Path Collision-aware Real-time Link Scheduling for TSCH Wireless Networks

A. Darbandi and Myung Kyun Kim*
Department of Computer Engineering, University of Ulsan, Korea
93 Daehak-ro, Nam-gu, Ulsan (44610), Korea
[e-mail: mkkim@ulsan.ac.kr]
*Corresponding author: Myung Kyun Kim

Received November 5, 2018; revised February 3, 2019; accepted February 28, 2019; published September 30, 2019

Abstract

As low-power and low-rate WSNs are being widely used for industrial applications, the scheduling of such applications becomes a critical issue to guarantee meeting the stringent requirements of determinism and tight latencies. This paper studies the link scheduling problem for real-time industrial applications in time-slotted channel hopping (TSCH) networks. We propose a heuristic algorithm for centralized link scheduling referred to as path-collision aware least laxity first (PC-LLF) algorithm, which dynamically prioritizes the packets based on the laxity time to the end-to-end deadlines and the amount of collisions that messages might deal with along their designated paths to the destination device. We propose schedulability analysis of real-time applications scheduled under our prioritization approach over TSCH networks, based on the literature on real-time schedulability analysis of multiprocessors and distributed systems. We show that our methodology provides an improved schedulability condition with respect to the existing approaches. Performance evaluation studies quantify to quantify the performance of our proposed approach under a variety of scenarios.

Keywords: TSCH networks, IEEE 802.15.4e, centralized scheduling, schedulability.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(KNRF-2017030208).

http://doi.org/10.3837/tiis.2019.09.006 **ISSN : 1976-7277**

1. Introduction

Wireless Sensor Networks (WSNs) are one of the most promising solutions in the field of industrial communications [1]. The WSNs with low-power and low-rate sensor devices have been increasingly deployed for many industrial applications, including factory automation [2], distributed and process control [3]–[6] (such as smart detection of liquid/gas leakage, for radiation check, in smart grids and in smart buildings). In such real-time applications, end-to-end delays are constrained by upper bounds (i.e., deadlines), e.g., tens of milliseconds for discrete manufacturing, seconds for process control, and minutes for asset monitoring [8]. In addition, high scalability of the protocol ensures that the algorithms running inside the network scale well when large geographic areas need to be monitored.

The IEEE 802.15.4 standard [11] can be considered as the reference standard to offer physical and media access control (MAC) infrastructure to the low-power and low-rate WSNs. However, as the basic channel access control in the IEEE 802.15.4 standard is through CSMA-CA, many previous works [12]-[14] show that this protocol is improper for novel real-time applications requiring predictable and robust communication. Hence, the IEEE802.15e Working Group defined the IEEE 802.15.4e amendment [15] in 2012, which introduces the MAC protocol named Time Slotted Channel Hopping (TSCH). In TSCH, the time slotted channel access scheme along with the multi-channel and frequency hopping potentials ensure deterministic latency and high resistance against interferences and multipath fading. In this paper, our scheduling efforts target the design algorithms and protocols over TSCH networks in order to comply with stringent requirements for determinism and short latencies.

Scheduling for TSCH networks has been investigated since the standard was published in 2012, and the results can be categorized into two main classes: The first class includes approaches that deal with distributed scheduling problems in TSCH networks [16]-[18]. In these works, since the focus is on continuous collision-free link establishment, no attention was devoted to traffic prioritization based on latency constraints. The second class includes approaches that study the problem of centralized TSCH scheduling [19]-[21]. These studies prioritized a sensor device solely based on its generated traffic and the amount of traffic passing through that device. The packet latency was not recognized as a major limitation. Hence, these approaches limit the overall end-to-end latency performance of industrial applications and cannot guarantee schedulability in terms of end-to-end deadlines.

The main contribution of our paper is a novel heuristic scheduling algorithm, that prioritizes each packet transmission dynamically, based on its laxity (i.e., the remaining time before the end-to-end deadline) and the amount of latency imposed by the collisions that it might deal with throughout its path to the destination device. Hence, this algorithm is referred to as Path Collision-aware Least Laxity First (PC-LLF) algorithm. Since our prioritization strategy dynamically takes in to account packets laxity and it dynamically predicts path latency, it can return a good quality feasible solution that minimize packet lateness, defined as the time phase between message arrival time at the destination device and its end-to-end deadline. As our extensive set of simulations show, the prioritization offered by our approach is a significant advantage in handling end-to-end latencies and dynamic changes in network workloads, compared to existing approaches in [22] and [26].

To guarantee schedulability of a given problem set, we introduce an upper bound delay for each packet transmission, based on the literature on real-time schedulability analysis of multiprocessors and distributed systems [28]-[30]. To do so, we approximate some pessimism on the worst-case contribution of high priority packets to the response time of a packet transmission. When the upper bound delay of each packet transmission is less than its deadline, our condition guarantees the schedulability of the problem set. Despite the fact that our upper bound is pessimistic, our evaluations show that it is closer to the schedulable ratio performance by heuristic algorithms, compared with the upper bound delay defined in [22].

The remainder of this paper is organized as follows. Section 2 describes the related works. Section 3 introduces system and application models and the problem formulation. Section 4 describes our proposed approach in detail. Section 5 presents evaluation results, with a conclusion following in Section 6.

2. Related Work

Conceptually, TSCH is not a new technique as it was already used in industrial networking technologies such as WirelessHART and ISA100.11a [25]. Many papers have targeted centralized scheduling of end-to-end packet communications over these protocols, mainly to deliver high throughput and low energy expenditure. In many cases, they approach the problem by introducing some assumptions to simplify the analysis. For example, [23] proposes a scheduling algorithm for tree-graphed WirelessHART networks. The authors assume that, each field single-buffer device generates a single data packet at the beginning of each superframe, and it must be delivered before the end of superframe. Under these conditions, they do not consider the case of periodic flows with generation rates and deadlines which may differ based on the application QoS constraints.

Strategies for centralized scheduling over TSCH networks have been studied in [19] and [20]. These approaches dynamically prioritize each sensor fieled device, based on the amount of traffic it generates and that it forwards from child nodes to parent nodes. Basically, a node with a large amount of packets in its sub-tree achieves high priority consecutively, until the amount of packets in all sub-trees is balanced. Hence, the authors do not emphasize the packet latency problem and, consequently, they do not investigate its causes or propose any solution to it. For example, continuously scheduling a node with a large sub-tree, may lead to packet latency and eventually packet drops (since WSNs are resource constrained) in small sub-trees.

The study in [10] proposes a centralized scheduling algorithm for a generalized WirelessHART network model, where nodes generate flows of packets at different rates. The authors prioritize each packet, based on rate monotonic strategy that assigns higher priority to packets with short generation rates. However, the rate monotonic is a static prioritization mechanism, which does not boost the priority of packets as they approach their deadlines or as they deal with high amount of collisions in their designated paths. In this paper, we thoroughly investigate the collisions as fundamental reasons for the packet latency problem. We show that the packet latency can be mitigated by appropriate packet prioritization.

In contrast to most of the previous works, our proposed analysis is based on a generalized network topology that supports a more realistic application domain. In our system model, every field device generates flows with different timing requirements (i.e., release times, periods and deadlines). Under this system model, the study in [22] proposes an optimization algorithm based on branch and bound algorithm. Each branch represents a partial schedule that must progress through further branching, in order to achieve a final feasible or infeasible solution. Authors propose an upper bound for packets latency to prune infeasible branches of the search tree. In the present paper, we show that, this upper bound is highly pessimistic and cannot guarantee schedulability of a large set of problems. We improve this schedulability condition by a tight upper bound estimation for packets delay, based on the literature on real-time schedulability analysis of multiprocessors and distributed systems [28]-[30]. Our evaluations show that, our schedulability condition efficiently increases the maximum schedulable utilization.

Since the optimization approaches have unreasonable amount of runtime, the studies by [22] and [26] propose heuristic algorithms for scheduling packet transmissions. The heuristic in [22] dynamically prioritizes the released transmissions on each time slot, based on the amount of conflicts it must deal with at the current field device (which is routing the packet toward the destination), and the amount of remaining time before the deadline. We show here that, our algorithm which carries the amount of collisions the packet will deal with along the remaining path, will achieve higher amount of success ratio for scheduling a given problem.

The study in [26], proposes an iterative Hop-wise Scheduling Algorithm (H-SA). At *k*-th iteration, H-SA schedules the transmissions on the *k*-th links on the routes of all the flows, during which the conflict relationships and the priorities must be accounted [26]. Authors define the priority of each transmission based on its remaining time before the deadline and the remaining number of hops to the destination. The flows with more hops away from their destinations and shorter deadlines are assigned larger weights [26]. Section 5 evaluates the performance of this approach compared with our proposed heuristic algorithm.

3. System Model and Problem Formulation

3.1 System Model

In TSCH, the communication takes place identically in periodic cycles called slotframes, as seen in **Fig. 1**. Each slotframe is divided into a fixed number of equal-sized time slots: N_S. A single time slot is long enough for the transmitter to send a maximum-length packet, and for the receiver to send back an acknowledgment. The total number of timeslots that have elapsed since the start of the network or an arbitrary start time determined by the Personal Area Network (PAN) coordinator is called the Absolute Slot Number (ASN). It increments globally in the network at the beginning of each time slot, and is used globally by devices as the slot counter. Initially, $n_{ch} \leq 16$ different channels are available for communication. Each channel is identified by a *channel-offset*, which is an integer value in the range [0, 15]. Within a time slot, the sender starts transmitting *TsTxOffset* μs after the beginning of time slot, and the receiver starts listening to the channel *guardTime* μs before. This mechanism requires that devices are never unsynchronized for more than *guardTime* μs, so as to be able to communicate. As each device supports communications on multiple channels, multiple node pairs communicate at the same time slots using different channel offsets, thereby increasing the network capacity. In TSCH, a communication cell is defined as the pairwise assignment of a directed communication between devices in a given timeslot for a given channel offset. Therefore, a communication cell between communicating devices can be represented by a pair (*time-slot*, *channel-offset*). Due to difficulty in detecting interference between nodes and the variability of interference patterns [24], and for simplicity in our schedulability analysis, we allow only one transmission in each communication cell across the entire network. This means that, a communication cell (*time-slot*, *channel-offset*) is dedicated to a single communication link $v_p v_q$. Hence, the maximum number of concurrent transmissions in any time slot cannot exceed the number of available channels.

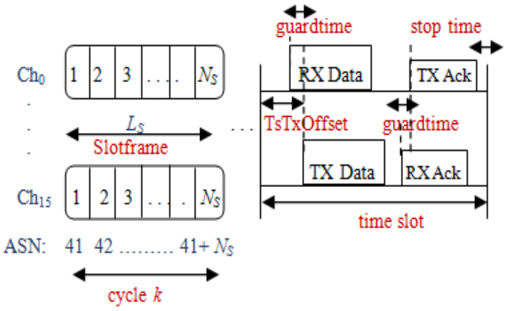


Fig. 1. A TSCH superframe consisting of N_S time slots and the transmission structure in a slotframe.

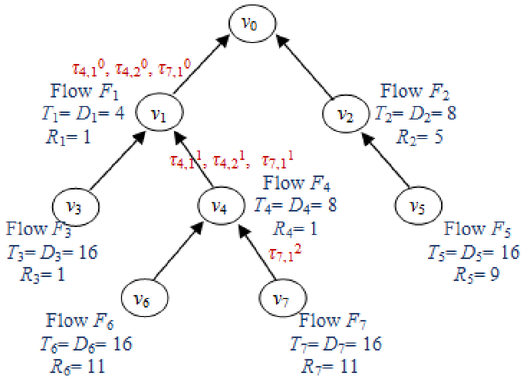


Fig. 2. An example network graph with 7 sensor devices sending information to the sink node, v_0.

3.2 Application Model

In this paper, the TSCH network forms a mesh network modeled as a Directed Acyclic Graph (DAG) G=(V,E), where V={v_1, v_2,…, v_N} is the set of all field devices and arcs in E are communication links, as seen in **Fig. 2**. Each link in E is identified by an ordered pair of nodes, e.g., v_pv_q where v_p and v_q are the transmitting receiving nodes, respectively. Two links duplex conflict with each other if they share a common sender or receiver. The duplex conflicting links cannot share the same time slot. We shall denote by $CNF(v_pv_q)$ the set of all links duplex conflicting with v_pv_q, e.g., $CNF(v_4v_1)$={v_1v_0, v_3v_1, v_6v_4, v_7v_4} in **Fig. 2**. Two links interfere with each other if receiver or sender of one link can overhear transmissions by sender of the other link. The interfering links cannot share the same communication cell with simultaneous transmissions. We shall denote by $INF(v_pv_q)$ the set of all links interfering with v_pv_q.

Designated device v_0 acts as the central controller, and each field device is either a sensor or an actuator or both. A sensor node periodically collects sensing information and sends the generated data to the central controller, an actuator device periodically receives the optimum course of actions from the central controller, and both devices support multi-hop routing. Each field device is assumed to be equipped with a half-duplex radio transceiver that cannot transmit and receive concurrently. Each device supports communications on multiple non-overlapping channels, but can only send or receive on one channel in a given time slot.

Each source node v_{i} generates a flow F_i, and F={F_1, F_2,…, F_M} is the set of all flows to be scheduled. Each flow F_i is initially released R_i time units after beginning of the slotframe, and is periodically activated with period T_i, as seen in **Fig. 3**. The j-th invocation of F_i in a slotframe generates the j-th packet denoted by $p_{i,j}$, its release time is $r_{i,j}$= R_i + (j-1)·T_i, its start of transmission is denoted by $s_{i,j}$, and its reception time at the destination is denoted by $f_{i,j}$. The transmission of $p_{i,j}$ along the edge v_pv_q, where v_p is k hops away from the destination node, is denoted by $\tau_{i,j}{}^{k}$. In **Fig. 2**, the transmissions of packets generated by flows F_4 and F_7 are inserted on the corresponding links. We shall denote by U the set of all transmissions of all packets that must be scheduled in the network. In addition, the length of the slotframe, denoted by L_S, equals LCM (Least Common Multiple) of the periods of all the flows in F.

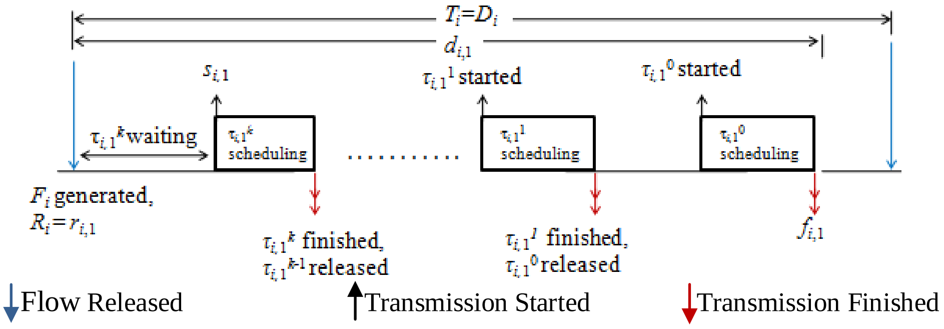

T_i=D_i
$d_{i,1}$
$s_{i,1}$
$\tau_{i,1}^1$ started
$\tau_{i,1}^0$ started
$\tau_{i,1}^k$ waiting
$\tau_{i,1}^k$ scheduling
$\tau_{i,1}^1$ scheduling
$\tau_{i,1}^0$ scheduling
F_i generated, R_i=$r_{i,1}$
$\tau_{i,1}^k$ finished, $\tau_{i,1}^{k-1}$ released
$\tau_{i,1}^1$ finished, $\tau_{i,1}^0$ released
$f_{i,1}$
Flow Released
Transmission Started
Transmission Finished

Fig. 3. System Parameters. The release times, scheduling start and finish times of transmissions of first packet generated by flow F_i, and the respective end-to-end deadline and delay.

Flow F_i has an end-to-end deadline denoted by D_i, which indicates that each packet $p_{i,j}$ generated by F_i must arrive at the destination before $r_{i,j} + D_i$. End-to-end delay for $p_{i,j}$ is denoted by $d_{i,j}$, and is the time interval between the release time of $p_{i,j}$ at the source node until its arrival time at the destination node (i.e., $d_{i,j}= f_{i,j} - r_{i,j}$). In order to meet the end-to-end deadline of the flow F_i, we must have $d_{i,j} \leq D_i$, for each j-th packet. In **Fig. 2**, the end-to-end deadline of each flow equals its period, meaning that each packet must arrive at the destination before the release time of the next packet. The values are in terms of the number of slots.

3.3 Problem Formulation

The problem defined in this paper is to find a feasible schedule for all transmissions of each flow along their respective designated path over IEEE 802.15.4e TSCH protocol, in order to minimize messages lateness. Given the TSCH network G and the flow set F, the periods in T and release times in R, the set of end-to-end deadline of each flow in D and the set of transmissions of each packet U, we would like to solve the following problem:

$$\arg\min_{\Gamma} \quad E(G,F,D,R,T,U,n_{ch}) \tag{1}$$

$$ASN(\tau_{i,j}^{k}) < ASN(\tau_{i,j}^{k+1}), \forall\, \tau_{i,j}^{k} \tag{2}$$

$$D_i \geq d_{i,j}, \forall p_{i,j} \tag{3}$$

$$X_{p,q}\left(t,ch\right) + X_{m,n}\left(t,ch\right) \leq 1, \forall v_m v_n \in INF(v_p v_q) \tag{4}$$

$$Y_{p,q}\left(t\right) + Y_{m,n}\left(t\right) \leq 1, \forall v_m v_n \in CNF(v_p v_q) \tag{5}$$

where $E(G,F,D,R,T,U,n_{ch})$ represents messages lateness or the schedule length, when schedule Γ is defined. The constraints in Eq. 2 and Eq. 3 state that the transmission $\tau_{i,j}^{k}$ must occur earlier than $\tau_{i,j}^{k+1}$, and the end-to-end delays must be restricted to end-to-end deadlines, respectively. The constraints in Eq. 4 and Eq. 5 state that two interfering links cannot be scheduled in the same communication cell (t, ch) and two conflicting links cannot be scheduled in the same time slot t. $X_{p,q}(t,ch)$ is a binary decision variable that is set to 1 when the transmission on the link $v_p v_q$ is allocated to the communication cell (t, ch), and 0 otherwise. Similarly, $Y_{p,q}(t)$ is a binary decision variable that is set to 1 when the transmission on the link $v_p v_q$ is allocated to the time slot t, and 0 otherwise.

4. Scheduling Solution

This section describes our proposed scheduling approach. The major components of our contribution are as follows: First in Section 4.1, we define the priority for each transmission of each packet. In Section 4.2, we present our heuristic scheduling algorithm. In Section 4.3, we elaborate on necessary and sufficient schedulability condition under our prioritization approach. The notations used in this section are listed in **Table 1**.

Table 1. Notations

Symbol	Meaning	Symbol	Meaning
$EST(\tau_{i,j}^{k}, S)$	Earliest start time for $\tau_{i,j}^{k}$ in slot S.	$LST(\tau_{i,j}^{k}, S)$	Latest start time for $\tau_{i,j}^{k}$ in slot S.
$n_{cnf}(\tau_{i,j}^{k}, S)$	Number of transmissions duplex conflicting with $\tau_{i,j}^{k}$ in time slot S.	$n_{inf}(\tau_{i,j}^{k}, S)$	Number of transmissions interfering with $\tau_{i,j}^{k}$ in time slot S.
$d_{cnf}^{max}(\tau_{i,j}^{k},S)$	Upper bound of $\tau_{i,j}^{k}$'s delay in time slot S, imposed by all duplex conflicting transmissions.	$d_{inf}^{max}(\tau_{i,j}^{k},S)$	Upper bound of $\tau_{i,j}^{k}$'s delay in time slot S, imposed by all interfering transmissions.
$D_{cnf}^{avg}(\tau_{i,j}^{k},S)$	Average scheduling latency of $\tau_{i,j}^{k}$, $\tau_{i,j}^{k-1}$, .. , $\tau_{i,j}^{1}$, $\tau_{i,j}^{0}$ in time slot S, imposed by all duplex conflicting transmissions.	$D_{inf}^{avg}(\tau_{i,j}^{k},S)$	Average scheduling latency of $\tau_{i,j}^{k}$, $\tau_{i,j}^{k-1}$, .. , $\tau_{i,j}^{1}$, $\tau_{i,j}^{0}$ in time slot S, imposed by all interfering transmissions.
$Delay_{UB}^{hp\text{-}cnf}$ $(\tau_{i,j}^{k}, t^{m})$	Upper bound of $\tau_{i,j}^{k}$'s delay, imposed by duplex conflicting transmissions with a higher priority, released at $(t^{m-1}, t^{m}]$.	$Delay_{UB}^{hp\text{-}inf}$ $(\tau_{i,j}^{k}, S)$	Upper bound of $\tau_{i,j}^{k}$'s delay, imposed by interfering transmissions with higher priority, released at $(t^{m-1}, t^{m}]$.
$hp\text{-}cnf$ $(\tau_{i,j}^{k}, t^{m})$	Set of transmissions duplex conflicting with $\tau_{i,j}^{k}$ and a higher priority released at $(t^{m-1}, t^{m}]$.	$hp\text{-}inf$ $(\tau_{i,j}^{k}, t^{m})$	Set of transmissions interfering with $\tau_{i,j}^{k}$ and a higher priority released at $(t^{m-1}, t^{m}]$.

4.1 Priority Assignment

Our proposed prioritization approach dynamically updates priority of each transmission $\tau_{i,j}^{k}$ in each time slot, based on two main parameters: the remaining time to the end-to-end deadline and the delay posed by the average amount of collisions that the packet might deal with at each remaining intermediate node. The intuition is that, a high amount of collisions means that the transmission is harder to schedule within a limited laxity to deadline and it corresponds to a high priority, whereas few collisions and a longer deadline usually implies larger space for the transmission in which to be flexibly scheduled.

In order to include deadline laxity at slot S in our prioritization, we define a time window for each $\tau_{i,j}^{k}$, denoted by $TW(\tau_{i,j}^{k}, S)$, which is the time phase from the earliest time slot to the latest time slot in which $\tau_{i,j}^{k}$ can start its transmission, in slot S. The earliest start time and the latest start time for $\tau_{i,j}^{k}$ in slot S are denoted by $EST(\tau_{i,j}^{k}, S)$ and $LST(\tau_{i,j}^{k}, S)$, respectively, and are computed as follows:

$$EST(\tau_{i,j}^{k}, S) = r_{i,j}^{k} + pre_{i,j}^{k} + \max\left(S - r_{i,j}^{k}, 0\right) \tag{6}$$

$$LST(\tau_{i,j}^{k}, S) = r_{i,j} + D_{i} - k \tag{7}$$

where $\tau_{i,j}^{k}$ is k hops away from its destination, and $pre_{i,j}^{k}$ hops away from its source node.

To define $TW(\tau_{i,j}^{k}, S)$ for each $\tau_{i,j}^{k}$ in time slot S, we determine $EST(\tau_{i,j}^{k}, S)$ by traversing DAG G beginning from the source node toward the destination node. As $EST(\tau_{i,j}^{k}, S)$ depends on the current time slot S, it must be dynamically re-computed in each time slot. $LST(\tau_{i,j}^{k}, S)$ can be computed by traversing DAG G beginning from the destination node toward the source node. However, because $LST(\tau_{i,j}^{k}, S)$ does not depend on the current time slot, it has a fixed value over time.

Table 2. $TW(\tau_{i,j}^{k},1)$, $d_{cnf}^{max}(\tau_{i,j}^{k},1)$ and $D_{cnf}^{avg}(\tau_{i,j}^{k},1)$ for each packet transmission in **Fig. 2**.

	$TW(\tau_{i,j}^{k},1)$	$d_{cnf}^{max}(\tau_{i,j}^{k},1)$	$D_{cnf}^{avg}(\tau_{i,j}^{k},1)$	$Pr(\tau_{i,j}^{k},1)$		$TW(\tau_{i,j}^{k},1)$	$d_{cnf}^{max}(\tau_{i,j}^{k},1)$	$D_{cnf}^{avg}(\tau_{i,j}^{k},1)$	$Pr(\tau_{i,j}^{k},1)$
$\tau_{7,1}^{2}$	[11,16:8]	4	10.66	3.34	$\tau_{7,1}^{1}$	[12,16:9]	13	14	0
$\tau_{7,1}^{0}$	[13,16:10]	15	15	-1					
$\tau_{6,1}^{2}$	[11,16:8]	4	10.66	3.34	$\tau_{6,1}^{1}$	[12,16:9]	13	14	0
$\tau_{6,1}^{0}$	[13,16:10]	15	15	-1					
$\tau_{5,1}^{1}$	[9,16:7]	2	6.5	8.5	$\tau_{5,1}^{0}$	[10,16:8]	11	11	4
$\tau_{4,1}^{1}$	[1,7]	10	10.5	-3.5	$\tau_{4,1}^{0}$	[2,8]	11	10	-3
$\tau_{4,2}^{1}$	[9,15]	10	10.5	-3.5	$\tau_{4,2}^{0}$	[10,16]	11	10	-3
$\tau_{3,1}^{1}$	[1,15]	14	14.5	0.5	$\tau_{3,1}^{0}$	[2,16]	15	15	0
$\tau_{2,1}^{0}$	[5,12]	9	9	-1	$\tau_{2,2}^{0}$	[13,16:4]	9	9	-1
$\tau_{1,1}^{0}$	[1,4]	10	10	-6	$\tau_{1,2}^{0}$	[5,8]	10	10	-6
$\tau_{1,3}^{0}$	[9,12]	10	10	-6	$\tau_{1,4}^{0}$	[13,16]	10	10	-6

In **Table 2**, we present the transmissions of the packets generated by each flow, over the network given in **Fig. 2**. Flow F_4 generates two packets, $\tau_{4,1}$ and $\tau_{4,2}$, within one slotframe with a length of 16 time slots. Time window $[x,16{:}y]$ denotes that, the time window starts from time slot x in the current slotframe and it ends at slot y in the next slotframe.

The second parameter that accounts for prioritizing $\tau_{i,j}^{k}$ in the current time slot S is the average delay posed by the collisions that each remaining transmissions (i.e., $\tau_{i,j}^{k}$, $\tau_{i,j}^{k-1}$, $\tau_{i,j}^{k-2}$, … , $\tau_{i,j}^{0}$) might deal with in time slot S. To do so, we consider two types of collisions: duplex conflicting collisions and interfering collisions.

A duplex conflicting collision is imposed by a transmission $\tau_{m,n}^{l}$ existed on a conflicting link in $CNF(v_p v_q)$, where its lifetime (i.e., $TW(\tau_{m,n}^{l}, S)$) overlaps with $TW(\tau_{i,j}^{k}, S)$. Let $n_{cnf}(\tau_{i,j}^{k},S)$ denotes the amount of transmissions duplex conflicting with $\tau_{i,j}^{k}$, in time slot S. Then, we define $d_{cnf}^{max}(\tau_{i,j}^{k},S)$ as the upper bound of the scheduling latency imposed on $\tau_{i,j}^{k}$ in the time slot S, imposed by $n_{cnf}(\tau_{i,j}^{k},S)$ conflicting transmissions. In the worst case, all these conflicting transmissions might delay the scheduling of $\tau_{i,j}^{k}$ to $n_{cnf}(\tau_{i,j}^{k},S)$ time slots later. Thus, we have $d_{cnf}^{max}(\tau_{i,j}^{k},S) = n_{cnf}(\tau_{i,j}^{k},S)$.

An interfering collision is posed by transmission $\tau_{r,s}^{q}$ existing in an interfering link in $INF(v_p v_q)$, where its lifetime $TW(\tau_{r,s}^{q}, S)$) overlaps with $TW(\tau_{i,j}^{k}, S)$. We shall also denote by $n_{inf}(\tau_{i,j}^{k},S)$ the amount of transmissions interfering with $\tau_{i,j}^{k}$ in time slot S. Then, again we define $d_{inf}^{max}(\tau_{i,j}^{k},S)$ to be the maximum delay that $\tau_{i,j}^{k}$ might deal with in the time slot S, due to the maximum amount of interfering transmissions. Based on the delay analysis in [28, 29], the equation $d_{inf}^{\max}(\tau_{i,j}^{k},S)=\left\lfloor n_{inf}(\tau_{i,j}^{k},S) \big/ n_{ch} \right\rfloor$ determines an upper bound on the amount of scheduling latency that $n_{inf}(\tau_{i,j}^{k},S)$ interfering transmissions may impose on scheduling $\tau_{i,j}^{k}$.

We shall now study the average latency imposed on the packet $\tau_{i,j}$ due to duplex conflicting transmissions and interfering transmissions, along the remaining path, in time slot S. These average delay are denoted by by $D_{cnf}^{avg}(\tau_{i,j}^{k},S)$ and $D_{inf}^{avg}(\tau_{i,j}^{k},S)$, respectively. The remaining path consists of current transmission $\tau_{i,j}^{k}$ and future transmissions $\tau_{i,j}^{k-1}$, $\tau_{i,j}^{k-2}$, …, $\tau_{i,j}^{0}$. We note that, $D_{cnf}^{avg}(\tau_{i,j}^{k},S)$ and $D_{inf}^{avg}(\tau_{i,j}^{k},S)$ are dynamic variables, which their value change in each time slot. Because, the amount of conflicts on a specific link in time slot S depends on the newly arrived transmissions, and the departed transmissions in previous time slots. $D_{cnf}^{avg}(\tau_{i,j}^{k},S)$ and $D_{inf}^{avg}(\tau_{i,j}^{k},S)$ are computed as follows:

$$D_{cnf}^{avg}(\tau_{i,j}^{k},S)=\sum\nolimits_{m\le k} d_{cnf}^{\max}(\tau_{i,j}^{m},S)\Big/k \tag{8}$$

$$D_{inf}^{avg}(\tau_{i,j}^{k},S)=\sum\nolimits_{l<k} d_{inf}^{\max}(\tau_{i,j}^{l},S)\Big/k \tag{9}$$

With the definitions in Eqs.6-9, the priority of transmission $\tau_{i,j}^{k}$ is defined by:

$$Pr(\tau_{i,j}{}^{k},S)=\left|TW(\tau_{i,j}{}^{k},S)\right|-\max\left(D_{cnf}^{avg}(\tau_{i,j}^{k},S),D_{inf}^{avg}(\tau_{i,j}^{k},S)\right) \tag{10}$$

where $|TW(\tau_{i,j}^{k},S)|$ is the number of available time slots in which $\tau_{i,j}^{k}$ can be scheduled, after time slot S. For example, in **Table 2** we present $d_{cnf}^{max}(\tau_{i,j}^{k},1)$, $D_{cnf}^{avg}(\tau_{i,j}^{k},1)$ and $Pr(\tau_{i,j}^{k},1)$ for the packets generated by each flow, in the network given in **Fig. 2**.

4.2 PC-LLF: Path Conflict Aware Least Laxity First Algorithm

Our proposed PC-LLF scheduling algorithm is defined as follows. The input parameters of the PC-LLF algorithm are G, F, D, R, T, U, n_{ch}, and the output is a feasible scheduling solution.

Algorithm 1: *PC-LLF* $(G, F, D, R, T, U, n_{ch})$

```
1:  Define the current time slot $S$=1, and define $Pr(\tau_{i,j}^{k},1)$ for each $\tau_{i,j}^{k}$.
2: Define $Released(S)$ as the set of released transmissions in the first time slot.
3. While (not all the transmissions are scheduled) do
   3.1: If $S\neq 1$, Modify $Pr(\tau_{i,j}^{k}, S)$ for all transmissions with Eqs. 6-10.
   3.2: For each $\tau_{i,j}^{k}$ in $Released(S)$ check the schedulability condition by Eq. 13.
   3.3: If there is a transmission that is not schedulable
         3.3.1. Return “unschedulable”, Exit.
   3.4 Else
         3.4.1. Initiate the channel counter, $m$=0.
         3.4.2 While ($m \le n_{ch}$) do
               3.4.2.1 Select a released transmission with the highest priority, call it
               $\tau$*. When several transmissions have the same priority, assign a
               higher priority to a transmission with the smallest length for its time
               window.
               3.4.3.2 Assign $\tau$* to current slot $S$ on current channel $m$.
               3.4.3.3 Remove $\tau$* and all transmissions conflicting with $\tau$* from
               $Released(S)$.
               3.4.4 Go to the next channel offset, $m$=$m$+1.
         3.4.3 End while.
         3.4.4 Go to the next slot, $S$=$S$+1.
         3.4.5 Modify the set Released($S$).
   3.5. End If
5. End While
6. Return the schedule.
```

In the first step, PC-LLF defines *Released*(S) to be the set of all released transmissions in the current time slot S, starting from first slot. Then, PC-LLF modifies the priority of all transmissions for that slot with Eqs. 6-10, in line 3.1. By Eq. 13 in the next section, we shall ensure the schedulability of each un-scheduled transmission. PC-LLF applies the delay analysis described in the next section, in order to compute the upper bound delay for the worst case response time of the transmissions in *Released*(S) at each iteration. If the upper bound schedulability analysis recognizes an un-schedulable transmission in *Released*(S), then PC-LLF exits the algorithm in line 3.3.1. Otherwise, PC-LLF continues to schedule the highest priority transmissions in the current time slot in line 3.4.2.

For the next time slot, PC-LLF modifies *Released*(S) by deleting all the scheduled transmissions and by adding newly released transmissions in the next time slot, it modifies the priority of transmissions, and repeats the scheduling procedure for the next time slot.

As we stated in Section 3.1, the length of the slotframe, denoted by L_S, equals LCM of periods for all the flows in F. However, the scheduling of a flow F_i with release time R_i>1 may not be completed within one slotframe. In this case, the scheduling of the flows must continue until the scheduling of all generated packets within one slotframe cycle has been computed. In **Fig. 2**, PC-LLF starts considering flow F_7 for scheduling from the eleventh time slot, until at most the tenth time slot in the next slotframe.

4.3 Upper Bound Delay Analysis

In this section, we introduce an upper bound for the worst case response time of each transmission $\tau_{i,j}^{k}$, under our scheduling strategy. The worst case response time for $\tau_{i,j}^{k}$ is the maximum scheduling latency that $\tau_{i,j}^{k}$ might cope with after its release time. This upper bound delay is denoted by $Delay_{UB}(\tau_{i,j}^{k})$. When the inequality $Delay_{UB}(\tau_{i,j}^{k})<|\mathrm{TW}(\tau_{i,j}^{k})|$ holds for all transmissions, the flow set is guaranteed to be schedulable.

The main reasoning behind the upper bound for the worst case delay analysis is borrowed from the literature on real-time scheduling on a global multiprocessor platform. Each channel in the network can be viewed as a processor, and each flow can be viewed as a task that is executed on a processor with period T_i, deadline D_i, and an execution time equal to the slot length [28]. However, unlike multiprocessor scheduling, our analysis is constrained by transmission conflicts, and accounting for the non-preemptability of the packets as additional blocking time [30]. Our analysis is mainly inspired from the schedulability analysis of multiprocessors and distributed systems in [30].

To define $Delay_{UB}(\tau_{i,j}^{k})$, we find the upper bound on the maximum contribution of each transmission to the worst-case response time of $\tau_{i,j}^{k}$. To do so, we only consider the set of transmissions where their respective time windows, defined at the beginning of the scheduling cycle, i.e., $TW(\tau_{m,n}^{l},1)$, intersect with $TW(\tau_{i,j}^{k},1)$. Such a transmission is called an overlapping transmission. Note that, the time windows defined in the first time slot have the maximum length. This fact helps us to account for the maximum amount of transmissions overlapping with $\tau_{i,j}^{k}$. To define the maximum contribution of each transmission, we categorize each overlapping transmission into one of the following sets:

- Set 1: This set consists of all overlapping transmissions where the time window of each transmission covers the time instance $t^0 = EST(\tau_{i,j}^{k},1)$. The maximum contribution of this set of transmissions to the scheduling latency of $\tau_{i,j}^{k}$ occurs when their scheduling time is delayed to the time instance t^0, so that they are still available for scheduling when $\tau_{i,j}^{k}$ is released. For example, even though $\tau_{5,1}^{0}$ can be released in the previous slotframe, it can still be available for scheduling in first slot of the current slotframe.
- Set 2: This set consists of the maximum amount of overlapping transmissions that can be available after $t^0 = EST(\tau_{i,j}^{k},1)$ and that may contribute to the response time of $\tau_{i,j}^{k}$. The maximum contribution of this set of transmissions to the scheduling latency of $\tau_{i,j}^{k}$ occurs when they are released at the beginning of their time windows. This is due to the fact that, the larger the releasing delay they have, the greater the probability they will be released after scheduling $\tau_{i,j}^{k}$.

We now calculate the contribution of the transmissions in Set 1, when they are released at $t^0 = EST(\tau_{i,j}{}^k, 1)$. This delay is denoted by $Delay^0{}_{UB}(\tau_{i,j}{}^k, t^0)$ and can be computed as follows:

$$Delay_{UB}{}^0(\tau_{i,j}{}^k, t^0) = \max\left(Delay_{UB}^{hp-cnf}\left(\tau_{i,j}{}^k, t^0\right), Delay_{UB}^{hp-inf}\left(\tau_{i,j}{}^k, t^0\right)\right) \tag{11}$$

where $Delay_{UB}{}^{hp\text{-}cnf}(\tau_{i,j}{}^k, t^0)$ and $Delay_{UB}{}^{hp\text{-}inf}(\tau_{i,j}{}^k, t^0)$ compute the upper bound delays imposed by transmissions in Set 1 that induce duplex conflicting collisions and interference collisions with $\tau_{i,j}{}^k$, respectively, and that have a higher priority than $\tau_{i,j}{}^k$. $Delay_{UB}{}^{hp\text{-}cnf}(\tau_{i,j}{}^k, t) = |hp_{cnf}(\tau_{i,j}{}^k, t)|$ and $Delay_{UB}{}^{hp\text{-}inf}(\tau_{i,j}{}^k, t) = \left\lfloor \left|hp_{inf}(\tau_{i,j}^k)\right| \middle/ n_{ch} \right\rfloor$, where $hp_{cnf}(\tau_{i,j}{}^k, t)$ and $hp_{inf}(\tau_{i,j}{}^k, t)$ are the set of transmissions duplex conflicting with $\tau_{i,j}{}^k$ and the set of transmissions interfering with $\tau_{i,j}{}^k$, respectively, which receive a higher priority in time slot t.

In order to calculate the contribution of transmissions in Set 2, we apply a recursive equation as follows:

$$Delay_{UB}^m(\tau_{i,j}{}^k, t^m) = Delay_{UB}^{m-1}(\tau_{i,j}{}^k, t^{m-1}) + \max\left(Delay_{UB}^{hp-cnf}\left(\tau_{i,j}{}^k, t^m\right), Delay_{UB}^{hp-inf}\left(\tau_{i,j}{}^k, t^m\right)\right) \tag{12}$$

where $t^m = Delay^{m\text{-}1}{}_{UB}(\tau_{i,j}{}^k, t^{m\text{-}1})$. At the m^{th} iteration, $Delay^m{}_{UB}(\tau_{i,j}{}^k, t^{m\text{-}1})$ adds to $Delay^{m\text{-}1}{}_{UB}(\tau_{i,j}{}^k, t^{m\text{-}1})$ the contribution of transmissions that are released in the time interval $(t^{m\text{-}1}, t^m]$. The iterations over Eq. 12 stop when stable solution is achieved, i.e., we have $Delay^{m\text{-}1}{}_{UB}(\tau_{i,j}{}^k, t) = Delay^m{}_{UB}(\tau_{i,j}{}^k, t)$. This means that, at the time instance $Delay^{m\text{-}1}{}_{UB}(\tau_{i,j}{}^k, t)$, no more overlapping transmissions in Set 2 can be released, due to their late earliest start time.

A given problem with flow set F is schedulable under our scheduling policy, if for each $\tau_{i,j}{}^k$ the following condition is verified:

$$\left|TW(\tau_{i,j}^k)\right| - Delay_{UB}(\tau_{i,j}^k) > 0 \tag{13}$$

As an example, the worst case response time of the transmission $\tau_{4,1}{}^1$ in Fig. 2, in a network with two channels, occurs when the transmissions in Set 1= $\{\tau_{1,1}{}^0, \tau_{2,2}{}^0, \tau_{3,1}{}^1, \tau_{3,1}{}^0, \tau_{5,1}{}^0, \tau_{5,1}{}^1, \tau_{6,1}{}^2, \tau_{6,1}{}^1, \tau_{6,1}{}^0, \tau_{7,1}{}^2, \tau_{7,1}{}^1, \tau_{7,1}{}^0\}$ are available at the beginning of the slotframe, i.e., at t^0=0. However, among these transmissions only $\tau_{1,1}{}^0$, $\tau_{2,2}{}^0$, $\tau_{6,1}{}^1$, $\tau_{6,1}{}^0$, $\tau_{7,1}{}^1$, $\tau_{7,1}{}^0$ recieve higher priority, thereby obtaining $Delay^0{}_{UB}(\tau_{4,1}{}^1, 0) = 5$. In the sixth time slot, the transmissions in Set 2= $\{\tau_{1,2}{}^0, \tau_{2,1}{}^0, \tau_{3,1}{}^1, \tau_{3,1}{}^0, \tau_{5,1}{}^0, \tau_{5,1}{}^1\}$ could be available at time t^0=5, i.e., beginning of the sixth time slot. However, among these transmissions only $\tau_{1,2}{}^0$ and $\tau_{5,1}{}^0$, achieve higher priority, and thereby obtaining $Delay^0{}_{UB}(\tau_{4,1}{}^1, 0) = 7$. Thus, in the worst case, transmission $\tau_{4,1}{}^1$ is schedulable.

Notice that, other scheduling strategies for the packets in the network can be adapted to our upper bound delay analysis. Our evaluations in the next section show that the schedulability condition derived from our upper bound delay analysis introduces a tight upper bound compared with the schedulability condition in [22], obtaining an upper bound on response times that are significantly lower, and increasing the maximum schedulable utilization.

5. Experimental Results

In this section, we evaluate the performance of PC-LLF, mainly in terms of schedualbility ratio. It is the number of experiments in which an approach successfully schedules the defined problem compared to the total experiments. We evaluate the schedulability ratio of our algorithm by varying network scale, period and deadline of flows, and number routes defined for each flow. We compare the performance of PC-LLF with the heuristic algorithms proposed by [22] and [26], called Collision free-LLF (C-LLF) and Hop wise Scheduling Algorithm (H-SA) here, respectively. Moreover, we elaborate on the schedulability performance of our schedulability condition defined by Eq. 13 comparing it with the schedulability condition defined in [22]. UP denotes our schedulability condition, and UP* denotes the schedulability condition defined in [22].

The main difference between PC-LLF, H-SA and C-LLF is due to their prioritization approach. As described in Section 2, H-SA assigns higher priority to the transmissions with more hops away from their destinations and shorter deadlines. This means that, to prioritize $\tau_{i,j}^{k}$, H-SA cannot take in to account the collisions in the remaining path to its destination. Therefore, two packet transmissions with the same deadline and the same distance to the destination achieve the same priority, even if one packet must deal with highly congested path.

In order to prioritize the transmission $\tau_{i,j}^{k}$, the C-LLF algorithm takes in to account the remaining time before the deadline and the amount of collisions at the current local link. A transmission that is closer to its deadline and has high amount of local collisions receives higher priority. Hence, C-LLF could be less successful when handling delay guarantees in highly congested routes. For example, consider two transmissions with same deadline and with the same amount of collisions in their local link. In this case, by C-LLF they would achieve identical priority. However by PC-LLF, a transmission that must deal with higher amount of collisions in its route achieves higher priority. This fact helps to handle delay issues more efficiently.

The schedulability condition UP*, is defined based on a pessimistic worst case scenario for scheduling latency. This worst case scenario is defined when all the transmissions throughout the network whose their life time overlap with the life time of $\tau_{i,j}^{k}$ achieve a higher priority than $\tau_{i,j}^{k}$, and they are scheduled before $\tau_{i,j}^{k}$. However, this definition of the worst case scenario provides us a pessimistic condition, since there may be no time instance at which all overlapping transmissions would be released before scheduling $\tau_{i,j}^{k}$. Furthermore, all overlapping transmissions may not achieve a higher priority than $\tau_{i,j}^{k}$.

In our experiments, all networks are generated with random topology. For each node, we assign a number of neighbor nodes randomly selected in the range [1-7]. For the link between each pair of neighbor nodes, we assign a random packet reception rate in the range [50-100]%, based on the distance between them. As in [22], 80% of nodes are used as sources and destinations of the flows. For each flow, we randomly pick two nodes from the network as its source-destination pair. Then, we use Dijekstra algorithm [27] to determine the most reliable route connecting the source to its destination. The algorithms have been written in Java and the tests have been performed on a Windows OS machine with 2.4 GHz Intel Core 2 Duo processor.

As the first case, we conduct experiments to evaluate the impact of network scale and period of flows on the performance of PC-LLF, C-LLF and H-SA. The experiments are arranged using case studies with loose, intermediate and tight periods. In the experiments with loose periods, we evaluate the performance of algorithms on case studies with periods randomly selected among the range [2^{6}-2^{10}] time slots. In the intermediate and hard experiments, we exploit case studies with periods randomly selected among the range [2^{4}-2^{10}] and [2^{4}-2^{9}] time slots, respectively. The end-to-end deadline of a flow F_i equals to its period T_i. Among the three experiments, the amount of packets generated by each flow is the largest and the network must handle highest amount of packets in the experiments with tight periods, whereas in the experiment with loose periods the number of packets per flow is the smallest.

In our experiments, to avoid generating un-reasonable case studies, we ensure that the period assignment to flows is proportional to the distance between source and destination pairs. The flows with distant source/destination pairs would achieve larger periods such that they can probably meet the deadline.

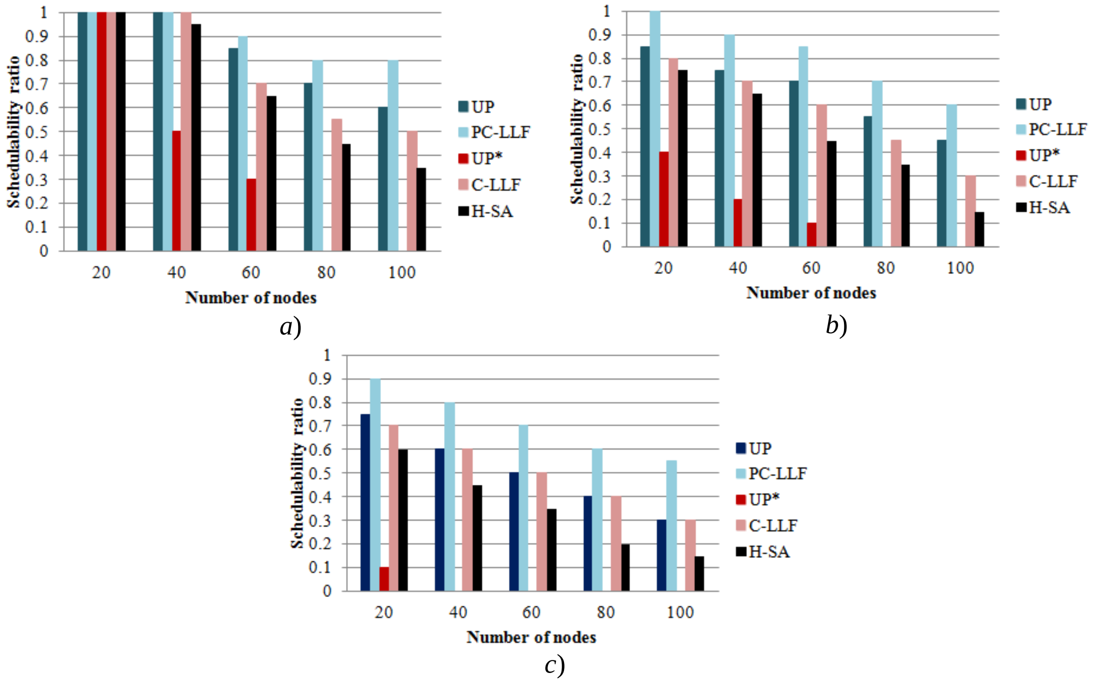
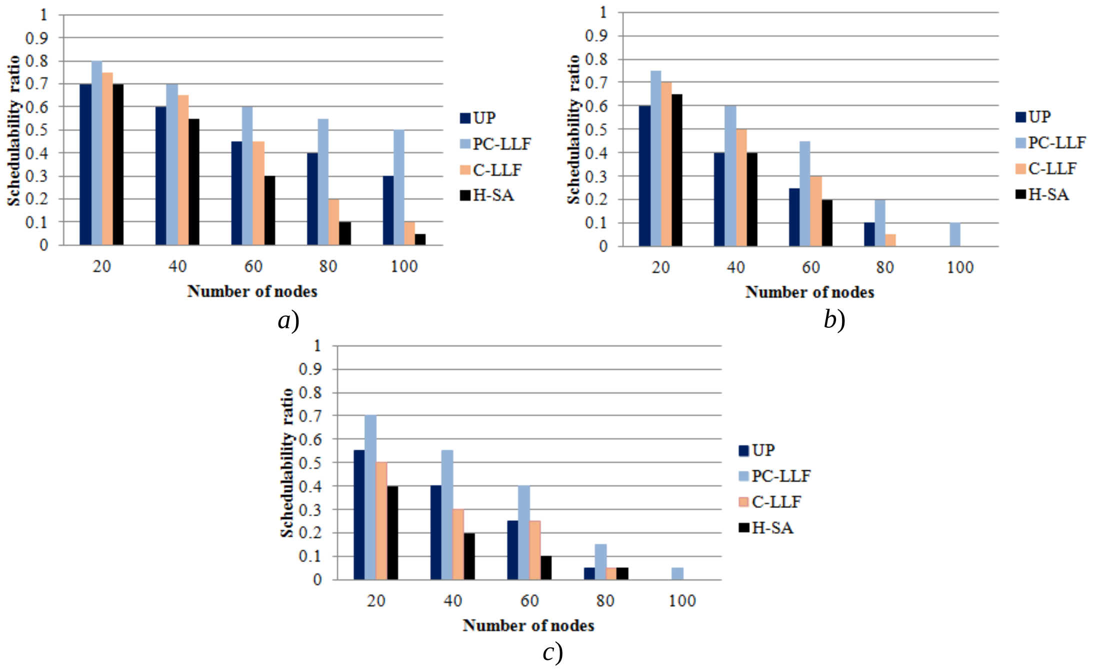


Fig. 4. Schedulability ratios by varying number of nodes in the network. Experiments with *a*) loose periods, *b*) intermediate periods and *c*) tight periods.

Fig. 4 shows the schedulability ratio per number of field devices in the network. The horizontal axis marks the number of nodes in the network, which is varied among the range [20, 40, 60, 80, 100]. That is, the number of nodes on the horizontal axis denotes the scale of the problem defined for experiments. We observe that, PC-LLF outperforms C-LLF and H-SA in all experiments, and the performance gap grows with increasing level of difficulty and the scale of networks. The main reason is that, at each scheduling step, the prioritization used by our algorithm properly identifies the important transmissions that must be forwarded through congested critical paths. Moreover, H-SA achieved lower performance in our experiments than that of C-LLF, mainly due to its poor capablity of distinguishing imortant transmissions that must traverse critical paths.

In addition, our experiments show that our schedulability condition highly improves the condition defined by the approach in [22]. Because, it excludes the contribution of transmissions that are impossible to delay $\tau_{i,j}^{k}$ schedule, due to their late release time or low priority. Therefore, our schedulability condition is able to eliminate much of the pessimism introduced by schedulability condition in [22], obtaining upper bound on response times that are significantly lower, and significantly increasing the maximum schedulable utilization.

Our second set of experiments evaluates the impact of network scale and deadline of flows on the performance of PC-LLF, C-LLF and H-SA. For each flow, we randomly assign a period in the range [2^4-2^{10}] time slots. We generate case studies with loose, intermediate and tight deadlines. In the experiments with loose deadlines, the deadline of each flow is 90% of its period. In the experiments with intermediate and tight deadlines, the deadline of each flow is 70% and 80% of its period, respectively. Deadlines are the most urgent in the experiments with tight deadlines.



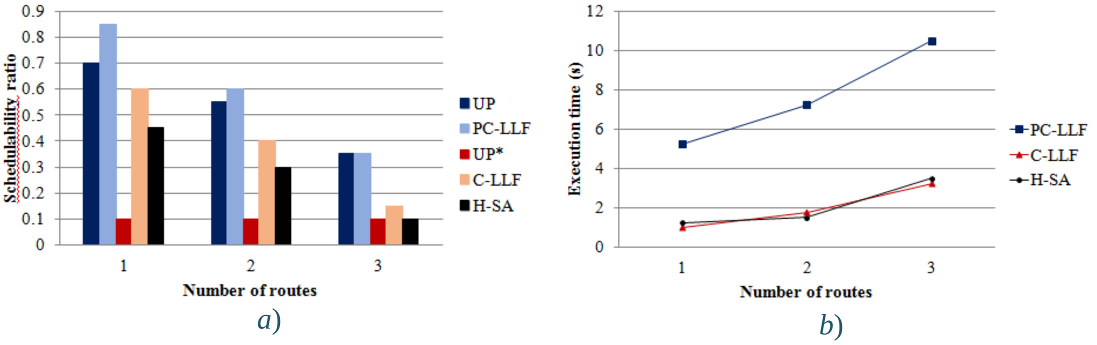
a)

b)

c)

Fig. 5. Schedulability ratio by varying deadlines and number of nodes in the network Experiments with *a*) loose deadlines, *b*) intermediate deadlines and *c*) tight deadlines.

Fig. 5 shows the schedulable ratio of each approach. It can be seen that, as the scale of networks increases or deadlines become tighter, the performance of all the approaches reduces. However, PC-LLF shows better performance. This is due to the fact that, when deadlines become tight, it is more important to consider delays imposed by the collisions that a packet could encounter through its path.



a)

b)

Fig. 6. Performance comparison under varying number of routes, *a*) Schedulability ratio, *b*) execution times.

In the next experiments, we evaluate the effect of multi-path routing on the performance of the algorithms. To this end, we use the randomly generated case studies in our first experiments, with 60 nodes and intermediate periods. The number of routes are varied among the range [1, 2, 3]. We use Dijekstra algorithm to find the most reliable routes. From **Fig. 6** it can be seen that, when the number of routes increases, it becomes more difficult for all the approaches to find feasible solutions. Growing number of routes highly increases the amount of packet workloads in the network. However, PC-LLF outperforms others in schedulability ratio, and it consumes the longest time among all evaluated heuristic algorithms.

4. Conclusion

This paper presents a scheduling heuristic approach for time-triggered TSCH networks and a schedulbiltity condition based on existing schedulbility analysis for multi-processors and distributed systems. First, a message prioritization approach is proposed, which dynamically prioritizes each message considering its end-to-end deadline and collisions it might deal with throughout its path to the destination. Then by our proposed scheduling approach, the messages are scheduled over time-triggered TSCH network. The simulation results have shown that the proposed heuristic approach and our schedulability condition outperform existing approaches for various settings, in increasing schedulable ratio.

References

[1] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Trans. Ind. Informat.*, vol. 4, no. 2, pp. 102–124, May 2008. Article (CrossRef Link)

[2] D. Miorandi, E. Uhlemann, S. Vitturi, and A. Willig, "Guest editorial: Special section on wireless technologies in factory and industrial automation, Part I," *IEEE Trans. Ind. Informat.*, vol. 3, no. 2, pp. 95–98, May 2007. Article (CrossRef Link)

[3] F. Salvadori, C. S. Gehrke, A. C. de Oliveira, M. de Campos, and P. S. Sausen, "Smart Grid Infrastructure Using a Hybrid Network Architecture," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1630-1639, Sep. 2013. Article (CrossRef Link)

[4] H. Ghayvat, J. Liu, S. C. Mukhopadhyay, and X. Gui, "Wellness sensor networks: a proposal and implementation for smart home for assisted living sign in or purchase," *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7341-7348, Dec. 2015. Article (CrossRef Link)

[5] "Gas Detection Handbook: Key Concepts and Reference Materials for Permanently Installed Gas monitoring Systems," *MSA The Safety Company*, Aug. 2007.

[6] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," in *Proc. of IEEE*, vol. 91, no. 8, pp. 1235–1246, Aug. 2003. Article (CrossRef Link)

[7] Embedded WiSeNTs Consortium, "Embedded WiSeNts research roadmap" [Online]. Available: http://www.embedded-wisents.org.

[8] M. P. Kazmierkowski "Networked embedded systems (Zurawski, R.; 2009) [Book News]," *IEEE Industrial Electronics Magazine*, vol. 3, no. 3, pp.57-57, 2009. Article (CrossRef Link)

[9] E. Borgia, "The Internet of Things vision: key features, applications and open issues," *Computer Communications*, vol. 54, no. 1, pp. 1-31, Dec. 2014. Article (CrossRef Link)

[10] X. Jin, F. Kong, L. Kong, W. Liu, and P. Zeng, "Reliability and temporality optimization for multiple coexisting WirelessHART networks in industrial environments," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6591-6602, 2017. Article (CrossRef Link)

[11] "IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," pp. 1–314, 2011. Article (CrossRef Link)

[12] G. Anastasi, M. Conti, and M. Di Francesco, "A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 7, no.1, pp. 52–65, 2011. Article (CrossRef Link)

[13] K. Yedavalli and B. Krishnamachari, "Enhancement of the IEEE 802.15.4 MAC protocol for scalable data collection in dense sensor networks," in *Proc. of Int. Symp. Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks* (*WiOpt08*)*, Berlin, Germany*, Apr. 2008. Article (CrossRef Link)

[14] S. Pollin, M. Ergen, S. Ergen, B. Bougard, L. Van der Perre, I. Moerman, A. Bahai, and F. Catthoor, "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer," *IEEE Trans. Wireless Commun*., vol. 7, no. 9, pp. 3359–3371, Sep. 2008. Article (CrossRef Link)

[15] L. a. N. Man, S. Committee, and I. Computer, IEEE Std 802. 15.4e-2012 (Amendment to IEEE Std 802.15.4-2011) IEEE Standard for Local and metropolitan area networksPart 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer, 2012th ed., IEEE, Ed. New York, USA: Institute of Electrical and Electronics Engineers, vol. 2012, April 2012. Article (CrossRef Link)

[16] A. Tinka, T. Watteyne, and Kris Pister, "A decentralized scheduling algorithm for time synchronized channel hopping," in *Proc. of Ad Hoc Networks, International Conference on Ad Hoc Networks*, pp 201-216, 2010. Article (CrossRef Link)

[17] A. Morell, X. Vilajosana, J.L. Vicario, and T. Watteyne, "Label switching over IEEE802.15.4e networks," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 5, pp. 458-475, 2013. Article (CrossRef Link)

[18] R. Soua, P. Minet, and E. Livolant, " Wave: a distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks," *Transactions on Emerging Telecommunications Technologies* 2015.

[19] M.R. Palattella, N. Accettura, M. Dohler, L.A. Grieco, and G. Boggia, "Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks," in *Proc. of IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications* (*PIMRC*), Sept. 2012. Article (CrossRef Link)

[20] 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. Article (CrossRef Link)

[21] R. Soua, P. Minet, and E. Livolant, "MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks," in *Proc. of 2012 IEEE 31st International Performance Computing and Communications Conference* (*IPCCC*), 1-3 Dec. 2012. Article (CrossRef Link)

[22] A. Saifullah,Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in *Proc. of 31st IEEE Real-Time Systems Symposium* (*RTSS*), Dec. 2010. Article (CrossRef Link)

[23] H. Zhang, P. Soldati, and M. Johansson, "Performance bounds and latency-optimal scheduling for convergecast in WirelessHART networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2688-2696, 2013. Article (CrossRef Link)

[24] HART Communication Protocol and Foundation. Austin, TX, USA, 2012. [Online]. Available: http://www.hartcomm2.org

[25] Wireless Systems for Industrial Automation: Process Control and Related Applications, Standard ISA-100.11a-2009, 2009.

[26] G. Chen, X. Cao, L. Liu, C. Sun and Y. Cheng, "Joint Scheduling and Channel Allocation for End-to-End Delay Minimization in Industrial WirelessHART Networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2829-2842, 2019. Article (CrossRef Link)

[27] T. H. Cormen, "Introduction to algorithms," *MIT press*, 2009.

[28] A. Saifullah, Y. Xu, C. Lu, Y. Chen, "End-to-end delay analysis for fixed priority scheduling in WirelessHART networks," in *Proc. of 17th IEEE Real-Time and Embedded Technology and Applications Symposium, USA*, May, 2011.

[29] M. Bertogna, and M. Cirinei, "Response-time analysis for globally scheduled symmetric multiprocessor platforms," in *Proc. of 28th IEEE International Real-Time Systems Symposium (RTSS), USA*, Dec. 2007. Article (CrossRef Link)

[30] J.C. Palencia, and M. González Harbour, "Schedulability analysis for tasks with static and dynamic offsets," in *Proc. of 19th IEEE Real-Time Systems Symposium* (*RTSS*), Dec. 1998. Article (CrossRef Link)

Armaghan Darbandi received the B.S. degree in computer engineering from Polytechnic University, Tehran, Iran, in 2003, and the M.S. and Ph.D. degrees in computer engineering from the University of Ulsan, Ulsan, South Korea, in 2017. Since September 2017, she has been a Research Scientist with the Wireless Sensor Networks Laboratory, University of Ulsan. Her research interests include wireless communications and networking, and real-time embedded systems.

Myung Kyun Kim, Ph.D., is currently a professor in School of IT Convergence, University of Ulsan, Korea. His main research area is real-time communications in wireless networks and industrial communication networks. He received the B.Sc. degree from Seoul National University of Korea and the M.Sc. and Ph.D. degrees from KAIST, Korea.