EDF scheduling and minimal-overlap shortest-path routing for real-time TSCH networks

Miguel Gutiérrez Gaitán[1]
CISTER – Research Centre in Real-Time and Embedded Computing Systems
Faculty of Engineering, University of Porto, Portugal
Institute of Engineering, Polytechnic Institute of Porto, Portugal
Facultad de Ingeniería, Universidad Andrés Bello, Chile
mgg@fe.up.pt

Luís Almeida
CISTER – Research Centre in Real-Time and Embedded Computing Systems
Faculty of Engineering, University of Porto, Portugal
lda@fe.up.pt

Pedro Miguel Santos
CISTER – Research Centre in Real-Time and Embedded Computing Systems
Institute of Engineering, Polytechnic Institute of Porto, Portugal
pss@isep.ipp.pt

Patrick Meumeu Yomsi
CISTER – Research Centre in Real-Time and Embedded Computing Systems
Institute of Engineering, Polytechnic Institute of Porto, Portugal
pmy@isep.ipp.pt

Abstract

With the scope of Industry 4.0 and the Industrial Internet of Things (IIoT), wireless technologies have gained momentum in the industrial realm. Wireless standards such as WirelessHART, ISA100.11a, IEEE 802.15.4e and 6TiSCH are among the most popular, given their suitability to support real-time data traffic in wireless sensor and actuator networks (WSAN). Theoretical and empirical studies have covered prioritized packet scheduling in extenso, but only little has been done concerning methods that enhance and/or guarantee real-time performance based on routing decisions. In this work, we propose a greedy heuristic to reduce overlap in shortest-path routing for WSANs with packet transmissions scheduled under the earliest-deadline-first (EDF) policy. We evaluated our approach under varying network configurations and observed remarkable dominance in terms of the number of overlaps, transmission conflicts, and schedulability, regardless of the network workload and connectivity. We further observe that well-known graph network parameters, e.g., vertex degree, density, betweenness centrality, etc., have a special influence on the path overlaps, and thus provide useful insights to improve the real-time performance of the network.

2012 ACM Subject Classification Computer systems organization → Real-time systems; Networks → Network algorithms; Networks → Data path algorithms

Keywords and phrases Real-time communication, Routing, Scheduling, TDMA, Wireless networks

Digital Object Identifier 10.4230/OASIcs.NG-RES.2021.2

Acknowledgements This work was partially supported by the project Safe Cities - Inovação para Construir Cidades Seguras, ref. POCI-01-0247-FEDER-041435, co-funded by the European Regional Development Fund (ERDF), through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020); also by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020).

[1] Corresponding author.

1 Introduction

Wireless sensor and actuator networks (WSAN) play nowadays an important role in industrial facilities. Wireless radio links, in general, bring the flexibility and scalability that wireline infrastructure lacks, but often with relatively lower bandwidth and reliability [4]. Yet, for many applications based on sensor and actuator networks, e.g., for real-time monitoring or even audio streaming [10], low data rate wireless technologies (up to 250 Kbps) are sufficient to satisfy typical bandwidth requirements. Similarly, the reliability of standards like WirelessHART, ISA100.11a, IEEE802.15.4e and 6TiSCH, based on IEEE 802.15.4-PHY, increased to levels that are, in many cases, compatible to wired networks [18].

Time-synchronized channel hopping (TSCH) is among the most popular standards in the scope of WSAN to support real-time data traffic. Salient features, such as time-division multiple-access (TDMA), centralized scheduling and frequency diversity, have gradually underpinned its adoption in a number of application domains, from factory automation and process control [9] to vehicles [16], paving the way for the Industrial Internet of Things (IIoT) and Industry 4.0 [15].

In these domains, real-time communication is essential to ensure satisfactory (and deterministic) performance. The predictable/analyzable (time-slotted) channel access of TSCH is appropriate for that purpose which, coupled with proper (typically centralized) scheduling and routing algorithms, can provide safe operational bounds for worst-case end-to-end delays and schedulability. Several research efforts have pursued real-time communication in TSCH networks, but mostly focusing on packet scheduling. Routing, in the other hand, is often assumed as standard, e.g., using the shortest-path algorithm, leading to sub-optimal real-time performance.

In this work, we deal with the so-called *real-time wireless routing* [19] for TSCH networks, whose primary goal is to enhance and/or guarantee the real-time properties of the network based on routing decisions. In this respect, Wu et al. [19] proposed a *conflict-aware routing* method for WirelessHART networks with packet transmissions scheduled using a fixed-priority policy. We tackle alike foundational questions from this work, but for TSCH WSANs under the earliest-deadline-first (EDF) scheduler, instead. We propose a *minimal-overlap shortest-path* routing based on a greedy heuristic driven by reducing path overlaps among network flows. We show, by leveraging on prior work on schedulability analysis, that our method considerably improves the network schedulability when compared to the conventional (hop-count) shortest-path method.

2 Related Work & Contribution

Theoretical and empirical studies for modelling and assessing the real-time performance of TSCH-like networks have been discussed in recent literature, e.g., [13, 6, 5, 8, 1, 3, 14], usually having as the main focus priority-based packet scheduling algorithms. The span of analytical works includes the design of methods based on response-time analysis [13], supply/demand-based tests [6, 5], network calculus [8], etc., often deriving theoretical/empirical bounds attempting to guarantee worst-case real-time network performance. Both fixed-priority and dynamic-priority schedulers have been covered, most of the times assuming a standard behaviour for the rest of network features, e.g., routing, channel assignment, etc.

While for routing there are many works available in the literature [12] addressing TSCH networks, only a few of them fit into the class of *real-time wireless routing* [19], i.e., tailored routing methods aiming to enhance and/or guarantee the real-time performance of wireless networks. Wu et al. [19] made a step ahead in this direction by proposing a *conflict-aware*

real-time routing for WirelessHART networks under a fixed-priority policy, but they did not address dynamic-priority schedulers. Their work leverages on a prior delay analysis for TSCH-like networks, which derives in part from the real-time CPU scheduling theory [7].

We highlight the importance of this prior analysis in our work, allowing to split up the end-to-end delay analysis into two components: (*i*) the effect of channel contention, and (*ii*) the effect of wireless transmission conflicts. The former is conveniently mapped to the multiprocessor contention concept, by assuming the number of cores as equal to the number of (radio) channels in TSCH networks. The second is specific to wireless transmission scheduling, and model the restriction of half-duplex transceivers to transmit/receive alternately, a challenging condition under mesh network topologies, which has a significant impact on end-to-end delays and schedulability.

As in [19], we focus on the latter factor to improve routing decisions, i.e., the effect of the transmission conflicts on real-time performance, but we target transmissions scheduled under EDF. Moreover, distinctly to [19], we do not generate routes one-by-one until they become schedulable; instead, we provide a set of paths with minimal (node-) overlaps between flows, and then, we test the overall schedulability. We draw attention to the effectiveness of our approach against to the more common (hop-count) shortest path method, as well as in terms of the bandwidth utilization benefits brought by EDF in comparison to fixed-priority schedulers. We note that to the best of our knowledge, this is the first joint EDF scheduling and *real-time wireless routing* framework for TSCH networks.

3 System Model: TSCH-based network and EDF scheduling

We consider a WSAN as the one represented in Figure 1. The network consists of a finite number of $N \in \mathbb{N}$ nodes, including one gateway, multiple access points (APs), and several field devices (i.e., sensors and actuators). The field devices are wirelessly connected to the APs forming a mesh network topology, and each of them is equipped with half-duplex omnidirectional radio transceivers. The APs are directly linked to the gateway, which, in turn, enables bidirectional communication between the field devices and other entities outside the network, e.g., the network manager, process controller, host application, etc. The network manager is a software module (typically running on the gateway) which collects network topological information and is responsible for both scheduling and routing functions.

We assume the network is TSCH-based, i.e., relies on an IEEE 802.15.4 compatible physical layer, and uses a centralized multi-channel TDMA protocol with global synchronization. The multi-channel feature enables concurrent per-slot (but not per-channel) transmissions based on a channel hopping technique over a number of m active (i.e., not-blacklisted) radio channels, with $1 \leq m \leq 16 \in \mathbb{N}$. The length of the time slots is fixed ($\sim$ 10ms), and corresponds to a (dedicated) time interval to allocate a single packet transmission, a maximum number of $w - 1$ retransmissions (with $w \in \mathbb{N}$), and their corresponding acknowledgements.

Sensor nodes periodically transmit data through the network to external entities, e.g., to a remote controller (located at the gateway's node position), which, in turn, deliver control commands to the actuators (see Fig. 1). We consider transmissions occur in per-slot/per-hop basis, following predefined multi-hop routes between the sensors and the gateway (uplink), and between the gateway and the actuators (downlink), both with stringent timing delivery constraints. As for the sake of simplicity, we consider the routes are set under *source routing*, i.e., using pre-defined single routes for both uplink and downlink.

Without loss of generality, we also assume the topology maintenance is a native in-built centralized service or that can be further implemented, e.g., as in [17].

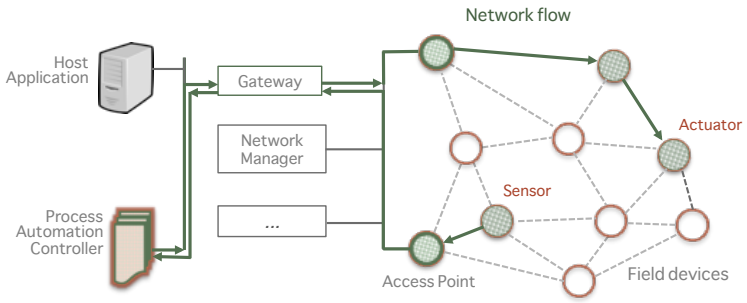

Figure 1 Pictorial representation of an industrial WSAN.

3.1 Network model

Given the above features, the network is modelled as a graph $G = (V, E)$, where V represents the set of vertices (or nodes), and E the set of edges (or links) between those nodes. We assume the graph is undirected and connected, but not complete, i.e., there is a path between any pair of nodes in the network, but not a link between every node pair. The total number of nodes N corresponds to the total number of vertices $|V|$, i.e., $N = |V|$, where one node acts as a gateway, and the rest $N - 1$ nodes correspond to the multiple APs and the several field devices. We further assume the gateway is the node with the highest *betweenness centrality*, i.e., the node if being removed, has the greatest impact on the overall network connectivity.

We consider a subset $n \in \mathbb{N}$ of the field devices are used to generate data (e.g., sensor measurements), and the rest $N - n - 1$ nodes act as relay. Note that when not transmitting their own data, the n transmitter field devices can act as relay too.

3.2 Flow model

We denote $F \stackrel{\text{def}}{=} \{f_1, f_2, \ldots, f_n\}$ the set of n real-time network flows to be transmitted from source to destination by following an EDF policy. Each f_i represents a periodic time-constrained end-to-end communication flow, characterized by a 4-tuple (C_i, D_i, T_i, ϕ_i). C_i denotes the effective transmission time between source and destination, T_i the period (or the sampling rate of sensors), D_i the relative deadline, and ϕ_i the routing path. These parameters are given with the interpretation that each flow f_i releases a potentially infinite number of transmissions. The γ^{th} instance of those transmissions, with $\gamma \in \mathbb{N}$, is denoted as $f_{i,\gamma}$, and is released at the time $r_{i,\gamma}$, such that $r_{i,\gamma+1} - r_{i,\gamma} \stackrel{\text{def}}{=} T_i$. Then, in accordance with the EDF policy, $f_{i,\gamma}$ is constrained to reach its destination before its absolute deadline, i.e., $d_{i,\gamma} \stackrel{\text{def}}{=} r_{i,\gamma} + D_i$. We assume this is a constrained deadline model, i.e., $D_i \leq T_i$, thus allowing only a single transmission of flow f_i at any time slot.

Note that here C_i is interpreted as the time required by flow f_i to be completely transmitted from source to destination, but when it does not suffer whatsoever interference from other flows. We thus assume C_i can be computed as $C_i = \zeta_i \times w$ (slots), where ζ_i is total number of links in the route path ϕ_i, and w is the number of transmission slots assigned to a flow in each link, including retransmissions. We adopt w fixed (as in WirelessHART) for all the links, thus C_i being exclusively dependent on topology and routing dynamics.

3.2.1 Supply/demand-based schedulability analysis for EDF

For the sake of completeness, we briefly revisit our prior work on schedulability analysis for WSANs [6]. Particularly, we leverage on the so-called *forced-forward demand-bound function* (FF-DBF) [2] when applied to the WSANs domain, which is a state-of-the-art supply/demand-bound schedulability assessment for TSCH-like WSANs under EDF [5].

We reproduce here the formal expression for the schedulability test, defined as the relationship between the *supply-bound function* (SBF) [20], i.e., the minimal transmission capacity offered by a network with m channels, and the FF-DBF, i.e., the upper-bound on the maximum possible demand for a set of real-time network flows $F = \{f_1, f_2, \ldots, f_n\}$ (with $n \in \mathbb{N}$) when evaluated over a given time interval of length ℓ

$$\sum_{i=1}^{n} \text{FF-DBF}(f_i, \ell) \leq \text{sbf}(\ell),\ \forall \ell \geq 0 \tag{1}$$

where the SBF is formally defined as follows

$$\text{sbf}(0) = 0\ \wedge\ \forall \ell, k \geq 0 : \text{sbf}(\ell + k) - \text{sbf}(\ell) \leq m \times k \tag{2}$$

and the FF-DBF (for TSCH-based WSANs) is defined as

$$\frac{1}{m} \sum_{i=1}^{n} \text{FF-DBF}^{ch}(f_i, \ell) + \sum_{i,j=1}^{n} \left(\Delta_{i,j} \max\left\{ \left\lceil \frac{\ell}{T_i} \right\rceil, \left\lceil \frac{\ell}{T_j} \right\rceil \right\} \right) \tag{3}$$

where $\Delta_{i,j}$ denotes the transmission conflicts delay (due to path overlaps) between any pair of flows f_i and $f_j \in F$, and T_i and T_j the respective transmission periods of these flows.

▶ Note. On (1), unlike to the common understanding on multiprocessors [2], the FF-DBF notion refers to the upper bound on network demand due to the contribution of two components: (*i*) channel contention, equivalent to the (core) contention concept on multi-core platforms, and (*ii*) transmission conflicts, an abstraction specific to wireless transmission scheduling. On (3), these two components are formally dissociate as a summation. On the left side, the expression denoted as FF-DBF^{ch} corresponds to the channel contention contribution, which is equivalent to the expression on multiprocessors, but here it models the restriction imposed to network flows to be simultaneously scheduled on different channels (see [6] for the complete expression of FF-DBF^{ch}). On the right side, the expression designates the contribution of transmission conflicts, which represents the delay experienced due to multiple flows encountering on a common node. We also note that this is a key aspect for the motivation and further understanding of the solution later proposed in the present work.

4 Problem Formulation

Given the network and flow models presented in Section 3, we consider the problem of finding the optimal set of flow paths $\Phi_{opt} \stackrel{\text{def}}{=} \{\phi_1^{opt}, \phi_2^{opt}, \ldots, \phi_n^{opt}\}$ that minimizes the *overall number of path overlaps* between any pair of flows in the network.

▶ **Definition 1.** *We denote as* Ω *the* overall number of path overlaps *between any pair of flows in the set* $F \stackrel{\text{def}}{=} \{f_1, f_2, \ldots, f_n\}$. *Particularly,* Ω *is the sum of all the individual* node overlaps δ_{ij} *between the routes of any pair of flows* f_i *and* f_j *in the set* F, *where* $i, j \in [1, n] \wedge i \neq j$.

▶ **Definition 2.** *We denote as* $F_0 \stackrel{\text{def}}{=} \{f_1^0, f_2^0, \ldots, f_n^0\}$ *the original set of flows in the network which set of flow paths* $\Phi_0 \stackrel{\text{def}}{=} \{\phi_1^0, \phi_2^0, \ldots, \phi_n^0\}$ *is obtained using a conventional (hop-count) shortest-path algorithm, ergo* Ω_0 *is the respective overall number of path overlaps for* Φ_0.

▶ **Definition 3.** *We denote as* $F_k \stackrel{\text{def}}{=} \{f_1^k, f_2^k, \ldots, f_n^k\}$ *(with* $k \in \mathbb{N}$*) the* k^{th} *variation of the set of flows* F_0 *when a sub-optimal set of routes* $\Phi_k \stackrel{\text{def}}{=} \{\phi_1^k, \phi_2^k, \ldots, \phi_n^k\}$ *is considered, thus* Ω_k *is the* k^{th} *overall number of path overlaps produced.*

Given a network graph G, an initial solution Φ_0 with respective Ω_0, and a number of $k_{max} \in \mathbb{N}$ sub-optimal sets of routes Φ_k, we formulate the problem of minimizing Ω as follows:

$$\begin{aligned} \underset{k}{\text{minimize}} \quad & \Omega_k = \sum_{\forall i,j \in [1,n] \wedge i \neq j} \delta_{i,j}(\Phi_k) \\ \text{subject to} \quad & k \in [1, k_{max}], \\ & \Phi_k \in [\Phi_1, \Phi_{k_{max}}] \end{aligned} \tag{4}$$

where $\delta_{i,j}(\Phi_k)$ is the number of *node overlaps* between the routes of the flows f_i^k and $f_j^k \in F_k$ (i.e. δ_{ij}^k), $\forall i, j \in [1, n] \wedge i \neq j$. The result is $\Omega = \Omega_k^{min}$, i.e., the minimal *overall number of overlaps* within all the k_{max} possible sets of flow paths $[\Phi_1, \Phi_{k_{max}}]$. We denote as Φ_{opt} the set of optimal routes, to any of the Φ_k sets of flow paths that produce Ω_k^{min}.

▶ Note. Each individual route ϕ_i^k in Φ_k is defined as a sub-optimal version of the shortest-path ϕ_i^0, i.e., the route length ζ_i^k of ϕ_i^k is always greater than or equal to ζ_i^0. In the same way, each individual transmission time C_i^k of f_i^k is always a sub-optimal version of C_i^0, thus C_i^k being always greater than or equal to C_i^0. Yet, we make clear that a larger C_i^k does not necessary implies a larger Ω_k, and vice-versa. We also note that although, in general, a larger C_i^k may lead to larger end-to-end delays, we conjecture that this impact is less detrimental than the impact of flow path overlaps, thus we target to minimize this latter factor.

5 Proposed Solution: Minimal-Overlaps (MO) Greedy Heuristic

We propose an approximate method to solve the problem formalized in (4). The approach is based on a greedy heuristic that recommends a single set Φ_k at each k^{th} iteration, and then computes the corresponding Ω_k. The smallest of the Ω_k after k_{max} iterations is designated as Ω_k^{min}, which is reported at the last iteration. We detail the proposed method as follows:

Step 1 (Initial solution)

- At $k = 1$, $\Phi_k = \Phi_1$ is computed as function of the path overlaps resulting from Φ_0, i.e., from the initial set of (hop count) shortest-paths, and from the graph $G = (V, E)$.
- Φ_1 is computed as the set of (weighted) shortest-paths of $G^1 = (V, E^1)$, a modified version of the unitary-weighted G whose set of edges is weighted as function of the node-overlapping degree derived from the set of flow paths Φ_0.
- The cost function $W_{i,j}(u, v)$ that weight any edge (u, v) in G whose nodes u and $v \in V$ are simultaneously in any pair of routes ϕ_i^0 and $\phi_j^0 \in \Phi_0$ is defined as follows:

$$W_{i,j}(u, v) = 1 + \sum_{e=1}^{\delta_{i,j}^0} \psi \tag{5}$$

where $\psi \in \mathbb{R}$ is an arbitrary factor[2], and $\delta^0_{i,j}$ is the number of *node overlaps* resulting from the routes ϕ^0_i and $\phi^0_j \in \Phi_0$, $\forall i, j \in [1, n] \wedge i \neq j$.

- Therefore, Φ_1 is the resulting set of (weighted) shortest-paths over G^1, and Ω_1 the corresponding *overall number of overlaps* for Φ_1.
- If $\Omega_1 < \Omega_0$, then $\Omega^{min}_k = \Omega_1$, else $\Omega^{min}_k = \Omega_0$.

Step 2 (Greedy search)

- For any $k \in]1, k_{max}]$, the search for a Ω^{min}_k is generalized.
- $G^k = (V, E^k)$ is defined as the modified version of $G^{(k-1)} = (V, E^{(k-1)})$, whose set of edges is weighted using the following cost function due the node-overlapping of flow paths:

$$W^k_{i,j}(u, v) = 1 + \sum_{e=1}^{\delta^{(k-1)}_{i,j}} \psi \tag{6}$$

where $\delta^{(k-1)}_{i,j}$ results from the routes $\phi^{(k-1)}_i$ and $\phi^{(k-1)}_j \in \Phi_{(k-1)}, \forall i, j \in [1, n] \wedge i \neq j$.

- Φ_k is thus computed as function of the path overlaps resulting from the set $\Phi_{(k-1)}$.
- If $\Omega_k < \Omega^{min}_k$, then $\Omega^{min}_k = \Omega_k$, else $\Omega^{min}_k = \Omega^{min}_k$.

Step 3 (Best solution)

- At $k = k_{max}$, the algorithm finishes and provides Ω^{min}_k, i.e., the minimal *overall number of overlaps* within all the k_{max} Φ_k sets recommended. Note that the quality of Ω^{min}_k will depend on the quality of the generated Φ_k sets, as well as on the total number of iterations.
- The optimal set of routes Φ_{opt} is thus the Φ_k which provides Ω^{min}_k.

6 Performance Evaluation

We report here the relevant information for the data sets generation and performance evaluation of both, the proposed *minimal-overlap* (weighted) shortest-path method denoted as **MO**, and the baseline (hop-count) *shortest-path* method, denoted as **SP**. We present the assessment of the real-time performance through the schedulability ratio metric by considering varying network topologies and workload conditions. We further assess the influence of varying topologies and workload on the number of overlaps, routes length, channel contention and transmission conflicts. We show that MO significantly outperforms SP in terms of schedulability ratio, number of overlaps and transmission conflicts, while having marginal impact on the average length of the routes and channel contention.

6.1 Simulation setup

The configuration details related to the generation of random network topologies and real-time network flows are described next:

[2] The arbitrary factor Ψ is a user-defined parameter, here assumed as constant, but that can be optimized to provide better (e.g., faster) solutions for Ω^{min}_k. Yet, this aspect is not covered in this work.

Network topologies

We prepared a set of 100 network topologies built upon the random generation of network graphs. Each graph was created based on a sparse uniformly distributed random matrix of size NxN (with $N \in \mathbb{N}$) and density Λ (with Λ in $[0, 1] \in \mathbb{R}$). Each sparse matrix acted as an adjacency matrix for the graph generation. The size of the sparse matrix (NxN) as well as the number of vertices in the graph (N) were fixed for all the simulation instances. We set $N = 66$ as in [19], for benchmarking purposes. The vertex with the highest betweenness centrality[3] was chosen as gateway, while the rest $N - 1$ vertices represented field devices and access points. A subset of $n \subset N$ of field devices is chosen as sensors, thus assumed to periodically transmit data to the gateway. For comparison, the range of n varied within $[2, 22]$ as in [19]. We considered varying values of $\Lambda = \frac{\lambda}{N}$, where λ indicates the median vertex degree of the graph. We controlled Λ by varying $\lambda \in \mathbb{N}$ in the range $[4, 12]$. We justify this choice since in practical WSANs deployments, each node is typically required to be connected at least to other 3 nodes (i.e., $\lambda \geq 3$). We note that, in general, this setup provided connected graphs, but in the few cases that nodes were disconnected, we forced a random connection (edge) with any of the other connected vertices.

Given the above configuration, we generated a set of shortest-path (hop-count) routes between the set of n sensors and the gateway, for each graph, thus providing 100 instances of sets of n routes for the baseline method. In the case of the MO method, the set of routes was generated from the baseline, but taken into consideration the degree of node-overlapping between routes as described in Section 5. The proposed set of (minimal-overlap) routes is thus the result of the evaluation of k_{max} edge-weighted versions of each baseline graph instance. On the weight functions, we used the arbitrary factor Ψ as equal to the graph density, i.e. $\Psi = \Lambda$, for all the cases. We considered this value based on empirical observations. For the sake of scalability, we considered for all the experiments $k_{max} = 100$. We observed that a greater number of iterations (k_{max}) can lead to a lower node-overlapping degree, but at the cost of higher execution times; thus we do not further explore this factor. Note that in the case of overall number of overlaps Ω_k that reaches zero, the algorithm stops.

Network flows

We consider a random set of $n \in [2, 22]$ real-time network flows for each of the 100 topologies generated. The complete set of flows corresponds to the set of periodic data transmissions generated by the $n \subset N$ sensor nodes in the graph. Each of these flows f_i is characterized by a 4-tuple (C_i, D_i, T_i, ϕ_i) following the model described in 3.2. Each C_i represents the effective transmission time (in slots) for the route ϕ_i, and can be obtained directly from the product of the number of hops (links or edges) traveled by the path ϕ_i from source to destination, and the number of transmissions assigned to each slot (we assume $w = 2$, as in WirelessHART). Thus, each of the 100 random graph instances is also generating, randomly, the C_i and ϕ_i occurrences. Hence, being applicable for both the MO and the SP methods. The corresponding T_i periods were assumed as random harmonically generated in the form of 2^η time slots, with $\eta \in \mathbb{N}$ in the range $[4, 7]$ (as in [19]). This assumption leads to a direct computation of the hyperperiod $H \in \mathbb{N}$ (a.k.a, superframe length) as the maximum period within the range of harmonic periods, or as generally defined, the least

[3] The betweenness centrality metric was chosen to maintain a consistent relevance of the gateway within the random topologies, thus avoiding an arbitrary gateway position (e.g. at the border). As different centrality metrics can be defined based on application needs, this consideration requires further research.

common multiple of the set of periods, i.e., $H = lcm(T)$, where $T = \{T_1, T_2, \ldots, T_n\}$. So, in this case, $H = 2^7 = 128$ slots (or equivalently 1280ms). We used $H = 128$ slots as the length of the time interval ℓ for the purposes of schedulability assessment, as well as for the performance evaluation of all the other metrics. The schedulability was evaluated using the test presented in 3.2.1, when considering a worst-case $\Delta_{i,j}$ as in [11]. Finally, we assume $D_i = T_i$ for all the cases, thus reducing the original problem to an implicit-deadline model.

▶ Note. For the sake of simplicity, we have only considered network flows that travel from sensors to the gateway, thus the case of uplink deadline-constrained single (non-redundant) set of paths (as in convergecast). Yet, we note the work can be extended to consider the downlink component, e.g., by considering additional routes traveling from the gateway to actuators in a deadline-constrained fashion, both for symmetric or asymmetric (uplink-downlink) routing, thus including the case of graph routing (as in [11]) with multiple redundant paths (and/or for multi-cast). We aim to further analyze these aspects in future research works.

6.2 Simulation Results

The average performance of 100 test cases for both the SP and MO methods under varying network topologies and workload conditions is reported next:

(*i*) Number of overlaps

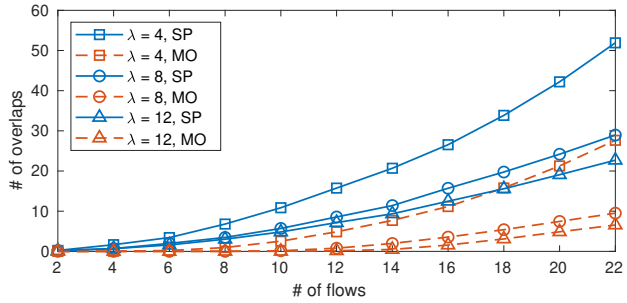

Figure 2 The average number of overlaps under varying number of flows $n \in [2, 22]$, and varying median vertex degree $\lambda = \{4, 8, 12\}$. $N = 66$ nodes, $m = 8$ channels.

(*ii*) Routes length

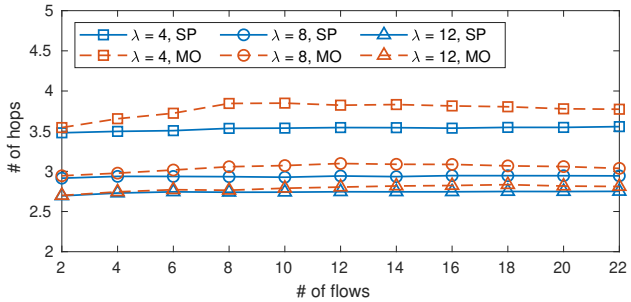

Figure 3 The average length of the routes under varying number of flows $n \in [2, 22]$, and varying median vertex degree $\lambda = \{4, 8, 12\}$. $N = 66$ nodes, $m = 8$ channels.

(*iii*) Channel contention

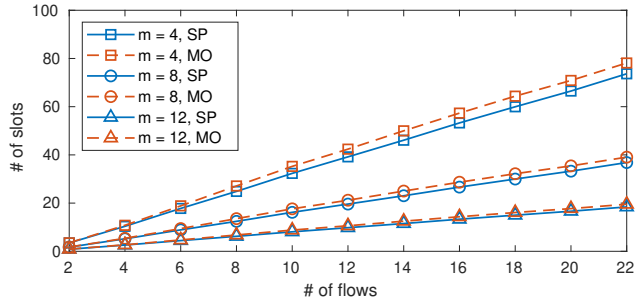

Figure 4 The average contention demand under varying number of flows $n \in [2, 22]$, and varying number of channels $m = \{4, 8, 12\}$. $N = 66$ nodes, median vertex degree $\lambda = 4$.

(*iv*) Transmission conflicts

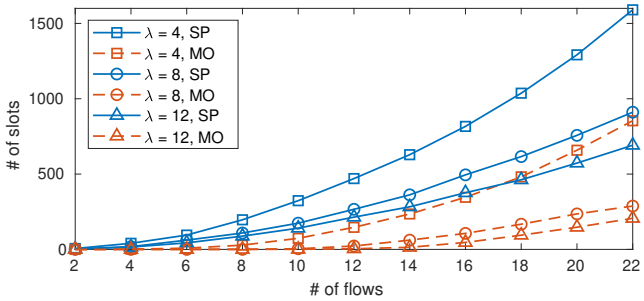

Figure 5 The average conflict demand under varying number of flows $n \in [2, 22]$, and varying median vertex degree $\lambda = \{4, 8, 12\}$. $N = 66$ nodes, $m = 8$ channels.

(*v*) Schedulability ratio

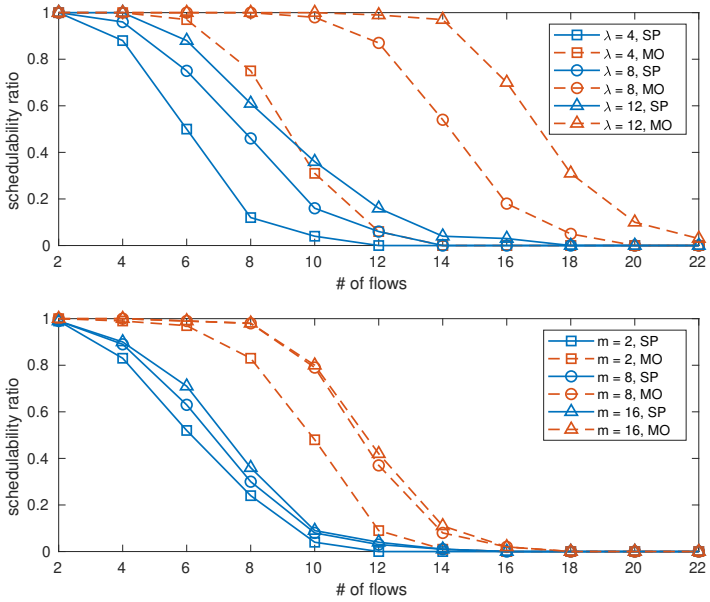

Figure 6 The schedulability ratio under varying number of flows $n \in [2, 22]$ and fixed number of nodes $N = 66$. On top, the case of varying median vertex degree $\lambda = \{4, 8, 12\}$ and $m = 8$ channels. On bottom, the case of varying number of channels $m = \{2, 8, 16\}$ and median vertex degree $\lambda = 4$.

6.3 Discussion

The performance comparison of MO and SP in terms the overall (average) number of overlaps confirms the main intuition behind the proposed method. The greedy heuristic search of MO albeit sub-optimal is able to effectively reduce the node-overlapping degree in up to half (and more) of the baseline (Fig. 2). This, depending of the number of flows and network density (or vertex degree). The exponential growth of the overlaps as function of the number of flows justifies the need for its mitigation. This trend is also observable in the growth imposed on transmission conflicts, which directly depends on the number of overlaps (Fig. 5). The compromise on the route lengths is marginal, whose impact is even negligible on more connected networks (Fig. 3). The influence on the channel contention is also minor, being discernible (in practice) only on networks with a lower number of active channels (Fig. 4). The benefits on the overall network schedulability are clear, regardless of the degree of connectivity (Fig. 6, top) or the available radio channels (Fig. 6, bottom), but suggesting superfluous effect on this latter parameter at a given point (see $m = 8$ and $m = 16$). All in all, the prospect for the proposed method is promising, both as a general technique to reduce the impact on conflict delays on wireless mesh network, or as an specific joint EDF scheduling and routing framework to provide real-time guarantees on TSCH-based networks.

7 Summary & Conclusion

We have developed an effective *real-time wireless routing* for TSCH-based WSANs with packet transmissions scheduled under an EDF policy. The approach based on a greedy heuristic for path-overlap minimization shown to be successful in reducing transmission conflicts and improving schedulability, while having marginal impact on contention and routes length. Simulation results under varying topologies and workload conditions revealed a remarkable dominance of our approach over the more common (hop-count) shortest path method. To conclude, we leverage on our prior work on schedulability analysis to frame both together as a novel *joint real-time scheduling and routing* framework for TSCH WSANs.

References

1 Giuliana Alderisi, Svetlana Girs, Lucia Lo Bello, Elisabeth Uhlemann, and Mats Björkman. Probabilistic scheduling and adaptive relaying for WirelessHART networks. In *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE, 2015.

2 Sanjoy Baruah, Vincenzo Bonifaci, Alberto Marchetti-Spaccamela, and Sebastian Stiller. Improved multiprocessor global schedulability analysis. *Real-Time Systems*, 46(1):3–24, 2010.

3 Keoma Brun-Laguna, Pascale Minet, and Yasuyuki Tanaka. Optimized scheduling for time-critical industrial IoT. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

4 Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. IEEE 802.15. 4e: A survey. *Computer Communications*, 88:1–24, 2016.

5 Miguel Gutiérrez Gaitán, Patrick M. Yomsi, Pedro M. Santos, and Luís Almeida. Work-in-progress: Assessing supply/demand-bound based schedulability tests for wireless sensor-actuator networks. In *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*, pages 1–4. IEEE, 2020.

6 Miguel Gutiérrez Gaitán and Patrick Meumeu Yomsi. FF-DBF-WIN: On the forced-forward demand-bound function analysis for wireless industrial networks. In *2018 30th Euromicro Conference on Real-Time Systems (ECRTS), Proceedings of the Work-in-Progress Session*, pages 13–15, 2018.

7 Miguel Gutiérrez Gaitán and Patrick Meumeu Yomsi. Multiprocessor scheduling meets the industrial wireless: A brief review. *U. Porto Journal of Engineering*, 5(1):59–76, 2019.

8 Harrison Kurunathan, Ricardo Severino, Anis Koubâa, and Eduardo Tovar. Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15. 4e. In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pages 1–9. IEEE, 2017.

9 Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proceedings of the IEEE*, 104(5):1013–1024, 2015.

10 Rahul Mangharam, Anthony Rowe, Raj Rajkumar, and Ryohei Suzuki. Voice over sensor networks. In *2006 27th IEEE International Real-Time Systems Symposium (RTSS)*, pages 291–302. IEEE, 2006.

11 Venkata Prashant Modekurthy, Dali Ismail, Mahbubur Rahman, and Abusayeed Saifullah. A utilization-based approach for schedulability analysis in wireless control systems. In *2018 IEEE International Conference on Industrial Internet (ICII)*, pages 49–58. IEEE, 2018.

12 Marcelo Nobre, Ivanovitch Silva, and Luiz Affonso Guedes. Routing and scheduling algorithms for WirelessHART networks: A survey. *Sensors*, 15(5):9703–9740, 2015.

13 Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. Real-time scheduling for WirelessHART networks. In *2010 31st IEEE Real-Time Systems Symposium (RTSS)*, pages 150–159. IEEE, 2010.

14 Stefano Scanzio, Mohammad Ghazi Vakili, Gianluca Cena, Claudio Giovanni Demartini, Bartolomeo Montrucchio, Adriano Valenzano, and Claudio Zunino. Wireless sensor networks and TSCH: A compromise between reliability, power consumption, and latency. *IEEE Access*, 8:167042–167058, 2020.

15 Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11):4724–4734, 2018.

16 Rasool Tavakoli, Majid Nabi, Twan Basten, and Kees Goossens. Topology management and TSCH scheduling for low-latency convergecast in in-vehicle WSNs. *IEEE Transactions on Industrial Informatics*, 15(2):1082–1093, 2018.

17 Federico Terraneo, Paolo Polidori, Alberto Leva, and William Fornaciari. TDMH-MAC: Real-time and multi-hop in the same wireless MAC. In *2018 IEEE Real-Time Systems Symposium (RTSS)*, pages 277–287. IEEE, 2018.

18 Mališa Vučinić, Tengfei Chang, Božidar Škrbić, Enis Kočan, Milica Pejanović-Djurišić, and Thomas Watteyne. Key performance indicators of the reference 6TiSCH implementation in Internet-of-Things scenarios. *IEEE Access*, 8:79147–79157, 2020.

19 Chengjie Wu, Dolvara Gunatilaka, Mo Sha, and Chenyang Lu. Real-time wireless routing for industrial internet of things. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 261–266. IEEE, 2018.

20 Changqing Xia, Xi Jin, and Peng Zeng. Resource analysis for wireless industrial networks. In *Proceedings of the 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pages 424–428. IEEE, 2016.