Throughput Maximizing and Fair Scheduling Algorithms in Industrial Internet of Things Networks

Mike O. Ojo, *Student Member, IEEE,* Stefano Giordano, *Senior Member, IEEE,*
Davide Adami and Michele Pagano

***Abstract*—Time-Slotted Channel Hopping (TSCH) mode in the IEEE 802.15.4-2015 standard provides ultra-high reliability and ultra-low power consumption to sensor devices. The key feature of TSCH is the scheduling of time slots and frequencies, which falls outside the current standards. In this paper, we focus on throughput maximizing and max-min fair scheduling problems in a centralized TSCH networks. At first, a polynomial time algorithm for the throughput maximizing scheduling problem is proposed. We proceed to investigate and deliberate on some instances of the problem with their combinatorial properties. Secondly, a novel auction based scheduling algorithm that uses a first-price sealed-bid auction mechanism is presented for the throughput maximizing problem. Simulation results show that the proposed algorithm obtains a close throughput performance to the optimal one obtained through CPLEX with a much lower complexity. Moreover, we propose a novel heuristic for the max-min fair scheduling problem and demonstrate its performance through extensive simulations in terms of the total throughput and fairness varying the number of nodes, frequencies and antennas. Simulation results indicate the effectiveness of the proposed algorithm and its close performance to the optimal solution.**

***Index Terms*—Industrial Internet of Things, max-min fairness, TSCH, Auctions, IEEE 802.15.4-2015, Graph Theory, Centralized Scheduling, Resource Allocation**

I. Introduction

Recent developments have been undertaken in the field of Industrial Internet of Things (IIoT) to shape the need for high Quality of Service to the industries. Developments such as Wireless and IP-support for industrial networks have led to IIoT and have been demonstrated as an attractive alternative for communication. It is rapidly becoming the choice of communication to provide a viable and cost effective solution in a growing class of applications such as automotive systems, smart grids, factory automation etc [1], [2].

IIoT, a new communication paradigm, will revolutionize manufacturing and also drive growth in productivity across various types of applications. A common requirement across various IIoT applications is to deliver both low-power and wire-like reliability. This can be achieved through the work of various standardization bodies, which have made so much effort in proposing standards to support the development of IIoT and also address fast growing market needs. One of the leading standard in the IIoT is IEEE 802.15.4, which defined the physical layer (PHY) and the Medium Access Control (MAC) for low-rate, low-cost, low-power wireless personal area networks. Various industrial and automation technologies like ISA.100.11a [3], WirelessHART [4], that reside under the IIoT umbrella, have incorporated the IEEE 802.15.4 PHY layer components.

Reliability, scalability, unbounded latency are some of the limits of IEEE 802.15.4 MAC. The third version of the IEEE 802.15.4 standard, called IEEE 802.15.4-2015 [5], addresses its limitation by improving the previous PHY and MAC layers. TSCH is defined as one of the operating MAC modes in IEEE 802.15.4-2015 standard, which aims to address the strict requirements concerning timeliness and reliability especially in the industrial domain. This is done with the combination of channel hopping and time synchronization. Previous researches have shown that TSCH could achieve 99% end-to-end reliability with a radio duty cycle below 1% [6]. Motivated by the great potential of IIoT, both IEEE and IETF standardization bodies propose the architecture for the IIoT shown in figure 1.

In industrial environments where TSCH is now relevant, changing batteries can be dramatically expensive and difficult. However, with the advances in energy harvesting, the need for supporting better performing solutions and increasing throughput is also gaining momentum. One way of increasing throughput is with the use of multiple transceivers. For example, OpenMote-B[1] board is a device with multiple antennas. The key feature of TSCH is the scheduling of time slots and frequencies, which falls outside of the current standards. This paper formulates the scheduling problem as throughput maximization. The scheduling problem makes frequency and time slot allocations to the nodes, avoiding collision among them, and ensuring that communications between the nodes and the gateway are maintained. The scheduling model is aimed at achieving time and frequency assignments in a heterogeneous multi-channel environment where nodes are equipped with multiple antennas, thereby improving communication reliability.

However, this scheduling model may favour some nodes with good channel conditions while leading to the deprivation of some other nodes. Therefore, sustaining some notion of fairness is an essential criteria a scheduling model should address.

M.O. Ojo, S. Giordano, D. Adami and M. Pagano are with the Department of Information Engineering, University of Pisa, 56122, Pisa, Italy (e-mail: mike.ojo@ing.unipi.it; s.giordano@iet.unipi.it; davide.adami@cnit.it; m.pagano@iet.unipi.it)

[1]We introduce multiple transceiver due to the recent trend in sensor devices in introducing multiple antennas. For example http://www.openmote.com/

To achieve this aim, we proceed to formulate a max-min fair scheduling (MFS) problem that attains fairness by maximizing the node's throughput that has the minimum throughput amidst other nodes. A unique attribute of the proposed MFS problem is that it takes into account the throughput values of the nodes in the past. In this way, reduction in throughput experienced by the nodes in the past can be compensated for in subsequent slot frame periods.

The significance of this article can be summarized as follows:

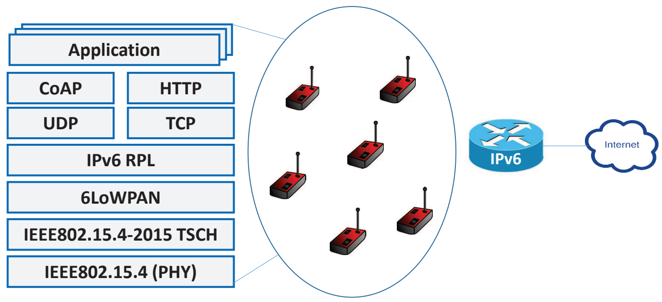

Fig. 1: IEEE/IETF Standardised Industrial IoT Architecture

1) We formulate an integer programming scheduling problem for throughput maximization in IIoT-TSCH networks.
2) For solving the throughput maximization scheduling problem, a polynomial time algorithm is proposed.
3) We examine and deliberate on some instances of the throughput maximization scheduling problem with their combinatorial properties.
4) We introduce a novel auction-based heuristic algorithm to address the throughput maximization scheduling problem. A first-price sealed-bid auction is used as the auction mechanism, and a time complexity analysis of the algorithm is also provided.
5) We proceed to formulate a MFS problem that provides throughput fairness. It considers the throughput historical information of the nodes in the past and use this information in the current scheduling decision.
6) We propose a heuristic approach based on greedy algorithm to address the MFS problem and the time complexity analysis of the heuristic algorithm is also given.
7) Finally, we evaluate both heuristic algorithms performance through simulations. We show that both algorithms yield a very close performance to the optimal one obtained through CPLEX [7].

The rest of this paper is organized as follows: Section II presents the related work. The system model and the problem formulation are presented in Section III. In Section IV, a polynomial-time algorithm for the throughput maximization problem is proposed. Then, in Section V, we present a novel scheduling approach based on auctioning. Section VI presents the MFS problem and the proposed heuristic algorithm. Section VII carries out the simulations and finally, in Section VIII, we conclude the paper with some final remarks.

II. TSCH in a NutShell and Related Works

TSCH is regarded as the keystone technology for the IIoT [8]. In IIoT-TSCH networks, communication occurs at well-defined times within a time slot, and it is orchestrated by a scheduler. The scheduler determines the time slot and frequency each node should transmit/receive data to/from its neighbours. Scheduling in IIoT-TSCH networks falls outside the current standards, and to construct a schedule is policy specific.

The TSCH scheduling task is an NP-hard problem [9]. Hence, most of the available scheduling algorithms use sub-optimal solutions, aiming at providing specific performance guarantees such as throughput, delay, fairness, just to mention a few. The centralized approaches rely on a central entity where the scheduler resides. Farias *et al.* [10] proposed a queue-based algorithm for the path computation element to increase the reliability in industrial scenarios. The authors in [11] proposed a cross-layer low-latency topology management and TSCH scheduling technique that provides a very high timeslot utilization to minimize communication latency. The works in [10], [11] does not concern itself with throughput – a challenge we aim to handle in the paper. Other centralized scheduling approaches in IIoT-TSCH networks worth mentioning are [12], which maximizes the energy efficiency, and [13] addressing latency issues.

Under distributed algorithms, DeTAS [14] relies on exchanging traffic information based on neighbour-to-neighbour signalling. Jung *et al.* [15] proposed a scheduling algorithm to minimize the energy consumption while guaranteeing reliability. It works adaptively with traffic intensity and reliability requirements. Orchestra [16] is an autonomous scheduling mechanism, where each node independently builds its own schedule without any negotiation. While orchestra is flexible and able to achieve 99.99% packet delivery, it is unable to address bursty traffic. Further works in the literature have addressed the issue of bursty traffic [17]. In this paper, we focus on a centralized approach by providing a more compact scheduling model by accomplishing many tasks at the same time, since a centralized method has been demonstrated to be more efficient in practice [18]. The scheduling model is able to provide time, frequency and data rate allocation in a multi-node, multi-channel environment where nodes are equipped with multiple antennas for data transmission. The use of multiple transceivers can improve link quality and throughput, ensuring communication reliability [19]. This paper is designed on maximizing the total throughput in an interference-aware system by proposing a polynomial time algorithm and an auction-based mechanism.

In the scheduling algorithms described above [10–17], the notion of fairness fails to exist. Several works in the literature have addressed fairness in the wireless networks domain [20–22]. In [20], the authors introduced various distributed algorithms that allows each node to calculate its max-min link share in a wireless ad-hoc network. In [21], the authors proposed a distributed MAC algorithm to share network resources in a fairly way in 802.11-based networks. Shi *et al.* [22] provided a fairness-aware scheduling algorithm to address the penalty mechanism in balancing reliability and fairness. However, none of these works takes into account the unique features of TSCH concept, therefore, they are unsuitable for implementation in IIoT-TSCH networks. This paper addresses time and frequency resource pair to be shared fairly among the nodes in the IIoT-TSCH networks. Unlike most works [20–22], our emphasis in this article is on IIoT-TSCH networks. We present a fair scheduling model that takes into account the joint provision of throughput and temporal fairness by considering the throughput values in the past and the use of this information in the current scheduling decision.

A typical scenario in IIoT applications can be applied to smart grids, where the link throughput between smart meters and aggregators can be maximized under specific constraints. The meters can be equipped with directional antennas, while transmitting at fixed power and the transmissions are coordinated to avoid packet collisions. It can also be used in industrial process monitoring applications to take advantage of flexible data rate allocation. For example, in the mining industry [23] require several sensing points such as speed monitors, belt conveyor monitors etc. When the dynamics of the belt changes, it triggers a spike in data generation until a new equilibrium point is reached.

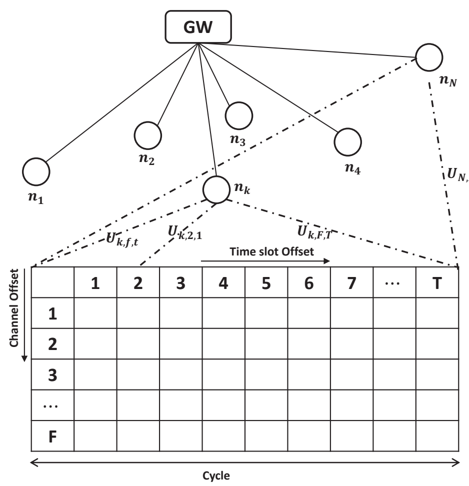

Fig. 2: TSCH slot-channel matrix where each n_k maintains a link with the gateway (GW) for both frequency f and time slot t, where $k \in \{1, ..., N\}$; $f \in \{1, ..., F\}$, $t \in \{1, ..., T\}$

III. System Model and Problem Formulation

Consider an IIoT-TSCH network consisting of a gateway (GW) and several nodes as shown in figure 2. We introduce a centralized architecture, where the gateway is responsible for the coordination and the management of the nodes in its area. The scheduling of time slots and frequencies are determined by the scheduler, which resides at the gateway.

The network can be modelled by a graph $G = (V, E)$, where each node $n_k \in V$ represents a device in the network. There is a link $e = (u, v) \in E$ if a common channel exists between u and v. We are given a set of nodes $k \in \mathcal{N}$, frequencies $f \in \mathcal{F}$ and time slots $t \in \mathcal{T}$. In other words, $\mathcal{N} = \{1, ..., N\}, \mathcal{F} = \{1, ..., F\}, \mathcal{T} = \{1, ..., T\}$. $|V| = N$ is the total number of nodes in the network. Each node is equipped with a radio transmitting at a fixed transmission power. We assume that each device is equipped with more than one transceiver and denote by a_k the number of transceivers at node k.

The system operates in a frame-based fashion, where all the nodes are synchronized. At the beginning of the frame, each node n_k calculates the transmission capacity of the channel for each available frequency, and transmits this information to the gateway in addition to the number of packets in its buffer (Q_k). The gateway builds a matrix $U = [U_{k,f,t}]$, where $U_{k,f,t}$ represents the maximum number of packets node k can send using frequency f in time slot t. Let $C_{k,f,t}$ denote the Shannon's capacity, which is a function of the signal-to-noise ratio $SNR_{k,f,t}$ and represents the theoretical upper bound. In the calculation of $C_{k,f,t}$, only the background noise is considered as we focus on an orthogonal frequency assignment. The number of packets that can be sent during a slot frame, $M_{k,f,t}$, equals $C_{k,f,t}T$, where T is denoted as the frame duration. Accordingly, $U_{k,f,t}$ becomes $min(M_{k,f,t}, Q_k)$. Upon collection of all state information, such as topology and traffic generated by each node, the scheduler decides on the frequency assignment with the goal of maximizing the throughput.
Introducing the binary variable $X_{k,f,t}$ denoted as:

$$X_{k,f,t} = \begin{cases} 1 & \text{if } n_k \text{ transmits using} \\ & \text{frequency } f \text{ in time slot } t; \\ & \\ 0 & \text{otherwise;} \end{cases}$$

where $X_{k,f,t}$ is a function of the information available to n_k and the gateway. Thus, we calculate the total throughput in TSCH network as

$$C = \sum_{k=1}^{N} \sum_{f=1}^{F} \sum_{t=1}^{T} U_{k,f,t} X_{k,f,t} \tag{1}$$

The derivation of the formula $U_{k,f,t}$ takes into account interference and spectrum availabilities. For instance, if a node k is very close to a node i that uses frequency f in time slot t, then node k cannot use frequency f in the same time slot, i.e., $U_{k,f,t} = 0$. $U_{k,f,t}$ value quantifies the slot availabilities (i.e., the higher the $U_{k,f,t}$ value is, the higher the data rate node k can have if it uses frequency f in slot t).

$U_{k,f,t}$ values are the input variables for the optimization problem in this article. We assume that the network conditions, (i.e., channel fading coefficients, etc.) are considered to be small to have any influence on $U_{k,f,t}$ values for T time slots duration in a IIoT-TSCH cell. A slot frame period consists of T time slots. In the following, $U_{k,f}$ is used instead of $U_{k,f,t}$ since $U_{k,f,t}$ value for node k and frequency f are constant in a slot frame period.

Given the $U_{k,f}$ values determined in the first phase, we introduce the problem formulation as a binary integer linear programming in order to maximize the total throughput as shown as follows:

$$\textbf{P1:} \quad \max \sum_{k=1}^{N} \sum_{f=1}^{F} \sum_{t=1}^{T} U_{k,f} X_{k,f,t} \tag{2}$$

s.t

$$\sum_{f=1}^{F}\sum_{t=1}^{T} X_{k,f,t} \geq 1; \quad \forall k \in \mathcal{N} \tag{3}$$

$$\sum_{k=1}^{N} X_{k,f,t} \leq 1 \quad \forall k \in \mathcal{N}; \forall t \in \mathcal{T} \tag{4}$$

$$\sum_{f} X_{k,f,t} \leq a_k \quad \forall k \in \mathcal{N}; \forall t \in \mathcal{T} \tag{5}$$

$$X_{k,f,t} \in \{0,1\}; \quad \forall k \in \mathcal{N}; \forall f \in \mathcal{F}; \forall t \in \mathcal{T} \tag{6}$$

In **P1**, the objective function in (2) shows how the total throughput of all the nodes in the network is maximized and executed by the gateway. At least one time slot is allocated to each node as indicated by constraint (3). Without this constraints (3), some nodes with bad channel conditions may end up not sending a packet for a long time. This constraints guarantees each node to send a packet. The constraint in (4) is used to prevent collision among the nodes, by ensuring that at most one node can transmit in a certain time slot and frequency. Node k cannot transmit more frequencies than the number of antennas at the same time as indicated by constraint (5) because a node antenna can only tune to no more than one frequency at once, and finally constraint in (6) indicates a binary decision variable. Note that, the problem formulation is valid for both single antenna $i.e., a_k = 1 \, \forall k \in \mathcal{N}$ and multiple antennas $i.e., a_k > 1 \, \forall k \in \mathcal{N}$. The use of multiple antennas provides the need for simultaneous data transmission using different frequencies.

In the simulations part of this article, we consider the buffers of the nodes to be continuously backlogged; i.e., there are always enough packets to be transmitted. This condition is relevant in order to adequately access our proposed scheduling algorithms by avoiding any possible traffic fluctuations.

IV. A POLYNOMIAL-TIME ALGORITHM FOR THE THROUGHPUT MAXIMIZATION PROBLEM

A. Preliminaries

Let Δ be a maximization problem formulation and let γ be a positive real number, $\gamma \geq 1$. A feasible solution s of an instance I of Δ is a γ factor approximation if $O_\Delta(I,s)$ is at least a factor γ of $O_{\Delta^*}(I)$ of I, i.e., $O_\Delta(I,s) \geq (O_{\Delta^*}(I)/\gamma)$, where $O_\Delta(I,s)$ denotes the objective function value of solution s of instance I and O_{Δ^*} denotes the objective function value of an optimal solution of instance I.

The essential concept is to lessen the problem variables in a way it will be computationally more adequate and easy to solve. On this basis, an equivalent simpler problem formulation is proposed, which will be demonstrated as follows. Assume Δ and Δ' are the optimization problems for which one problem instance can be mapped to another problem instance in a way that the close optimal solution of Δ' can be transformed back to yield close optimal solution of Δ. If there is an approximation algorithm for Δ' and an adequate approximation-preserving reduction from problem Δ to problem Δ', by composition, an approximation algorithm for Δ also exists.

With the preliminaries explained, we denote by Δ the optimization problem involving only $X_{k,f,t}$ variables in the form of $\sum_{t=1}^{T} X_{k,f,t}$. We denote Δ' to be the optimization problem obtained from Δ by substituting

$$Y_{k,f} = \sum_{t=1}^{T} X_{k,f,t} \quad \forall k \in \mathcal{N}; \forall f \in \mathcal{F} \tag{7}$$

in O_Δ except constraints (4) - (6). Constraints (4) - (6) are replaced by the following constraints as displayed below.

$$\sum_{k=1}^{N} Y_{k,f} \leq T; \quad \forall f \in \mathcal{F} \tag{8}$$

$$\sum_{f=1}^{F} Y_{k,f} \leq a_k T; \quad \forall k \in \mathcal{N} \tag{9}$$

$$Y_{k,f} \in \{0, \mathcal{T}\}; \quad \forall k \in \mathcal{N}, \forall f \in \mathcal{F} \tag{10}$$

Δ and Δ' are two similar optimization problems. It is adequate to show that $O_{\Delta'}(Y) = O_\Delta(X)$ in polynomial time. This holds for an optimum solution X^* which makes both problems to have the same optimal solution. With a solution X of Δ, the solution Y defined in (7) clearly satisfies all constraints except (4) - (6) by substituting them with the constraints below respectively. These constraints are obtained by summing up T inequalities of (4) - (6).

Under the approximation-preserving reduction conditions, Δ and Δ' are equivalent. Given Δ', the throughput maximization scheduling problem is proportionate to determining the vector Y with the following ILP formulation.

$$\textbf{P2:} \quad \max \sum_{k=1}^{N}\sum_{f=1}^{F} U_{k,f} Y_{k,f} \quad \forall k \in \mathcal{N}; \forall f \in \mathcal{F}; \tag{11}$$

s.t

$$\sum_{f=1}^{F} Y_{k,f} \geq 1; \quad \forall k \in \mathcal{N} \tag{12}$$

(8), (9) and (10)

It can be seen that the problem formulation (objective function and constraints) of Δ' has less variables than Δ formulation. With this, it is computationally easy to solve with the use of some optimization and modelling tool such as ILOG-CPLEX [7] to find close optimal solutions. To describe the polynomial time algorithm, we begin by providing some preliminaries concerning matchings as defined in [24].

Consider a graph $G = (V, E)$ with a weight function w, where w: $E \rightarrow \mathbb{R}$ is a weight (or cost) function defined on its edges, and also denoting H to be a function of natural numbers with each vertex in V. Let $\delta_G(v)$ be the set of incident edges of v in G, i.e., $\delta_G(v) = \{e \in E | v \in e\}$, and $d_G(v) = |\delta_G(v)|$ denotes the degree of vertex v, which is equal to the number of incident to v in graph G. Given a function h: $E \rightarrow \mathbb{N}$, a H-matching is such that for $v \in V$, $\sum_{e \in \delta_G(v)} h(e)$ is in the interval of $H(v)$.

If $H(v) = \{0, 1\}$ for each vertex v, then a H-matching is in fact a *matching*, i.e., a set of vertex-disjoint edges. A *perfect matching* is a H-matching such that $H(v) = 1$ for each $v \in V$. A *maximum* H-matching is a H-matching h such that $\sum_{e \in E} h(e)w(e)$ is maximized. A polynomial time solution exists for a maximum H-matching [25]. A sub-graph B of G is equivalent to H-matching h. In the rest of the paper, B will also be called a H-matching.

In the sequel, we build a bipartite graph $G = (U, V, E)$ from vector Y such that $U = \{u_1, u_2, ..., u_N\}$, $V = \{v_1, v_2, ...v_F\}$ and $E = \{(u, v)|u \in U, v \in V\}$ are the edges of G. In our case, we have $Y_{k,f}$ parallel edges associating $u_k \in U$ and $v_f \in V$. The degree of vertex $v_f \in V$ is at most T by constraint (8), and the degree of vertex $u_k \in U$ is at most a_kT by constraint (9). Introducing another bipartite graph $G' = (U', V', E')$ obtained from G by substituting each vertex $u_k \in U$ with a_k vertices in U' and dividing the at most a_kT edges adjacent to u_k to these new vertices in a manner that each vertex receives at most T edges. This ensures that the degree of each vertex G' is at most T. In addition, each vertex $v_f \in V$ is the same in V'. To ensure perfect matching, we introduce another bipartite graph $G^* = (U^*, V^*, E^*)$ attained from G' by adding $||U'| - |V'||$ dummy vertices to either U' or V' such that $|U''| = |V''|$, and also adding dummy edges as long as we have vertices within the graph with a degree less than T. Such graph using Hungarian Algorithm [26] can be found in polynomial time as demonstrated by Tutte [27]. If we remove the perfect matching from G^*, we remain with a $(T - 1)$ regular bipartite graph. Applying this inductively, G^* is partitioned into T perfect matchings $B_1^*, B_2^*, ..., B_T^*$. Moreover, we obtain T matchings $B_1', B_2', ..., B_T'$ of G' by eliminating all the dummy vertices and edges of G^* from the perfect matchings. In each matching B_t', where $t \in \mathcal{T}$, a_k vertices of U' is contracted back to the node u_k for every $k \in \mathcal{N}$ to get T bipartite graphs $B_1, B_2, ..., B_T$, where each vertex $u_k \in U$ has degree at most a_k and each node $v_f \in V$ has degree at most 1. We denote $X_{k,f,t} = 1$ if u_k is adjacent to v_f in B_t, and 0 otherwise. With this, it can be concluded that X satisfies constraints (4)–(6).

B. Algorithms for the Throughput Maximization Scheduling Problem

Theorem 1. There is a polynomial time algorithm to address the throughput maximization scheduling problem.

Proof. We show that Algorithm 1 is the optimal algorithm for throughput maximization scheduling problem. In Algorithm 1, we can see that the lower bound 1 and upper bound a_kT of $H(u_k)$ in line 7 is equivalent to constraints (12) and (9), respectively. Moreover, line 8 which denotes the upper bound T of $H(v_f)$ corresponds to constraint (8). As G is a bipartite graph, step 9 is also solvable in polynomial time. □

Algorithm 1 Throughput Maximization Scheduler *(Throughput-maximum)*

Require: $\mathcal{F}$, $\mathcal{N}$, a_k, $k \in \mathcal{N}$
Ensure: $Y_{k,f}$ values $\forall k \in \mathcal{N}$; $\forall f \in \mathcal{F}$

1: $E = \emptyset$
2: Build an edge weighted bipartite graph $G = (U, V, E)$ as follows:
3: For each node $k \in \mathcal{N}$ add a vertex u_k to U.
4: For each frequency $f \in \mathcal{F}$ add a vertex v_f to V.
5: For each pair of vertices $u_k \in U$ and $v_f \in V$, add the edge $\{u_k, v_f\}$ to E with $U_{k,f}$
6: Define the function H in G
7: $H(u_k)$ = $[1, a_kT]$ $\forall k \in \mathcal{N}$;
8: $H(v_f)$ = $[0, T]$ $\forall f \in \mathcal{F}$
9: Find a maximum weighted B of G
10: For all $k \in \mathcal{N}$ and $f \in \mathcal{F}$, we denote by $Y_{k,f}$ the number of edges between vertices u_k and v_f in B

We deliberate on the combinatorial properties of some instances of the throughput maximization scheduling problem.
Instance 1: If constraint (12) is ignored and also assuming that the number of antennas for a node k is one i.e., $a_k = 1$ $\forall k \in \mathcal{N}$, then a solution exists to the maximum weighted bipartite matching (MWBM) problem between the nodes ($\forall k \in \mathcal{N}$) and the frequencies ($\forall f \in \mathcal{F}$). The matching result can be applied to every time slot $\forall t \in \mathcal{T}$. Once the mapping process is done, there are various graphical approaches for solving the MWBM problem. Hungarian Algorithm [26] is often used because of its simplicity, its efficiency and its computational complexity limited to $O(n^3)$.
Instance 2: If constraints (9) and (12) are ignored in Δ problem formulation, then an optimal solution is obtained when a node with maximum $U_{k,f}$ is allocated a frequency f in every time slot. Specifically, $X_{k,f,t} = 1$ when $k = argmax_k U_{k,f}$, and 0 otherwise, $\forall t \in \mathcal{T}$, $\forall f \in \mathcal{F}$. Ignoring constraint (9) corresponds to having $a_k \geq F$, $\forall k \in \mathcal{N}$.

V. Auction Based Mechanism Scheduler

In this section, we propose an auction based scheduling algorithm for the throughput maximization problem. At first, we briefly introduce the concept of auction theory, then we propose an auction based scheduler and also discuss about its computational complexity.

A. Auction Theory

Auction theory, originally developed in economics, is a branch of game theory that has been applied to solve various problems in engineering such as the network resource allocation. An auction mechanism involves buyers and sellers, where buyers submit bids for purchasing commodities, and sellers are asked to sell commodities. We have four primary auction types described in the literatures: the ascending-bid auctions (English auctions), descending-bid auctions (Dutch auctions), first-price sealed-bid auctions, and the second-price sealed-bid auctions, also called Vickrey auctions [28].

B. Auction Theory Based Scheduler

In this section, an auction based first-price sealed-bid mechanism is proposed to solve for the throughput maximization scheduling problem formulated in (2) - (6). Our motivation for

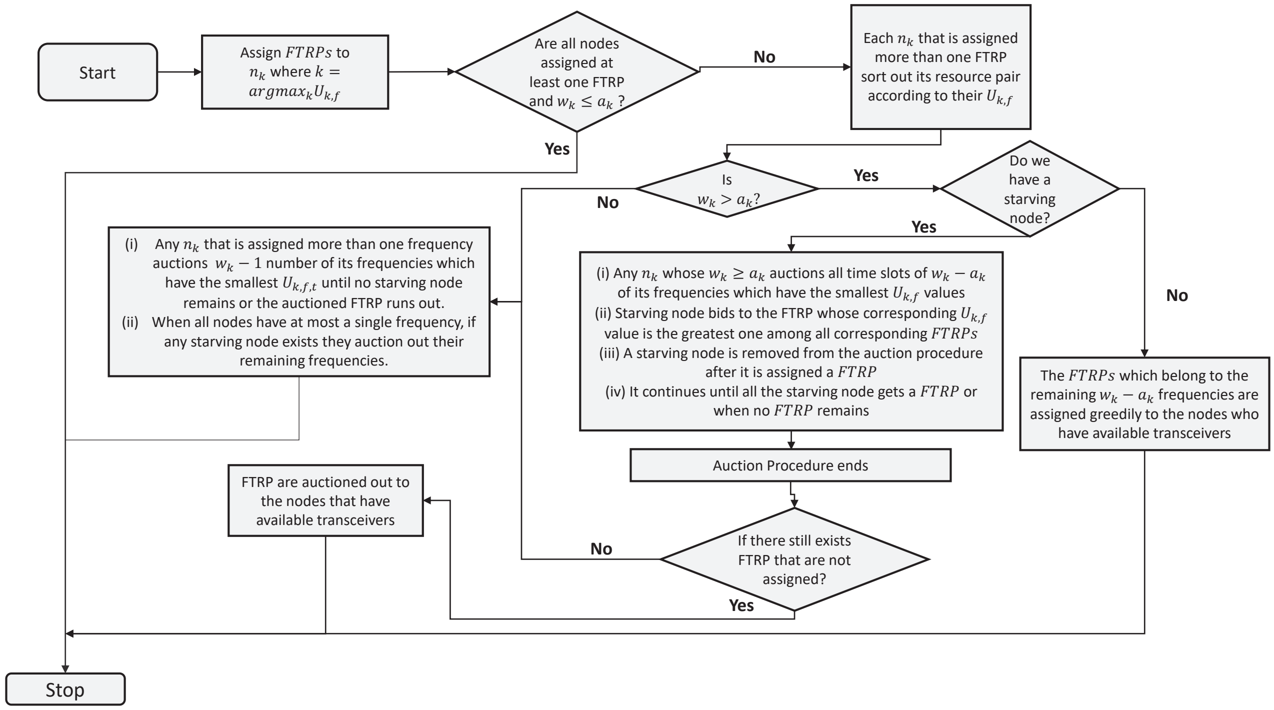

Fig. 3: Auction theory based scheduler flow diagram

using a first-price sealed-bid auction in designing suboptimal scheduler for the throughput maximizing scheduler is manifold. First, $U_{k,f}$ values of any node k are independent from other nodes' values, and each node knows only its own $U_{k,f}$, which is the bid value. Since the bid values are independent and does not influence other bid values, the auctions are held in a sealed bid fashion. Secondly, first-price sealed-bid auctions are used in order to maximize the total throughput where an auctioned frequency time resource pair ($FTRP$) is assigned to the node who has the highest bid among all other bids. The result of an auction with respect to an $FTRP$ does not alter another auction result if they occur concurrently.

We assume that all nodes have the same number of transceivers when they are managed by the same gateway. The auctioning procedure requires three main identifiers which are: (1) The auctioned resources r, (2) The bidders, (3) The auctioneer We indicate the auctioned resources as $FTRP$; the bidders as the nodes in the network; and lastly the auctioneer as the gateway (where the scheduler resides).

If we ignore constraint (3), (5) and (6), the optimal solution is achieved when a $FTRP$ is assigned to the node k that has the maximum $U_{k,f}$ value for the frequency f (of this resource r). The aim is to assign at least one $FTRP$ to each node, and to avoid any starving node at the end of the algorithm. We define a starving node as the node that are yet to be allocated any $FTRP$ during the execution of the algorithm. Our proposed scheduling algorithm is explained through steps 1 to 6 below and for the sake of clarity, we also provide an action flow diagram in figure 3.

STEP 1: For each frequency f, determine the node k that transmits the maximum $U_{k,f}$ using that frequency f. Assign f to that node k for all time slots of the slot frame period. To be specific, assign f to node k where $k = argmax_k U_{k,f}$. The number of frequencies allocated to node k at the end of step 1 is identified as w_k.

STEP 2: If at least one $FTRP$ is assigned to every node and $w_k \le a_k, \forall k$, end. Otherwise, each n_k having more than one $FTRP$ sort out its resource pairs according to their $U_{k,f}$ values. If any node k has $w_k > a_k$, go to step 3, otherwise go to step 6.

STEP 3: Do we have a starving node? if yes, go to step 4, otherwise go to step 5.

STEP 4: Any n_k whose $w_k \ge a_k$ auctions all time slots of $w_k - a_k$ of its frequencies which have the smallest $U_{k,f}$ values. The $FTRPs$ are auctioned simultaneously. $FTRP$ whose corresponding $U_{k,f}$ value is the highest one are bided by the starving node. When the starving node gets a $FTRP$, it is removed from the auction procedure. The auctions continue until all the starving node are assigned at least a $FTRP$ or when there are no more $FTRPs$. In the termination of the auctioning procedure, if we still have some $FTRPs$ that are not allocated to any node, the remaining $FTRPs$ are auctioned out to the nodes that have available antennas. Otherwise, if there still exists a starving node and all resources are assigned, then go to step 6.

STEP 5: Any n_k whose $w_k \ge a_k$ keeps its a_k frequencies with the largest $U_{k,f}$ values. The auction algorithm greedily assigns the $FTRP$ belonging to the remaining $w_k - a_k$ to the nodes who have available antennas.

STEP 6: Any n_k that is allocated more than one f auctions $w_k - 1$ number of its frequencies which have the smallest $U_{k,f}$ values until when the auctioned $FTRP$ runs out or no starving node remains.

The use of first-price sealed-bid auctions simplifies our algorithm and makes it more efficient, by preventing any additional operations for ensuring the termination of the algorithm.

C. Computational Complexity

In this section, we compute the computational complexity of our algorithm. At the end of step 1, assigning a $FTRP$ to each node has a complexity of $\mathcal{O}(NF)$ because for each frequency, we determine the maximum number of packets $U_{k,f}$. This is the best case scenario, when every node is assigned at least one frequency. In the worst case scenario, i.e., when all the frequencies are assigned to only one node (which in practice is very unlikely), we need to take into account the additional complexity of steps 2-6. In step 2, the frequencies are sorted out and will require $\mathcal{O}(F \log F)$ in terms of its complexity. Since we have $N-1$ starving nodes to bid for the frequencies, it will require $\mathcal{O}((N-1)\log(N-1))$ time to sort out their bids according to $N-1$ starving node. Therefore the total computational complexity of the algorithm is $\mathcal{O}(NF)+\mathcal{O}(F\log F)+(N-1)\times\mathcal{O}((N-1)\log(N-1))$, that simplifies to $\mathcal{O}((NF)+N^2 logN)$.

VI. Max-Min Fairness Scheduler

In this section, we propose a fairness based scheduling model for IIoT-TSCH networks. In prior algorithms, the throughput maximization scheduling problem is aimed at maximizing the total throughput while providing time and frequency allocation in a multi-channel environment where nodes are equipped with multiple antennas. However, this scheduling model may favour some nodes that have good channel conditions while leading to the deprivation of some other nodes. Hence, sustaining some notion of fairness among the nodes is an essential criteria a scheduling model should address. To achieve this aim, we propose a max-min fair scheduler. The goal is to fully maximize the total throughput while sustaining a certain fairness among the nodes. We present a scheduling model that attains fairness by maximizing the node throughput that has the minimum throughput amid all nodes, and the same time having an efficient communication between the nodes and the gateway. We utilize the exponential weighted low pass filter [29] to get the average throughput of node k in the slot frame period. This can be expressed as

$$\overline{M}_{\omega}^{k} = \left(1-\frac{1}{\omega}\right)M_{\omega}^{k} + \frac{1}{\omega}\sum_{f=1}^{F}\sum_{t=1}^{T}U_{k,f}X_{k,f,t} \tag{13}$$

where M_{ω}^{k} is expressed as the average throughput of node k accumulated in the last ω slot frame periods and is expressed as packets per time slot. The values of M_{ω}^{k} are initialized to zero. $\sum_{f=1}^{F}\sum_{t=1}^{T}U_{k,f}X_{k,f,t}$ indicates the throughput of node k in the current slot frame period with $\frac{1}{\omega}$ as the weight given to it. M_{ω}^{k} is updated as $M_{\omega}^{k} \leftarrow \overline{M}_{\omega}^{k}$ at the end of each slot frame period.

Given the values of F, N, T, a_k, ω, M_{ω}^{k} and $U_{k,f}$, the max-min fair scheduling problem is formulated as follows:

$$max \quad D \tag{14}$$

s.t

$$D \leq \left(1-\frac{1}{\omega}\right)M_{\omega}^{k} + \frac{1}{\omega}\sum_{f=1}^{F}\sum_{t=1}^{T}U_{k,f}X_{k,f,t} \quad \forall k \in \mathcal{N} \tag{15}$$

$$(3),(4),(5),(6) \tag{16}$$

The objective function in (14) and the constraints in (15) and (16) maximizes

$$\min_{k\in\mathcal{N}} \ \overline{M}_{\omega}^{k} \tag{17}$$

Providing fairness is indicated by introducing a windowing mechanism that takes into account the throughput of the nodes in the recent slot frame periods. For example, if a node suffers from low throughput due to interference from other devices within its vicinity in the current slot frame period, it can be compensated from the loss in the next slot frame period due to the history of the throughput information accumulated in M_{ω}^{k}. ω can be referred to the number of past slot frame periods during which the network conditions changes are significant. If ω is very small, the scheduler will be stringent in reacting to the changes in the network conditions. If ω is too large, the scheduler will react quickly to the fluctuations in the network conditions. When $\omega=1$, it denotes that the network conditions in the current slot frame period are considered without regarding what happened in the past.

Theorem 2. We denote Γ_{MFS}^{OPT} to be the optimum solution of the problem formulation of the max-min fair scheduling (MFS) problem in (14) - (16) for $\omega=1$. Let Γ_{MFS}^{UB} be the upper bound for the optimum solution of MFS for $\omega=1$ and Γ_{ThrMax}^{OPT} be the total throughput of the problem formulation of the throughput maximizing scheduling problem in (2) - (6) for $\omega=1$. Therefore

$$\Gamma_{MFS}^{OPT} \leq \Gamma_{MFS}^{UB} \leq \frac{\Gamma_{ThrMax}^{OPT}}{N} \tag{18}$$

Proof. We denote Γ_{MFS}^{TOT} as the total throughput resulting from the execution of the max-min fair scheduling problem. Since the throughput of the node with minimum throughput is indicated as Γ_{MFS}^{OPT}, the throughput of any other node among the remaining $N-1$ nodes is at least Γ_{MFS}^{OPT}.
Then,

$$\Gamma_{MFS}^{TOT} \geq \Gamma_{MFS}^{OPT} + (N-1)\Gamma_{MFS}^{OPT} \tag{19}$$

So,

$$\Gamma_{MFS}^{TOT} \geq N \times \Gamma_{MFS}^{OPT} \tag{20}$$

Since,

$$\Gamma_{ThrMax}^{OPT} \geq \Gamma_{MFS}^{TOT} \tag{21}$$

Therefore

$$\Gamma_{MFS}^{OPT} \leq \frac{\Gamma_{ThrMax}^{OPT}}{N} \tag{22}$$

□

A. Algorithm for the Max-Min Fair Scheduling Problem

We present a heuristic algorithm for the max-min fair scheduling problem and it is highlighted in steps 1-6. Γ_k is denoted as the sum of $U_{k,f}$ values that are yet to be allocated to node k during any phase of the heuristic algorithm.
STEP 1: Γ_k are initialized to zero.
STEP 2: We denote N_k^{av} as the set of nodes that have an available antenna for frequency assignment. N_k^{av} is initialized to N because all node's antennas are accessible for all time slot at the start of the algorithm. In other words $N_k^{av} \leftarrow N \;\; \forall k \in \mathcal{N}$.
STEP 3: Each $FTRP$ is assigned to the node that has the minimum Γ value among the nodes that have an available antenna (ties are broken arbitrarily).
STEP 4: We denote $Z(k)$ as the new objective function value, i.e., sum of Γ_k values if $FTRP$ is assigned to node k.
STEP 5: The assignment is made to the selected k^*. If node k^* is assigned as many $FTRP$ as antennas, then the node is removed from the list of available nodes for time slot t (N_k^{av}).
STEP 6: The algorithm terminates after all $FTRP$ are assigned to some node.

The fairness heuristic algorithm for max-min scheduler greedily selects the node that attains the best possible objective function at each iteration. This aims to increase the throughput of the node with minimum throughput in max-min fair scheduling.

B. Computational Complexity

The algorithm scans for the list of nodes in N_k^{av}, the size of which is no more than N throughout the assignment of f. Since there are F frequencies, the complexity of the fairness heuristic scheduler is $\mathcal{O}(NF)$.

VII. Performance Evaluation

In this section, we examine the performance of the proposed algorithms through a simulation-based study. We simulate our results using Java. A TSCH network consisting of sensor nodes randomly placed in a square area of $100m \times 100m$ was considered with the gateway (GW) located in the center of the network. The x and y coordinates of each node follow a uniform distribution. Every node is equipped with a radio that has a transmission range of 30 meters. $U_{k,f}$ values differ owing to the changes in network conditions, and are obtained for 3000 slot frame periods in each set of simulations where the average is considered. The same set of the $U_{k,f}$ values are compared for each scheduling algorithms. Each slot frame period consists of 72 time slots, where $t = 10ms$. We use ILOG-CPLEX [7] to solve the throughput maximizing scheduling problem in (2) - (6) and the max-min fair scheduling problem in (14) - (16). Table I summarizes the basic simulation parameters.

TABLE I: Summary of the Simulation Parameters

Number of sensor nodes	[2, 30]
Deployment area	square, 100m × 100m
Time slot duration	10ms
Slotframe size	72
Number of frequencies	16
$U_{k,f}$	Maximum number of packets that can be sent by n_k at link $l_{k,f}$ during a slot frame
Channel Bandwidth, B	$2MHz$

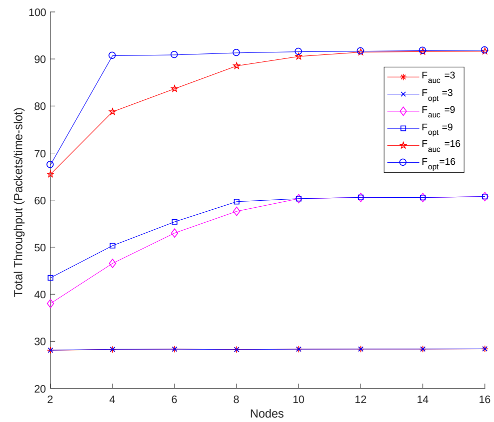

Fig. 4: Average total throughput comparison of the proposed auction scheduling algorithm vs the optimum results obtained through CPLEX simulations

In this paragraph, the proposed auction based heuristic scheduler is compared with the optimal result obtained through ILOG-CPLEX. Figure 4 highlights how the average network throughput is affected by varying the number of nodes and frequencies. The number of frequencies used in the ILOG-CPLEX simulations is indicated as F_{opt}, while F_{auc} is indicated as the number of frequencies used in our proposed auction based heuristic algorithm. We notice from the figure that the performance of our heuristic is very close to the optimum value in all cases. Moreover, we observe from the same figure that the average network throughput is almost invariant for varying the number of nodes when we have small number of frequencies in the system (i.e. $F = 3$) in both cases. This is due to the fact that the number of frequencies present for the nodes is small, which makes no much difference increasing the number of nodes in the system, reason being that almost all the resources are already occupied when the number of nodes is small. As the number of frequencies increases, the average network throughput grows with the number of nodes. This behaviour does not change up to a threshold where network throughput saturates.

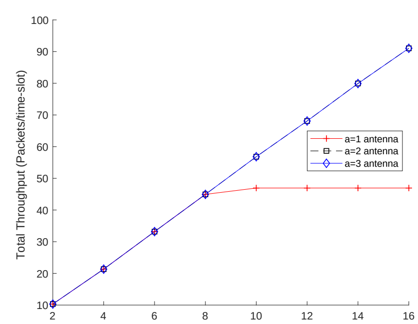

Fig. 5: Average total network throughput for the throughput maximization scheduling problem by varying the number of antennas and frequencies

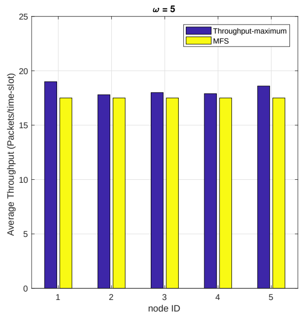

Fig. 6: Average throughput of all nodes

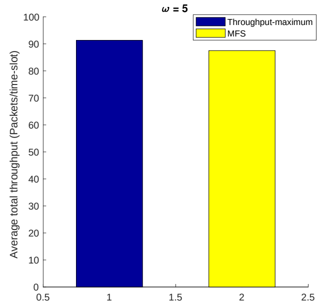

Fig. 7: Average total throughput

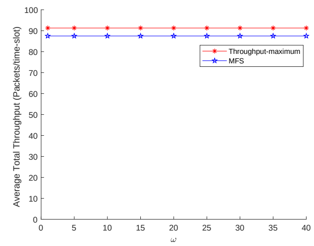

Fig. 8: Average total throughput vs ω

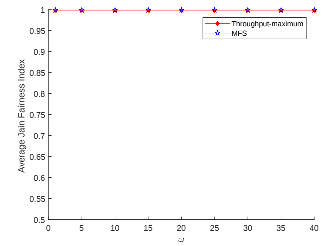

Fig. 9: Average Jain's Fairness Index vs ω

We also point out in figure 5 how the number of frequencies and antennas affects the performance of the throughput maximization scheduler. We can observe that it is reasonable to increase each node antenna in the network provided there is a definite number of frequencies in the network. For example, having $a_k = 1$ has the same performance as $a_k > 1$ when the number of frequencies is small (when $F < 8$). On the other hand, there is a significant difference in the throughput with an increase in the number of antennas from 1 to 2 if number of frequencies are between 8 and 16. The reason for this behaviour is highlighted by constraint (3), where each node is assigned at least one $FTRP$. To be compliant with this constraint, some frequencies are allocated to the first antenna of each node by the scheduler, and the scheduler continues to allocate frequencies to other antennas. For instance, in figure 5, we assume that $N = 8$, and until a certain point when $N = F$, frequencies are allocated to the first antennas of each node by the scheduler. A corresponding effect occurs on the total throughput between the increase in the number of antennas and also the increase in the number of nodes. Moreover, from when $N = F$ to $F = 2N$, the second antenna of each node is assigned a frequency by the scheduler.

Figure 6 and figure 7 presents the performance of the throughput maximizing scheduler *(Throughput-maximum)* and the max-min fair scheduler *(MFS)* with $N = 5$, ω = 5, $F = 15$ and $a_k = 2$. It can be seen that MFS achieves uniform throughput among the nodes compared to the throughput maximizing scheduler at the expense of a minimal reduction in the total throughput, thus attaining fairness.

A fairness measure is essential to assess the degree of fairness. Jain's index is predominantly used as a fairness measure [30]. It is defined as

$$J(T) = \frac{\left(\sum_{k=1}^{N} \Gamma_k\right)^2}{N \times \sum_{k=1}^{N} \Gamma_k^2} \tag{23}$$

where Γ_k denotes the throughput of node k, and N denotes the total number of nodes. Jain's index approaches *one* when the throughput values of the nodes move closer to each other. Jain's fairness index ranges from $\frac{1}{N}$ (only one node is served) to 1 (all nodes are served) [30]. Figure 8 and 9 illustrates the average total throughput values and the average Jain's fairness index values respectively varying ω for both throughput maximizing scheduler and MFS. When determining the average Jain's index values, the mean is measured and put into consideration after every slot frame period is taken into account. The performance of the throughput maximizing scheduler in respect to the average total throughput values and the average Jain's fairness index is invariant since ω is not a parameter of the formulated scheduling problem. Jain's index and the total throughput performances of MFS are almost invariant of ω.

In Figure 10 and figure 11, we present the average total throughput and average Jain's Index values for the MFS and throughput maximizing scheduler using $\omega = 5$ while varying the number of nodes. The number of frequencies and the number of antennas are set to be $F = 15$ and $a_k = 2$ respectively. It is apparent that the average total throughput for both schedulers is almost invariant when increasing the number of nodes. In the case of the average Jain's fairness, the throughput maximizing scheduler decreases while increasing the number of nodes, while the average Jain's fairness is invariant for MFS when the number of nodes is increased. In most cases, throughput maximizing scheduler allocates frequencies and time slots to the node that has the best channel conditions. As the number of nodes in the IIoT-TSCH cell increases, the probability to have a node with a better condition also increases. This multi-node diversity of the throughput maximizing scheduler increases the differences between the throughput values of the nodes. Moreover, as the number of nodes increases, the Jain's fairness index decreases based on the differences between the node's throughput values. MFS exhibits low total throughput and high Jain's fairness index since its goal is to ensure close proximity between the throughput values.

The average minimum throughput of MFS results (obtained using ILOG-CPLEX) and our proposed heuristic fairness algorithm *(HMFS)* are presented in figure 12 when $F = 15$, $a_k = 2 \, \forall k \in \mathcal{N}$, $\omega = 5$ while varying the number of nodes. It can be seen that the performance of our heuristic is very close to the optimum value obtained from ILOG-CPLEX. Moreover, as the number of node increases, the average minimum throughput decreases. This happens because there are many nodes contending for the same amount of resources which makes the average minimum throughput to

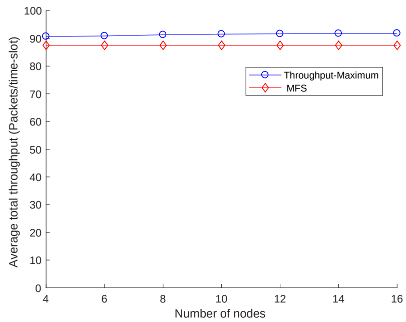

Fig. 10: Average total throughput for varying the number of nodes

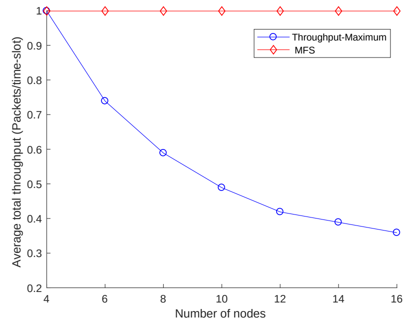

Fig. 11: Average Jain's Fairness Index for varying the number of nodes

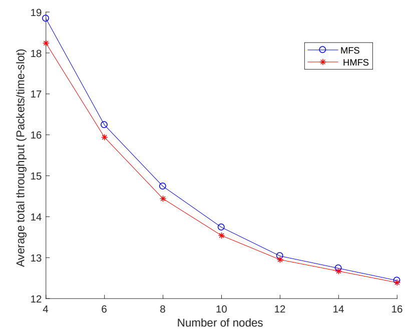

Fig. 12: Average minimum throughput for HMFS while varying the number of nodes

decrease.

VIII. Conclusions

In this article, scheduling models for throughput maximizing and fairness were proposed. First, a polynomial-time algorithm, which shows the optimal algorithm for the throughput maximizing scheduling problem was proposed. Certain instances of the problem were discussed along with their combinatorial properties. In addition to the polynomial time algorithm, a first price sealed bid auction was presented to address the throughput maximizing scheduling problem with a lower complexity. Moreover, the auction based scheduling algorithm yield a close result to the optimal algorithm. In addition, we made an assessment of MFS with respect to the average total throughput and Jain's fairness index varying ω and the number of nodes. We notice based on simulation results that increasing ω does not affect MFS. We developed a heuristic algorithm to address MFS problem, and based on simulations, the algorithm yield a close performance to the optimal one. We observed that the scheduling model can execute MFS when the number of nodes is large, which makes fairness to be significantly considered. In addition, there is no priority difference between nodes and the channel conditions between the nodes is fairly random.

References

[1] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Transactions on industrial informatics*, vol. 4, no. 2, pp. 102–124, 2008.

[2] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless networks*, vol. 23, no. 1, pp. 23–41, 2017.

[3] ISA-100.11a-2011, "Wireless Systems for Industrial Automation: Process Control and Related Applications," *ISA.100.11a-2011*, 2011.

[4] D. Chen, M. Nixon, S. Han, A. K. Mok, and X. Zhu, "Wirelesshart and IEEE 802.15. 4e," in *Industrial Technology (ICIT), 2014 IEEE International Conference on*. IEEE, 2014, pp. 760–765.

[5] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, April 2016.

[6] D. Stanislowski, X. Vilajosana, Q. Wang, T. Watteyne, and K. S. Pister, "Adaptive synchronization in ieee802. 15.4 e networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 795–802, 2014.

[7] I. I. CPLEX, "V12. 1: Users manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[8] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TISCH: Deterministic IP-Enabled Industrial Internet of Things (IOT)," *Communications Magazine, IEEE*, vol. 52, no. 12, pp. 36–41, 2014.

[9] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in wireless sensor networks: Distributed edge-coloring revisited," *Journal of Parallel and Distributed Computing*, vol. 68, no. 8, pp. 1122–1134, 2008.

[10] Á. A. Farías and D. Dujovne, "A queue-based scheduling algorithm for pce-enabled industrial internet of things networks," in *Embedded Systems (CASE), 2015 Sixth Argentine Conference on*. IEEE, 2015, pp. 31–36.

[11] R. Tavakoli, M. Nabi, T. Basten, and K. Goossens, "Topology management and tsch scheduling for low-latency convergecast in in-vehicle wsns," *IEEE Transactions on Industrial Informatics*, 2018.

[12] M. Ojo, S. Giordano, G. Portaluri, D. Adami, and M. Pagano, "An energy efficient centralized scheduling scheme in tsch networks," in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 570–575.

[13] I. Hosni and F. Théoleyre, "Self-healing distributed scheduling for end-to-end delay optimization in multihop wireless networks with 6tisch," *Computer Communications*, vol. 110, pp. 103–119, 2017.

[14] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, "Decentralized Traffic Aware Scheduling for Multi-Hop Low Power Lossy Networks in the Internet of Things," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops*. IEEE, 2013, pp. 1–6.

[15] J. Jung, D. Kim, J. Hong, J. Kang, and Y. Yi, "Parameterized slot scheduling for adaptive and autonomous tsch networks," *Channels*, vol. 2, no. 3, p. 5.

[16] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 337–350.

[17] X. Fafoutis, A. Elsts, G. Oikonomou, R. Piechocki, and I. Craddock, "Adaptive static scheduling in ieee 802.15. 4 tsch networks," in *Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on*. IEEE, 2018, pp. 263–268.

[18] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On Optimal Scheduling in Duty-Cycled Industrial IOT Applications Using IEEE802. 15.4E TSCH," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3655–3666, 2013.

[19] Z. Chu, T. A. Le, D. To, and H. X. Nguyen, "Sum throughput optimization for wireless powered sensor networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[20] X. L. Huang and B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2001, pp. 221–231.

[21] Y. Le, L. Ma, W. Cheng, X. Cheng, and B. Chen, "A time fairness-based mac algorithm for throughput maximization in 802.11 networks," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 19–31, 2015.

[22] H. Shi, M. Zheng, W. Liang, Z. Luo, and S. H. Hong, "A fairness-aware scheduling algorithm for industrial wireless sensor networks with multiple access points," in *Enterprise Systems (ES), 2017 5th International Conference on*. IEEE, 2017, pp. 287–293.

[23] L. Wang and L. Zhigang, "Research on control system of belt conveyor in coal mine," in *Electrical, Information Engineering and Mechatronics 2011*. Springer, 2012, pp. 885–891.

[24] L. Lovász and M. D. Plummer, *Matching theory*. American Mathematical Soc., 2009, vol. 367.
[25] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*. Springer Science & Business Media, 2003, vol. 24.
[26] D. Jungnickel and T. Schade, *Graphs, networks and algorithms*. Springer, 2005.
[27] W. T. Tutte, "The factorization of linear graphs," *Journal of the London Mathematical Society*, vol. 1, no. 2, pp. 107–111, 1947.
[28] V. Krishna, *Auction theory*. Academic press, 2009.
[29] J. S. Hunter *et al.*, "The exponentially weighted moving average." *J. Quality Technol.*, vol. 18, no. 4, pp. 203–210, 1986.
[30] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984, vol. 38.

Mike O. Ojo received the bachelor degree in Electrical and Electronic Engineering from Ladoke Akintola University of Technology, Ogbomoso, Nigeria in 2009, and a M.Sc degree in Telecommunication Engineering from the Politecnico di Milano, Milano, Italy in 2014. He is currently pursuing the Ph.D. degree in information engineering with the University of Pisa. His current research interests are in the field of Industrial Internet of Things, Software Defined Networking, Optimization, Resource Allocation and Scheduling Algorithms.

Stefano Giordano (M'89, SM'10) received the Laurea (cum laude) degree in electronics engineering and the Ph.D. degree in information engineering from the University of Pisa, where he is currently a Full Professor with the Department of Information Engineering and responsible for the telecommunication networks laboratories. He is the Representative of the University of Pisa in the Scientific Committee of CNIT (the Italian National Consortium for Telecommunications) and the University of Pisa in the GTTI (Group for Telecommunications and Information Theory). He has authored over 400 papers in international conferences and journals. He is a member of IFIP WG 6.3, Internet Society since its foundation in 1992, and a member of the Board of Directors of its Italian charter. He is the former Chair of the Communication Systems Integration and Modelling Technical Committee. He founded Juniper Networks, the first European Juniper Networks Higher Learning Center. He is chairing a Standardization Research Group of Comsoc on IoT Communications and Networking Infrastructure. He has been a reviewer for the NSF in the U.S., the EU, the Italian Ministry of Industry, and the Italian Ministry of Research (member of the albo of experts of the ministry). He is an associate editor for several journals. He has been the general chair, the TPC chair, and a TPC member for many international conferences. He was co-founder of three start-ups (Nextworks, Netresults, Natech) and co-founder of the Cubit Consortium where at present is president of the Scientific and Technical Committee).

Davide Adami received the degree in electronic engineering from the University of Pisa, Italy, in 1992. From 1993 to 1997, he was with Consorzio Pisa Ricerche, taking part in a lot of research projects funded by the EU, such as the MAESTRO ACTS Project. In 1997, he joined CNIT, where he is a Senior Researcher in the field of telecommunication networks. He has conducted research for several research projects funded by ASI, ESA, EU (FP6 RINGRID, FP7 DORII, FP7 OFELIA, FP7 Fed4FIRE, FP7 SCOUT), and the Italian MIUR. He has authored many papers in scientific journals and international conference proceedings. His research interests mainly concern software defined networking and network function virtualization in cloud data centers, the design and development of new innovative solutions for the integration of Cloud applications and networks architectures providing QoS support.

Michele Pagano received the Laurea (cum laude) in electronics and the Ph.D. degree in information engineering from the University of Pisa in 1994 and 1998, respectively. From 1997 to 2007, he was a Researcher with the Department of Information Engineering, University of Pisa, and then became an Associate Professor in 2010. He is currently the official instructor of the courses on telematics, performance of multimedia networks, network security and architectures, components and network services. He gave lectures on network performance analysis in different Polish and Russian universities. He has co-authored more than 200 papers published in international journals and conference proceedings. His research interests are related to statistical traffic characterization and network performance analysis, statistical traffic classification, scheduling algorithms, anomaly detection, queuing theory, security issues in distributed architectures and green networking. He has been involved in the activities of the NoE EuroNGI (design and dimensioning of the next generation Internet) and in several national and international projects, being the local coordinator for the 2006 PRIN Robust and Efficient Traffic Classification in IP networks and the 2008 PRIN Energy Efficient Technologies for the Networks of Tomorrow. In 2006 and 2007, he was a supervisor for Dr. Marchenko in an INTAS grant.