: Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2016, pp. 1–6. [doi:10.1109/PIMRC.2016.7794839](#).
  - [65] G. Gaillard, D. Barthel, F. Theoleyre, F. Valois, Kausa: KPI-aware Scheduling Algorithm for Multi-flow in Multi-hop IoT Networks, in: Proceedings of International Conference on Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW), 2016, pp. 47–61. [doi:10.1007/978-3-319-40509-4\\_4](#).
  - [66] Y. Jin, P. Kulkarni, J. Wilcox, M. Sooriyabandara, A Centralized Scheduling Algorithm for IEEE 802.15.4e-TSCH based Industrial Low Power Wireless Networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2016, pp. 1725–1730. [doi:10.1109/WCNC.2016.7565002](#).
  - [67] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, K. Whitehouse, Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks, in: Proceedings of IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN), 2010, pp. 303–314. [doi:10.1145/1791212.1791248](#).
  - [68] W.-B. Pottner, H. Seidel, J. Brown, U. Roedig, L. Wolf, Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks, ACM Transactions on Sensor Networks 10 (3) (2014) 44:1–44:31. [doi:10.1145/2494528](#).
  - [69] K. Dang, J.-Z. Shen, L.-D. Dong, Y.-X. Xia, A graph route-based superframe scheduling scheme in wirelesshart mesh networks for high robustness, Wireless Personal Communications: An International Journal 71 (4) (2013) 2431–2444. [doi:10.1007/s11277-012-0946-2](#).
  - [70] S. Zhang, G. Zhang, A. Yan, Z. Xiang, T. Ma, A highly reliable link scheduling strategy for wirelesshart networks, in: Proceedings of 2013 International Conference on Advanced Technologies for Communications, 2013, pp. 39–43. [doi:10.1109/ATC.2013.6698073](#).
  - [71] T. Huynh, F. Theoleyre, W.-J. Hwang, On the interest of opportunistic anycast scheduling for wireless low power lossy networks, Computer Communications 104 (2017) 55 – 66. [doi:10.1016/j.comcom.2016.06.001](#).
  - [72] J. Lee, W. C. Jeong, B. C. Choi, A multi-channel timeslot scheduling algorithm for link recovery in wireless multi-hop sensor networks, in: 2016 International Conference on Information and Communication Technology Convergence (ICTC), 2016, pp. 871–876. [doi:10.1109/ICTC.2016.7763319](#).
  - [73] H. Wu, D. J. Lee, Robust qos scheduling using alternate path for recovery from link failures in ieee 802.15.4e, in: International Conference on Mobile Computing and Ubiquitous Networking (ICMU), 2014, pp. 99–100. [doi:10.1109/ICMU.2014.6799076](#).
  - [74] K. K. Chintalapudi, i-MAC - a MAC That Learns, in: International Conference on Information Processing in Sensor Networks (IPSN), ACM/IEEE, 2010, pp. 315–326. [doi:10.1145/1791212.1791250](#).
  - [75] K.-H. Choi, S.-H. Chung, A New Centralized Link Scheduling for 6TiSCH Wireless Industrial Networks, in: Proceedings of Springer International Conference on Internet of Things, Smart Spaces, and Next Generation Networks and Systems NEW2AN (ruSMART), 2016, pp. 360–371. [doi:10.1007/978-3-319-46301-8\\_30](#).
  - [76] S. Montero, J. Gozalvez, M. Sepulcre, G. Prieto, Impact of mobility on the management and performance of WirelessHART industrial communications, in: Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2012, pp. 1–4. [doi:10.1109/ETFA.2012.6489704](#).
  - [77] Q. Wang, K. Jaffrès-Runser, Y. Xu, J.-L. Scharbarg, A certifiable resource allocation for real-time multi-hop 6TiSCH wireless networks, in: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS, IEEE, 2017, pp. 1–9.
  - [78] S. Duquennoy, B. Al Nahas, O. Landsiedel, T. Watteyne, Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH, in: Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys), 2015, pp. 337–350. [doi:10.1145/2809695.2809714](#).
  - [79] A. Aijaz, U. Raza, DeAMON: A Decentralized Adaptive Multi-hop Scheduling Protocol for 6TiSCH Wireless Networks, IEEE Sensors Journal PP (99) (2017) 1–1. [doi:10.1109/JSEN.2017.2746183](#).
  - [80] I. Rhee, A. Warrier, J. Min, L. Xu, DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks, in: Proceedings of 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2006, pp. 190–201. [doi:10.1145/1132905.1132927](#).



- [81] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, M. Dohler, Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation, *IEEE Internet of Things Journal* 2 (6) (2015) 455–470. [arXiv:9605103](#), [doi:10.1109/JIOT.2015.2476915](#).
- [82] A. Morell, X. Vilajosana, J. L. Vicario, T. Watteyne, Label switching over IEEE802.15.4e networks, *Transactions on Emerging Telecommunications Technologies* 24 (5) (2013) 458–475. [doi:10.1002/ett.2650](#).
- [83] R.-H. Hwang, C.-C. Wang, W.-B. Wang, A distributed scheduling algorithm for {IEEE} 802.15.4e wireless sensor networks, *Computer Standards & Interfaces* 52 (2017) 63 – 70. [doi:https://doi.org/10.1016/j.csi.2017.01.003](#).
- [84] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, T. Engel, On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks, *IEEE Sensors Journal* 16 (2016) 550–560. [doi:10.1109/JSEN.2015.2480886](#).
- [85] T. Chang, T. Watteyne, Q. Wang, X. Vilajosana, LLSF: Low Latency Scheduling Function for 6TiSCH Networks, in: *Proceedings of International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2016, pp. 93–95. [doi:10.1109/DCOSS.2016.10](#).
- [86] E. Municio, S. Latré, Decentralized Broadcast-based Scheduling for Dense Multi-hop TSCH Networks, in: *Proceedings of ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, 2016, pp. 19–24. [doi:10.1145/2980137.2980143](#).
- [87] K.-H. Phung, B. Lemmens, M. Goossens, A. Nowe, L. Tran, K. Steenhaut, Schedule-based multi-channel communication in wireless sensor networks: A complete design and performance evaluation, *Elsevier Ad Hoc Networks* 26 (2015) 88 – 102. [doi:10.1016/j.adhoc.2014.11.008](#).
- [88] F. Theoleyre, G. Z. Papadopoulos, Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks, in: *Proceedings of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2016, pp. 102–110. [doi:10.1145/2988287.2989133](#).
- [89] I. Hosni, F. Théoleyre, N. Hamdi, Localized scheduling for end-to-end delay constrained low power lossy networks with 6tisch, in: *Symposium on Computers and Communication (ISCC)*, IEEE, 2016, pp. 507–512. [doi:10.1109/ISCC.2016.7543789](#).
- [90] T. P. Duy, T. Dinh, Y. Kim, Distributed cell selection for scheduling function in 6tisch networks, *Computer Standards & Interfaces* 53 (2017) 80 – 88. [doi:10.1016/j.csi.2017.03.008](#).
- [91] B. Dezfouli, M. Radi, O. Chipara, Real-time communication in low-power mobile wireless networks, in: *Proceedings of IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2016, pp. 680–686. [doi:10.1109/CCNC.2016.7444862](#).
- [92] B. Dezfouli, M. Radi, O. Chipara, Mobility-aware real-time scheduling for low-power wireless networks, in: *Proceedings of 35th Annual IEEE International Conference on Computer Communications (INFOCOM)*, 2016, pp. 1–9. [doi:10.1109/INFOCOM.2016.7524594](#).
- [93] B. Dezfouli, M. Radi, O. Chipara, REWIMO, *ACM Transactions on Sensor Networks (TOSN)* 13 (3) (2017) 1–42.
- [94] Y.-S. Lee, S.-H. Chung, An Efficient Distributed Scheduling Algorithm for Mobility Support in IEEE 802.15.4e DSME-Based Industrial Wireless Sensor Networks, *Hindawi International Journal of Distributed Sensor Networks* 2016 (2016) 1–14. [doi:10.1155/2016/9837625](#).
- [95] Y. Al-Nidawi, H. Yahya, A. H. Kemp, Tackling Mobility in Low Latency Deterministic Multihop IEEE 802.15.4e Sensor Network, *IEEE Sensors Journal* 16 (5) (2016) 1412–1427. [doi:10.1109/JSEN.2015.2500502](#).
- [96] A. Tinka, T. Watteyne, K. Pister, A decentralized scheduling algorithm for time synchronized channel hopping, in: *Proceedings of ICST Second International Conference on Ad Hoc Networks (ADHOCNETS)*, 2010, pp. 201–216. [doi:10.1007/978-3-642-17994-5\\_14](#).
- [97] V. Mathew, E. Manuel, Cognitive scheduling in tsch based mobile wsn using wavelet packet analysis, in: *Proceedings of International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 2015, pp. 1–5. [doi:10.1109/SPICES.2015.7091440](#).
- [98] J. Lee, T. Kwon, J. Song, Group Connectivity Model for Industrial Wireless Sensor Networks, *IEEE Transactions on Industrial Electronics* 57 (5) (2010) 1835–1844. [doi:10.1109/TIE.2009.2033089](#).
- [99] M. Nobre, I. Silva, L. A. Guedes, Routing and Scheduling Algorithms for WirelessHART Networks: A Survey, *MDPI Sensors* 15 (5) (2015) 9703. [doi:10.3390/s150509703](#).
- [100] S. C. Ergen, P. Varaiya, TDMA scheduling algorithms for wireless sensor networks, *Springer Wireless Networks* 16 (4) (2010) 985–997. [doi:10.1007/s11276-009-0183-0](#).
- [101] J. N. Tsitsiklis, K. Xu, On the Power of (Even a Little) Centralization in Distributed Processing, *ACM SIGMETRICS Performance Evaluation Review* 39 (1) (2011) 121–132. [doi:10.1145/2007116.2007131](#).
- [102] C. E.-A. Campbell, S. Khan, D. Singh, K.-K. Loo, Multi-Channel Multi-Radio Using 802.11 Based Media Access for Sink Nodes in Wireless Sensor Networks, *MDPI Sensors* 11 (5) (2011) 4917. [doi:10.3390/s110504917](#).
- [103] R. Musaloiu-E., A. Terzis, Minimising the Effect of WiFi Interference in 802.15.4 Wireless Sensor Networks, *Inderscience International Journal of Sensor Networks* 3 (1) (2008) 43–54. [doi:10.1504/IJSNET.2008.016461](#).
- [104] A. Gong, O. Landsiedel, P. Soldati, M. Johansson, Revisiting Multi-channel Communication to Mitigate Interference and Link Dynamics in Wireless Sensor Networks, in: *Proceedings of IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2012, pp. 186–193. [doi:10.1109/DCOSS.2012.15](#).
- [105] S. Y. Shin, Throughput analysis of IEEE 802.15.4 network under IEEE 802.11 network interference, *{AEU} - International Journal of Electronics and Communications* 67 (8) (2013) 686 – 689. [doi:10.1016/j.aeue.2013.02.007](#).
- [106] O. Chipara, C. Lu, T. C. Bailey, G.-C. Roman, Reliable Clinical Monitoring Using Wireless Sensor Networks: Experiences in a Step-down Hospital Unit, in: *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2010, pp. 155–168. [doi:10.1145/1869983.1869999](#).

- [107] G. Gao, H. Zhang, L. Li, A reliable multipath routing strategy for wireless mesh networks using subgraph routing, *Journal of Computational Information Systems* 5 (2013) 2120–2124.
- [108] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, P. Levis, The  $\beta$ -factor: Measuring Wireless Link Burstiness, in: *Proceedings of 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2008, pp. 29–42. doi:[10.1145/1460412.1460416](https://doi.org/10.1145/1460412.1460416).
- [109] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: A survey, *ACM Transactions on Sensor Networks (TOSN)* 8 (4) (2012) 34.
- [110] E. Livolant, P. Minet, T. Watteyne, The cost of installing a 6tisch schedule, in: *Proceedings of 15th International Conference on Ad Hoc Networks and Wireless*, 2016, pp. 17–31. doi:[10.1007/978-3-319-40509-4\\_2](https://doi.org/10.1007/978-3-319-40509-4_2).
- [111] A. Iyer, C. Rosenberg, A. Karnik, What is the right model for wireless channel interference?, *IEEE Transactions on Wireless Communications* 8 (5) (2009) 2662–2671. doi:[10.1109/TWC.2009.080720](https://doi.org/10.1109/TWC.2009.080720).
- [112] U. M. Colesanti, C. Crociani, A. Vitaletti, On the Accuracy of Omnet++ in the Wireless Sensor Networks Domain: Simulation vs. Testbed, in: *Proceedings of ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2007, pp. 25–31. doi:[10.1145/1298197.1298203](https://doi.org/10.1145/1298197.1298203).
- [113] P. Bahl, R. Chandra, J. Dunagan, SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-hoc Wireless Networks, in: *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2004, pp. 216–230. doi:[10.1145/1023720.1023742](https://doi.org/10.1145/1023720.1023742).
- [114] C. M. Chao, H. C. Tsai, A Channel-Hopping Multichannel MAC Protocol for Mobile Ad Hoc Networks, *IEEE Transactions on Vehicular Technology* 63 (9) (2014) 4464–4475. doi:[10.1109/TVT.2014.2314466](https://doi.org/10.1109/TVT.2014.2314466).
- [115] I. Rhee, A. Warrier, M. Aia, J. Min, M. L. Sichitiu, Z-MAC: A Hybrid MAC for Wireless Sensor Networks, *IEEE/ACM Transactions on Networking* 16 (3) (2008) 511–524. doi:[10.1109/TNET.2007.900704](https://doi.org/10.1109/TNET.2007.900704).
- [116] J.-H. Lee, S. H. Cho, Tree TDMA MAC Algorithm Using Time and Frequency Slot Allocations in Tree-Based WSNs, *Wireless Personal Communications* (2017) 1–23. doi:[10.1007/s11277-016-3938-9](https://doi.org/10.1007/s11277-016-3938-9).
- [117] D. Dujovne, L. A. Grieco, M. R. Palattella, N. Accettura, 6TiSCH 6top Scheduling Function Zero (SF0), Internet-Draft draft-ietf-6tisch-6top-sf0-02, Internet Engineering Task Force, work in Progress (Oct. 2016). URL <https://tools.ietf.org/html/draft-ietf-6tisch-6top-sf0-02>
- [118] P. Suriyachai, U. Roedig, A. Scott, A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks, *IEEE Communications Surveys & Tutorials* 14 (2) (2012) 240 – 264. doi:<http://dx.doi.org/10.1109/SURV.2011.020211.00036>.
- [119] R. Silva, J. S. Silva, F. Boavida, Infrastructure-supported mobility in wireless sensor networks - A case Study, in: *Proceedings of IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 1895–1900. doi:[10.1109/ICIT.2015.7125373](https://doi.org/10.1109/ICIT.2015.7125373).
- [120] IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Std 802.15.4e-2012 (Revision of IEEE Std 802.15.4-2011) (April 2012).
- [121] N. Karowski, A. C. Viana, A. Wolisz, Optimized Asynchronous Multichannel Discovery of IEEE 802.15.4-Based Wireless Personal Area Networks, *IEEE Transactions on Mobile Computing* 12 (10) (2013) 1972–1985. doi:[10.1109/TMC.2012.169](https://doi.org/10.1109/TMC.2012.169).
- [122] C. Kiefer, F. Behrendt, Smart e-bike monitoring system: real-time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles, *IET Intelligent Transport Systems* 10 (2) (2016) 79–88. doi:[10.1049/iet-its.2014.0251](https://doi.org/10.1049/iet-its.2014.0251).
- [123] X. Vilajosana and K. Pister, Minimal 6TiSCH Configuration, draft 17, IETF (novembre 2016).
- [124] P. Bellavista, G. Cardone, A. Corradi, L. Foschini, Convergence of MANET and WSN in IoT Urban Scenarios, *IEEE Sensors Journal* 13 (10) (2013) 3558–3567. doi:[10.1109/JSEN.2013.2272099](https://doi.org/10.1109/JSEN.2013.2272099).
- [125] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for SDN? Implementation challenges for software-defined networks, *IEEE Communications Magazine* 51 (7) (2013) 36–43. doi:[10.1109/MCOM.2013.6553676](https://doi.org/10.1109/MCOM.2013.6553676).
- [126] G. Gaillard, D. Barthel, F. Theoleyre, F. Valois, Monitoring KPIs in synchronized FTDMA multi-hop wireless networks, in: *Proceedings of Wireless Days (WD)*, 2016, pp. 1–6. doi:[10.1109/WD.2016.7461516](https://doi.org/10.1109/WD.2016.7461516).
- [127] S. H. Yeganeh, A. Tootoonchian, Y. Ganjali, On scalability of software-defined networking, *IEEE Communications Magazine* 51 (2) (2013) 136–141. doi:[10.1109/MCOM.2013.6461198](https://doi.org/10.1109/MCOM.2013.6461198).
- [128] T. Winter and P. Thubert and A. Brandt and J. Hui and R. Kelsey and P. Levis and K. Pister and R. Struik and JP. Vasseur and R. Alexander, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, rfc 6550, IETF (2012). doi:[10.17487/RFC6550](https://doi.org/10.17487/RFC6550).
- [129] O. Iova, F. Theoleyre, T. Noel, Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks, in: *Proceedings of IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013, pp. 2098–2102. doi:[10.1109/PIMRC.2013.6666490](https://doi.org/10.1109/PIMRC.2013.6666490).
- [130] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, F. J. d. S. e. Silva, The Mobile Hub concept: Enabling applications for the Internet of Mobile Things, in: *Proceedings of IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015, pp. 123–128. doi:[10.1109/PERCOMW.2015.7134005](https://doi.org/10.1109/PERCOMW.2015.7134005).
- [131] L. D. Xu, W. He, S. Li, Internet of Things in Industries: A Survey, *IEEE Transactions on Industrial Informatics* 10 (4) (2014) 2233–2243. doi:[10.1109/TII.2014.2300753](https://doi.org/10.1109/TII.2014.2300753).
- [132] Y. Yun, Y. Xia, Maximizing the Lifetime of Wireless Sensor Networks with Mobile Sink in Delay-Tolerant Applications, *IEEE Transactions on Mobile Computing* 9 (9) (2010) 1308–1318. doi:[10.1109/TMC.2010.76](https://doi.org/10.1109/TMC.2010.76).