CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

RPL over DSME: A Technical Report

Technical Report

Harrison Kurunathan
Ricardo Severino
Anis Koubâa
Eduardo Tovar

CISTER-TR-180110

2018

RPL over DSME: A Technical Report

Harrison Kurunathan, Ricardo Severino, Anis Koubâa, Eduardo Tovar

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: hhkur@isep.ipp.pt, rar@isep.ipp.pt, aska@isep.ipp.pt, emt@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

IEEE 802.15.4e and RPL are the standard communication protocols for IoT medium access control (MAC) and routing protocols, respectively. These two standard were designed independently but with common objectives to satisfy the requirement of IoT devices in terms of limited energy and computation and storage resources. However, there has been little work in the literature that combined these two protocols together to leverage their integration in building scalable and large-scale IoT networks. This paper tackles this problem and contribute with the integration of RPL over IEEE 802.15.4e DSME MAC protocol to build scalable IoT networks with real-time QoS requirements. Our approach leverages the link asymmetry of the IEEE 802.15.4e when using different transmission powers to build a hierarchical network using RPL routing and taking benefits from DSME multi-channel time-slot allocation mechanism for admission control and QoS guarantees. We propose a new multi-channel multi-time slot scheduling algorithm called Symphony for DSME, by which we aim at integrating RPL over DSME and providing a QoS efficient schedule for GTS placement. We also provide a performance evaluation of the delay of our system using probabilistic analysis

RPL over DSME: A Link-Asymmetric QoS-Qware Communication Protocol Stack for Scalable IoT Networks

John Harisson[§] Anis Koubaa[§*], Ricardo Severino [§], Eduardo Tovar[§],
[§]CISTER/INESC TEC and ISEP-IPP, Porto, Portugal
[*] Prince Sultan University, Saudi Arabia.
Email: (mhgha, aska, emt)@isep.ipp.pt, akoubaa@psu.edu.sa

Abstract*—IEEE 802.15.4e and RPL are the standard communication protocols for IoT medium access control (MAC) and routing protocols, respectively. These two standard were designed independently but with common objectives to satisfy the requirement of IoT devices in terms of limited energy and computation and storage resources. However, there has been little work in the literature that combined these two protocols together to leverage their integration in building scalable and large-scale IoT networks. This paper tackles this problem and contribute with the integration of RPL over IEEE 802.15.4e DSME MAC protocol to build scalable IoT networks with real-time QoS requirements. Our approach leverages the link asymmetry of the IEEE 802.15.4e when using different transmission powers to build a hierarchical network using RPL routing and taking benefits from DSME multi-channel time-slot allocation mechanism for admission control and QoS guarantees. We propose a new multi-channel multi-time slot scheduling algorithm called Symphony for DSME, by which we aim at integrating RPL over DSME and providing a QoS efficient schedule for GTS placement. We also provide a performance evaluation of the delay of our system using probabilistic analysis.

I. OVERVIEW

A. IEEE 802.15.4e - DSME

IEEE 802.15.4e is a standard specifically developed to meet the quality requirements of industrial sensor network systems. Among the several MAC behaviors introduced in IEEE 802.15.4e to support low rate and robust real time communication, Deterministic Synchronous Multi-channel Extension (DSME) stands out because of its exclusive features. Some of its unique features of DSME include ways to increase the overall scalability and at the same time providing deterministic communication.

All nodes in a DSME network are synchronized by a collection of superframes called the multi superframe structure shown in figure 1. The rows that span across the superframe indicate the channels and the columns represent the timeslots. Every superframe within a multi superframe consists of Contention Access Period (CAP) and Contention Free Period (CFP) similar to that of its parent standard : legacy IEEE 802.15.4. Within CAP nodes contends to occupy the timeslots using standard CSMA/CA. CFP is made of guaranteed timeslots which are allocated to the nodes by the PAN coordinator. For Guaranteed communication every GTS slot in the CFP region of the superframe accommodates the transmission of data and its eventual acknowledgment.

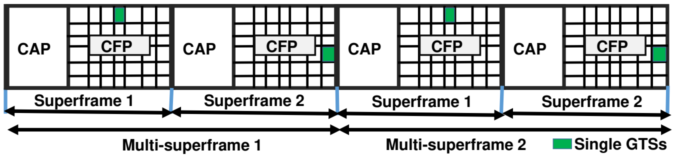


Fig. 1: IEEE 802.15.4e multi superframe structure

Under multichannel access, DSME specifies two mechanisms namely: *channel adaptation* and *channel hopping*. Under adaptation, DSME PAN coordinator has the capability to assign nodes to a DSME GTSs in a single channel or different channels based on link quality metrics. In channel hopping, the transmissions that are supposed to accommodate the guaranteed timeslots is pre-determined. The same hopping pattern will be repeated till the end of the data transmission. Whereas, channel adaptation provides a different approach in which the transmissions are allowed to hop over that channels based on their link quality.

The multichannel access mechanisms of DSME allow several transmission to occur in the same timeslot within different channels. This is called *link asymmetry*. Using this feature will help reducing the latency and delay of the network efficiently. This paper provides a scheduling algorithm that will help accommodating link asymmetry optimally in DSME enabled networks. The channel adaptation techniques will be supported by the RPL protocol which will be discussed later in this section.

B. RPL

IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) is a protocol designed for Low power and Lossy Networks (LLN). It integrates technologies such as IEEE 802.15.4 and IPv6 protocols. In order to ensure an effective routing in 6LoWPAN network, IETF ROLL working group proposed a routing protocol named RPL. RPL supports both mesh as well as hierarchical topologies. RPL is specifically designed to support networks that are prone to highly exposed packet losses and limited resources in terms of computation and energy. RPL is a distance vector (DV) and a source routing protocol which operates on the IEEE 802.15.4 PHY and MAC layers. It supports point-to-point and point to- multi-point traffic.

RPL is based on hierarchical Directed Acylic graphs (DAGs). Contrast to a classical tree, in a DAG a node can associate itself with many parent nodes. The destination nodes of an RPL is called a sink and the nodes through which a route is provided to internet are called gateways.RPL organizes these nodes as Destination-Oriented DAGs (DODAGs). Several DODAGs can be present in a network. Every node in a DODAG has a rank, which is the individual position of a node with respect to its neighbors in the system. A basic example of an RPL network is shown in Figure The rank increases outwards from the DODAG root as shown in Figure 2.

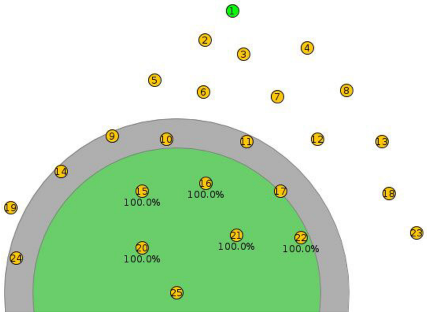


Fig. 2: Rank increasing towards the DODAG root

In order to construct a network topology, every router in the system identifies and associates with a parent in a specific DODAG root. This is done based on an objective function. Objective function helps in computing the rank of a node(s) and providing them an optimal routing path using metrics such as latency and power efficiency.

II. RELATED WORKS

Due to increased pervasiveness *Wireless Sensor Networks* (WSN) has become an emerging field to support applications based on *Internet of Things* (IoT) and *Cyber Physical Systems* (CPS). These networks demand various quality of service for specific applications. Reduced latency and determinism has become one of the most demanded attributes for a sensor network. Standards like IEEE 802.15.4e and LoRA [1] with multi-channel functionality and adaptive data rates have been issued to meet these needs.

In the literature, there have been several distributed scheduling algorithms like *Depth Based TDMA*[2] that supports intra cluster tree topology based IEEE 802.15.4 networks. Using this protocol local scheduling was provided for a multi-hop network using service. Local scheduling aims at reducing power consumption by scheduling transmission and reception time slots in each node's power schedule and sleeps during idle period. However, the schedules dynamically change with the network demand changes. Though IEEE 802.15.4 was one of the prominent protocols, it lacked features like multichannel access which is a key aspect of implementation of an IoT.

The DSME MAC behavior provides more determinism in multi channel environment. Several researchers like in [3] and [4] have showcased the advantages of DSME in terms of better frame error rates and aggregate throughputs.

Several researchers have worked on developing scheduling algorithms that use the functionalities of the standard to produce an optimal service. For example in case of TSCH, an other prominent MAC behaviors of IEEE 802.15.4e, a new enhancement called *Adaptive-TSCH* [5] developed by Peng Du. In this algorithm, the author provides the nodes the ability to hop amongst a subset of channels which are deemed reliable based on their respective link qualities. It was also inferred that this technique guards the transmissions from any outside interference. An average increase of ETX (Expected Transmission Count) by 5.6 % was observed under this algorithm. Similar enhancements like new transmission schemes and beacon broadcast schemes have been introduced for DSME in [6] which helps the network to achieve better performance results in terms of throughput.

Several earlier works like S-MAC [7] and T-MAC [8] have focused modifications in existing protocols to increase the QoS. In S-MAC, virtual clusters were made out of the neighboring nodes and the sleep schedules were periodically maintained, whereas, T-MAC provided fixed sleep and adaptive wakeup schedules which resulted in better service. But when came to very stringent deadlines and a need to accommodate a denser network, cross-layer protocols were more effective. One of the very commonly amalgamated protocol with existing standards in RPL, because, it helps it providing optimal routing solutions to the transmissions, thus increasing the overall service.

Following the standardizing efforts of cross layer protocols like 6LoWPAN [9] and ROLL[10], the *Internet Engineering Task Force*(IETF) focuses on implementing 6TiScH [11] a combination of the TSCH MAC behavior of IEEE 802.15.4e, IPV6 and RPL. *Orchestra* [12] is one of the open source implementations based on 6TiScH, in which, the nodes automatically compute their own local schedules and maintain several schedules for different traffic scenarios. Orchestra, in order to maintain schedules, relies on the existing network stack of TSCH and not any distributed or central scheduler. Orchestra was implemented on TelosB nodes and ContikiOS. It was inferred that this cross layer protocol was more efficient than the legacy TSCH because of its unique scheduling process. Orchestra was able to deliver 99.9% of end-end delivery ratios at the same time it maintains a good margin of latency-energy balance.

In this paper, we propose a cross layer enabled scheduling algorithm. We aim at improving the Quality of Service of DSME by integrating it with RPL. RPL will be used in providing an optimal schedule for the multi-channel timeslot allocation and our proposed algorithm *Symphony* will help in efficient placement of the GTS schedule.

III. SYSTEM ARCHITECTURE

A. Symphony in a nutshell

The DSME network will consist of a central PAN coordinator and several coordinators with routing capabilities. Apart from these nodes which are Full Function Devices (FFD), there can also be provisions for Reduced Function Devices (RFD) which are the slave/end nodes with no routing capabilities. The RFD will always connected to their associated coordinator within its range. We propose a strict star topology between the coordinators and RFD.

This paper aims in introducing a new approach to scheduling the GTS of DSME network with RPL functionalities. Symphony is a dynamic algorithm that changes in accordance with the change in link metrics provided by the MAC layer which eventually effects the optimal routing decisions. The approach developed in this paper is general and it can be applied for scheduling in MAC layers with RPL functionality over multichannel access.

The coordinators will maintain schedules locally and will have their own superframes to accommodate the nodes associated to them. The coordinators will maintain a routing table towards its routing children with a lesser rank in their respective DODAG. RPL supports both broadcast and unicast for disseminating the performance metrics using *DIO*, request DODAG information using *DIS* and disseminate the routing path using a *DAO*. Symphony will allow DSME to be flexible enough to form a complete mesh network with the nodes that have routing capabilities.

The symphony schedule will comprise for different superframes of varying lengths. Every superframe will comprise of different traffics: The periodic beacons for synchronization, RPL signaling traffic and application data traffic. The nodes will select the timeslots based on the scheduling rules, which makes symphony more appealing as it will both utilize the high scalability features of DSME at the same time, maintain a network with a peculiar RPL based objective function like power efficiency.

A concrete example of symphony (Figure 3) is as follows:

- A dedicated beacon broadcast for synchronization between every superframe for every ”X” slots, where ”X” is the superframe duration of every individual superframe.
- A dedicated beacon broadcast for synchronization every multi superframe for every ”Y” slots, where ”Y” is greater than ”X” and is the multi superframe duration coordinating every superframe with the duration of ”X”.
- A Enhanced beacon common for all coordinators in the network carrying the broadcast+unicast for RPL signaling (DIO, DIS, DAO), repeating every ”Y” slots
- Dedicated unicast signal from the slave node to the parent node for a time interval ”Z”.
- N unicast signals from the coordinator to the slave nodes for ”Z”.
- Dedicated unicast signal from the coordinator to the RPL preferred parent for a time interval of ”H”, where H is lesser than ”Z”.
- N unicast signals from the coordinator/PAN coordinator to the associated coordinators for every ”H”.

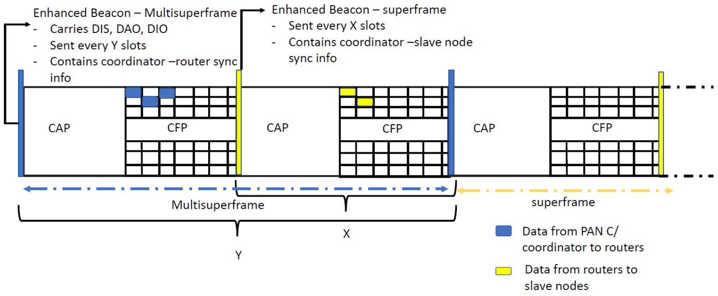


Fig. 3: Symphony: an example

Symphony ensures in maintaining a proper schedule in which the transmissions can happen in parallel without a overlap. We select the channels and the GTS timeslots for the scheduler based on the nodes unique MAC identifier. Symphony aims in providing a contention free scheduling based on ranking and priority based scheduling rules.

B. System model and assumptions

We consider an IEEE 802.15.4e network composed of a set of sensor nodes and a master node (the PAN coordinator). The IEEE 802.15.4e network deployed the DSME behavior on all the nodes. We consider a *data collection network* were all the sensor nodes have to report their readings to the PAN coordination.

We assume that the PAN coordinator sets-up a network with a multi-superframe structure where `MO` defines the Multi-frame Order, the `BO` defines the Beacon Order, and `SO` defines the superframe order. The multi-frame duration, beacon interval and superframe duration are determined using Equations (1), (2) and (3) respectively.

$$MD = aBaseSuperframeDuration \ \times 2^{MO} symbols \tag{1}$$

$$BI = aBaseSuperframeDuration \ \times 2^{BO} symbols \tag{2}$$

$$for \ 0 \leq BO \leq 14$$

$$SD = aBaseSuperframeDuration \ \times 2^{SO} symbols \tag{3}$$

$$for \ 0 \leq SO \leq BO \leq 14$$

In contrast to traditional DSME networks, which are mainly organized in a star topology, we consider a multi-hop network organized in a mesh structure, where the mesh is built using the RPL protocol. The question now is *how to build a multi-hop mesh network using RPL, with an IEEE 802.4.15e/DSME MAC behavior underneath*? This obviously does not make sense if all the nodes have a bi-directional communication with the PAN coordinator. In fact, in this case all the nodes will be able to reach the PAN coordinator, and can also be reached by the PAN coordinator. In such network, the scalability is naturally limited as all the nodes are constrained to be in each other communication range with the PAN coordinator. In addition, if all nodes use their maximum transmission power to extend their coverage, it will result in quickly draining their energy resources and thus dramatically shorten the network lifetime.

link asymmetry: : To improve the scalability of the network, multi-hop topology needs to be adopted. Our idea consists to exploiting the link asymmetry in an IEEE 802.15.4e/DSME network by adopting different transmission ranges in the PAN coordinators and other nodes, which considers their role in the network, and their energy resources. In fact, we assume that the PAN coordinator is able to transmit at the maximum transmission power, whereas as other nodes can transmit at lower transmission powers, which will creates a link asymmetry between the PAN coordinator and the remaining sensor nodes in the network.

First, the link asymmetry resulting from different transmission power is nowadays feasible with current COTS technologies. In fact, the commercially available XBee IEEE 802.15.4-compliant modules operate at different maximum transmission powers allowing them to have different communication ranges. Table 1 presents an overview of the different XBee modules and their characteristics. It is observed that XBee module can allow a communication range for up to 1500 meters with a transmission power 60 mW (18 dBm) of transmission range, and a communication range up to 100 meters with a transmission power 1 mW (0dBm) [1].

On the other hand, this link asymmetry property is very appropriate with the energy resources of the PAN coordinator and the nodes. In fact, the PAN coordinator in IoT application is general mains-powered as it is the connection point between the low-power sensor networks and the Internet. Thus, the transmission at the highest power, and consequently large communication range, will not affect the energy of the PAN coordinator. However, the remaining sensor nodes are typically battery powered and do not have the luxury to transmit at the highest transmission powers to avoid depleting their batteries. For that purpose, it is more convenient to use lower transmission powers for these nodes to extend their lifetime. In this case, they will not be able to directly reach the PAN coordinator only through multiple-hop routing, and this where RPL comes into play.

C. Scheduling Problem Statement

1) Network model:

a) Synchronization: We consider a network composed of a PAN coordinator routers and slave nodes. The PAN coordinator is able to send at two transmission powers depending of the type of packet to send:

- **Control packets:** The PAN coordinator sends control packets like beacon frames to all nodes in the network in a single hop using its maximum transmission power.
- **Data packets:** The PAN coordinator sends other data packets using its minimum transmission power using RPL multi-hop routing.

The following strategy guarantees that all nodes in the network are synchronized through the reception of the beacon at the same time from the PAN coordinator in a single hop transmission. We assume that the propagation time is negligible.

b) Node Association : Once a node receives the beacon frame from the PAN coordinator, it can join the network by associating to another router node in the network. The PAN coordinator when it forms a network, sends Enhanced Beacons as a broadcast to all the nodes in its range, the coordinators with routing capabilities and slave nodes at the vicinity of the range of the PAN-C can associate with the sink. The node requesting to join a network, must send the PAN-ID in his request, and this PAN-ID can only be obtained from the beacon frame sent by the PAN Coordinator and that the node should have received. The coordinators with routing capabilities, on the other hand also start to send enhanced beacons within their own range to get associated with nodes, thus eventually leading to a formation of a network topology. Following the node association, the router adds it as a child and send a confirmation to the node.

At the network level, the association process follows the RPL routing node joining process. The PAN Coordinator will act as a DODAG root and sends DODAG messages using its minimum transmission power. All the routers in the RPL overlay network keep sending their DIO messages to announce the DODAG.

A node will listen to DIO message only if joins the PAN by receiving a beacon frame from the PAN Coordinator. If the node does not receive a beacon frame, it will ignore all DIO messages. When a node wants to join the DODAG it receives a DIO message from a neighbor router, it (*i.*) adds the DIO sender address to its parent list, (*ii.*) computes its rank according to the Objective Function specified in the OCP (Objective Code Point) field, which is an identifier that specifies what Objective Function the DODAG uses. This ensures that the nodes rank is greater than that of each of its candidates, (*iii.*) forward the DIO message with the updated rank information. The client node chooses the most preferred parent among the list of its parents as the default node through which inward traffic is forwarded.

We used the contiki to form a mesh network based on an objective function. The objective functions can be several variable QoS (Quality of Service) metrics such as power consumption and latency. In Figure 4 , we show a mesh network with a single PAN coordinator and 7 nodes based on objective functions.

When a node joins the network, it can either send its packets through the RPL network during the CAP period of the superframe, or request to allocate a time slot for critical packets. The routing a critical packets also occur on the RPL tree, but will be performed during dedicated time slots, and not contention-based at MAC level.

c) Time-Slot Request/Response: If a node needs to allocation a time slot in the CFP of a multi-frame, it must first send its request to the PAN coordinator through the RPL network,

---

[1] We can also investigate the LORA protocol, which allows very large communication ranges.

TABLE I: XBee standards and characteristics

XBee Standards	Frequency	Data rate	Communication protocol	Maximum transmit power	Channels utilized
XBee ZigBee (S2C)	2.4 GHz	RF: 250kbit/s, Serial: 1Mbit/s	ZigBee	XBee ZigBee: 6.3 mW (8dBm)	XBee ZigBee: 16
XBeePRO 900HP	900 MHz	10kbit/s or 200kbit/s	Proprietary	250 mW (24dBm)	FHSS
XBee 802.15.4	2.4 GHz	250kbit/s	802.15.4	1 mW (0dBM)	XBee 802.15.4: 16
XBee DigiMesh 2.4	2.4 GHz	250kbit/s	Proprietary	XBee DigiMesh 2.4: 1 mW (0dBm)	XBee DigiMesh 2.4: 16
XBee 868LP	868 MHz	10kbit/s or 80kbit/s	Proprietary	25 mW (14dBm)	30
XBee-PRO XSC	900 MHz	10kbit/s or 20kbit/s	Proprietary	250 mW (24dBm)	FHSS
XTend 900 MHz	900 MHz	10kbit/s or 125kbit/s	Proprietary	1000 mW (30dBm)	FHSS (50 channels)
XBee Wi-Fi	2.4 GHz	72Mbit/s	802.11b/g/n	16dBm	13

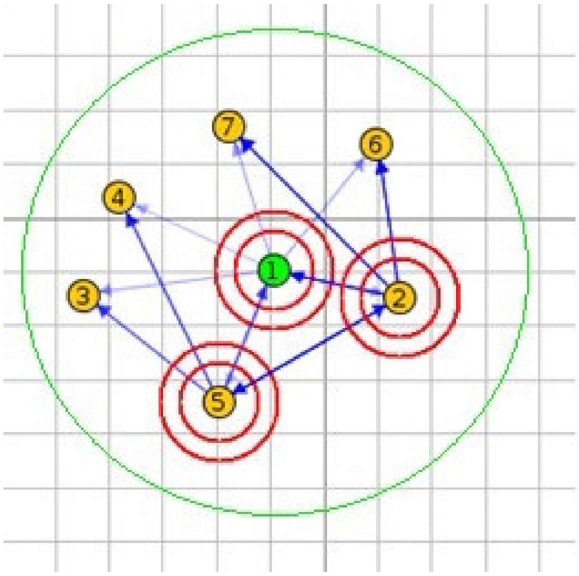


Fig. 4: Mesh network using Contiki

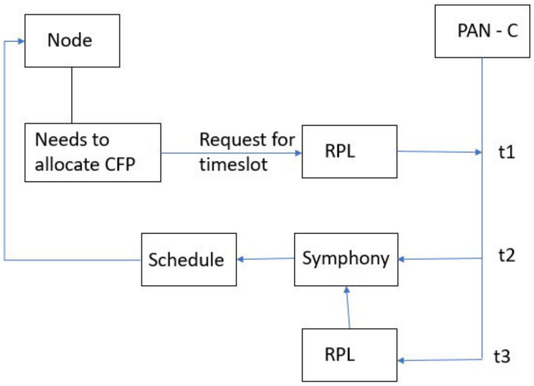


Fig. 5: Timeslot request - response process

by contending for the medium at each hop. The request is forwarded to the PAN Coordinator by intermediate routers using RPL. Once the request reaches the PAN Coordinator, the latter will run a scheduling algorithm to find the most efficient allocation in the time-frequency domain that satisfies multiple objectives. We present an UML sequence (Figure 5) diagram in which we illustrate this timeslot request process. In the figure $t1$ represents the request, $t2$ represents the response, and $t3$ provides the the change in the scheduling algorithm based on RPL.

Once the allocation is done, the PAN Coordinator will send a response back to the requesting node in the next beacon frame using the maximum transmission power, so in a single hop transmission, leveraging the link asymmetry property discussed in the previous section. Thanks to the time synchronization between the PAN Coordinator and all nodes in the network using the one-hop beacon transmission, the requesting node will be informed on the slot to use in the time frequency domain, and will use that time slot with the specified frequency to transmit to the PAN Coordinator through the intermediate routers. The scheduling algorithm should determine a QoS-guaranteed path through all routers by allocating a slot in the time frequency domain from the requesting node until the PAN Coordinator.

IV. SYMPHONY

The *Symphony* is a routing aware algorithm based on *Constraint Satisfaction Problem* (CSP). Symphony does a schedule placement very similar to the classic *eight queens problem* [13]. The solution for the eight queens problem follows the strategy of assigning a slot and checking if the constraint is satisfied. Symphony follows the same strategy but with more slot deciding parameters involved. The problem of optimal assignment of time-slots and frequencies is known to be an NP-Hard problem [14]. Symphony will aim at providing dynamic allocation of timeslots based on the routing information provided by RPL.

A. Symphony Design

Routing aware Algorithm: Symphony will usually prefer the RPL preferred parent for the node. With the time, link quality changes between the nodes. With the link quality deterioration, the RPL will evolve to select new pairing routes for the nodes. This will be updated using an enhanced beacon in the next period of the superframe. The joining priority will be fixed with the maximum prioritization passing down the ranks (i.e PAN coordinator $>$ Coordinator/router $>$ slave nodes). Using this we can achieve a higher reliability within the higher rank of nodes, eventually avoiding any possibility of dire failures. Over the entire network lifetime, Symphony will update the scheduling information using the information provided by the RPL. The RPL helps in building the topology from the initialization of the network. The nodes in the network will be given unique IDs during the initialization process. RPL will update any change to the routing table when the routing path varies from one node to another.

DSME time synchronization: The DSME time synchronization shall be carried out using periodic beacons (for individual superframes), Enhanced beacon (one per every multi superframe) and RPL beacon frames. There will be individual synchronization carried out for the nodes in the vicinity of the RPL parent node it is associated with. Beacons to multiple slave nodes from the coordinator shall always be a broadcast. For efficient synchronization purpose Group acknowledgment will be neglected from the symphony architecture of DSME. Individual unicast acknowledgments shall be passed from the nodes to their parents, in case of an acknowledgment miss, the node will wait for a keep alive period and restart the association process with the node using a unicast beacon.

B. Scheduling constraints

The problem is bounded by the following two constraints, which will be a determining factor in establishing an optimal solution. Using the constraints, we aim at providing a schedule for the DSME frame format. The input before scheduling the timeslots will be provided utilizing RPL, which will be a set of transmissions that helps forming the mesh network based on a certain objective function like latency.

Constraint 1: *No same nodes either involving in transmission or reception must fall under the same timeslot.* This constraint helps in avoiding all the interference in the network. We give a possibility for *different* nodes to communicate in a *same* timelot simultaneously in *different* channels, whereas, the same nodes can communicate in *different* timeslot within the same or different channels.

Constraint 2 *Maximum number of channels and minimum number of timeslots should be used*. This constraint is more of a quality constraint that helps in verifying the optimality of the algorithm. This constraint helps in achieving the fact that the bandwidth is utilized to the maximum and the same time minimal timeslots are used, so that the overall network throughput and scalability of the network is significantly increased and minimal latency is also maintained by the schedule.

C. Symphony Architecture

The cross layer architecture can be structured as shown in Fig 6. The association and the building of nodes has to be carried out in a DSME level. At the network level RPL forms efficient routing paths for the data to be transferred from one end to the other.The routes formed at the network level shall be disseminated using the IPv6 low-power backbone router. Now the routes proposed by the RPL shall be placed in the DSME GTS in accordance with the symphony algorithm.

An application layer can also allocate a selective priority GTS. The Higher layer then sends *MLME DSME GTS REQUEST* (management layer request), providing details of the preferred GTS and a bitmap which marks all the available GTS. Following this process the allocation commands are received in the management layer. Usually the higher layer tries to allocate the preferred GTS slot first, if successful a GTS notification shall be passed to the neighbors in the nodes vicinity.

Apart from the network layers, the Datagram Transparent Layer security (DTLS) helps in providing automatic key management for data encryption and integrity. DTLS is a a very secure protocol that requires numerous message interaction before a session is established. Protocols for emerging Authentication of Network Access (PANA) can also be used in this scenario. An MCP (Internet Message Control Protocol) will run in parallel for sending error messages and operational information.

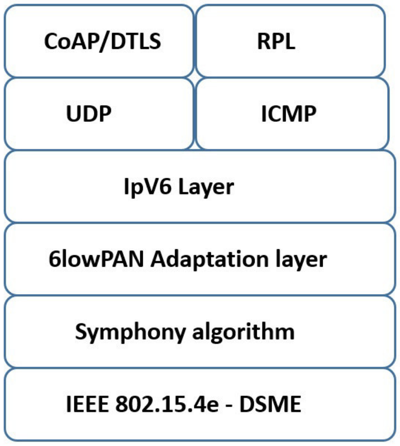


Fig. 6: Symphony Architecture

In the following section, we will be providing an ILP model based upon the precedence relations of every transmission in the mesh network obtained out of RPL. From this model, we obtain a placement schedule for DSME timeslots.

V. ILP MODEL

Problem Statement:
Placing the schedule in the the available guaranteed timeslots in an optimal way such that minimum number of timeslots are used and maximum bandwidth is utilized

assumptions:
For the Integer liner programming (ILP) model, We us take an example of a mesh network with 5 different nodes which are interconnected with each other as shown in Fig 7. As a realistic

assumption, we consider that all the nodes in the network are able to transmit as well as receive information. The network model can also be extended to slave nodes with minimal functionality. We only consider the guaranteed timeslots in the CFP region of the DSME superframe in our model. For an optimal schedule, the transmissions are expected to be packed in the GTSs in such a way lesser timeslots are used, utilizing the advantage of multichannel access.

Step 1: dependency graph formation: For formulating our ILP model, we first define all the transmissions as flows for easy readability. The flows are defined as:

$c \rightarrow d(a1), c \rightarrow a(a2), a \rightarrow b(a3), b \rightarrow e(a4), b \rightarrow d(a5), d \rightarrow f(a6), f \rightarrow a(a7), e \rightarrow f(a8)$

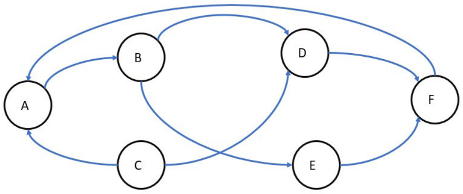


Fig. 7: example of a mesh network

w.r.t Constraint 1, it can be determined that flow $a1$ cannot go along with $a2$, $a5$ and $a6$ in a same timeslot. Hence, if we consider $a1$ to be in the first timeslot either of $a2$, $a5$ or $a6$ shall not be placed within that same timeslot. Therefore, the start time of $a1$ will be precedent to those which are not in dependency with it. Based on these constraints we can build a dependency graph as follows:

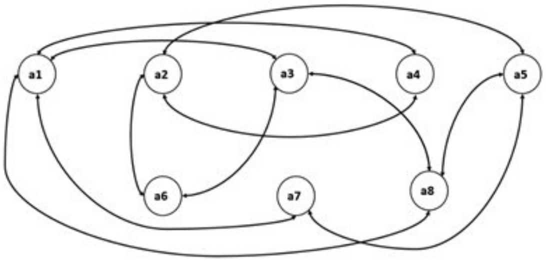


Fig. 8: Dependency graph of the network

Step 2 - Cyclic scheduling formulation:
The schedule of the network will be repeated periodically, the overall flow will be repeated for several cycles, hence making this a cyclic-scheduling allocation problem. Let us consider α to be start time of a transmission. Let $\widetilde{\alpha_1}$ be the start time within the initial period for the transmission $a1$. In our problem we assume to have fixed guaranteed time that are supposed to accommodate the transmission and an eventual acknowledgment. We consider this time index value to be $T_{(i,j)}$.

$$\alpha_1 = \widetilde{\alpha_1} + T_{(i,j)}BI \forall \widetilde{\alpha_1} \in (0, BI-1), T_{(i,j)} > 0 \tag{4}$$

Following $\widetilde{\alpha_1}$, let us consider another transmission say $a2$ and define its formulation. It must be noted that $a1$ and $a2$ do not fall under the same time line. If in a time-line with $a1$ at $\widetilde{\alpha_1}$, $a2$ definitely should be placed at the end of the time-index of $a1$. The value of X is a binary decision that range from either (1,0) depending on the start value accepted for schedule. The precedence constraints for the relative deadlines can be defined as,

$$\widetilde{\alpha}_1 + T_{(c,d)} - BI[X] \leq \widetilde{\alpha}_2 \forall (0, BI), T_{(c,d)} > 0 \tag{5}$$

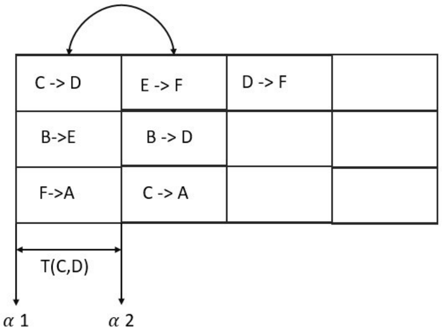
The above given notion is defined in [15] as cyclic scheduling algorithm. $\widetilde{\alpha_2}$ on the other hand can be represented as,

$$\widetilde{\alpha}_2 + T_{(c,d)} - BI[1-X] \leq \widetilde{\alpha_1} \forall (0, BI-1), T_{(c,d)} > 0 \tag{6}$$

Let us consider the time index to be unity for the sake a simplicity in the problem. It also should be noted that the transmissions within dependency graph will have the same start time, for example $a1$ and $a8$ will have the same precedence constraints as both the transmissions call fall under the same timeslot.

The ILP model defined above will be based on the binary decision of X that can be set to 0, so that $\widetilde{\alpha_1}$ starts before the $\widetilde{\alpha_2}$ and the value of BI will be set for a maximum value like 10 before the algorithm is processed.

Step 3 - Schedule placement:
We have implemented the above mentioned algorithm using *linprog* functionality of Matlab. We get the scheduling in such a way that the conflicts are avoided. For the alignment of the transmissions, we place them across several channels and various examples to check its optimality.



C -> D	E -> F	D -> F	
B->E	B -> D		
F->A	C -> A		

Fig. 9: Scheduling results of ILP

Step 4 - Optimal check: The optimality is checked by the following calculation:

$$NT = \lceil (n/R) \rceil \tag{7}$$

In the above equation NT represents the number of timeslots occupied, n represents the total number of transmissions and R is the number of channels used. This formulation has some anomalies which is later addressed under the heuristics. Proving the optimality eventually satisfies both **Constraint 1**and **Constraint 2**.

VI. HEURISTICS

We looked into several existing heuristics that can help provide an optimal solution similar to that of ILP. Some of the heuristics we used to get the optimal solution is listed below:

1. Exhaustive search Exhaustive search is a brute force technique like the *N queens problem* [16]. In this technique, an allocation will be initialized and following the process all possible solutions w.r.t the allocation will be checked exhaustively. For the same example taken in ILP, we used exhaustive search and it took 28 processing times in order to obtain an optimal solution.

The main disadvantage of the exhaustive search algorithm is, it very time consuming in terms of processing to build the overall schedule. This kind of techniques will not be suitable for a time-critical network with large number of nodes with stringent deadlines.

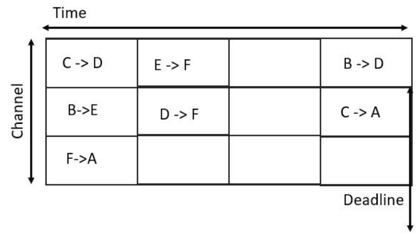
2. Simulated Annealing Simulated annealing [17] is a an allocation algorithm which was built based on the annealing which is a metallurgical process in which the material is heated and cooled in such a way the size of the crystals are altered to reduce the defect. Under simulated annealing the transmissions are chosen intuitively based on the neighboring transmission. At every iteration the annealing process checks if the constraint is obtained and if it is then it moves on to the next level. Unlike the exhaustive search method, simulated annealing takes a subset of transmissions and checks its possibilities of optimal assignment. We have provided the pseudo code for simulated annealing under Algorithm 1.

```
Objective
get an optimized schedule
time initialization:
$t = t_0$
$X(t)$= slot allocation for a transmission
$Z(t) = \begin{cases} 0, & \text{for no scheduling at t} \\ 1, & \text{for scheduling at t} \end{cases}$
Procedure
$X^*_{(i,j)} \leftarrow X_{(i,j)}$
$t \leftarrow 0$
Assign scheduling toward channel 1 timeslot 1
$t \leftarrow t + 1$
generate random (i,j) within the dependency tree
Assign scheduling toward channel 2 timeslot 1
$Z(t) = \Big\{ X_{(i,j)} \epsilon [0, 1]$
iteration
$X_{(i,j)} \epsilon 0$
```

Algorithm 1: Simulated Annealing algorithm

Though simulation annealing is not a brute force technique and it considers based on the dependency tree, it should look for all the possibilities of the allocation based on the random transmission selection. The randomness in the selection procedure and the higher processing times makes this algorithm less suitable despite providing an optimal solution.

3. Maximum Dedicated Timeslot Algorithm MDT scheduling algorithm is a method that aims at maximizing the number of dedicated timeslots to establish substitute paths. This algorithm was implemented for the tree topology, in which dedicated timeslots were used to accommodate the root paths of the tree. This algorithm mainly aims at data packet reliability but the quality aspects in terms of latency gets affected. Considering the transmissions in dependency tree to be route we remapped the mesh network for this algorithm.



C -> D	E -> F		B -> D
B->E	D -> F		C -> A
F->A			

Fig. 10: Scheduling results of MDT

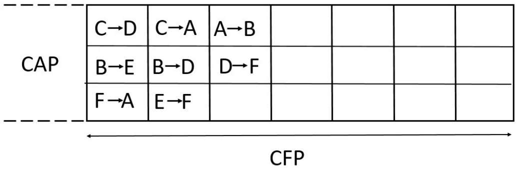
4. Symphony Algorithm
Our algorithm is a two step process, first we provide a rank to the nodes based on the number of transmissions that they make. For explaining our hueristic we consider the same example output from RPL taken for the ILP formulation. From the example, nodes B and C have a transmission rank of 2, as both the nodes have two links formed from them. We denote this transmission based ranking as *TBR* in our algorithm. As an output of TBR, we shall receive subsets of transmissions based on their respective ranks. There can be some transmissions which need to be scheduled prior to others, in such cases to include traffic differentiation, we also provide a priority indicator which can be issued as an information element through the Enhanced Beacon. The transmitting nodes that request for the priority indicator shall be provided a complete timeslot. Thereby, these transmitting nodes will have the liberty to choose among the channels with the best quality to transmit their data robustly. Similar to any constraint satisfaction problem we have a constraint, which is no same node shall occupy the same timeslot. This is denoted as the **constraint** in our algorithm.

In the example we provide in Figure 7, the rank of nodes C, B will be 2, and the ranks of A, D, E, F will be 1. we start placing transmissions from C in adjacent timeslots followed by transmissions from B in the rows of the same timeslots. Then we follow backtracking to assign all the transmissions in accordance to the constraints. Using this algorithm, we receive an optimal solution as shown in Figure 11

The node carrying priority information can request for priority beacon, in which case the PAN coordinator gives a priority flag in its enhanced beacon. Let us take an example that the transmission from node D is given a higher priority, it would result in a schedule shown in Figure 12 Using this feature it can be noted that, not only the transmission with the highest priority gets scheduled first, it also has the ability to occupy the entire timeslot and all the channels. This gives

```
Initialize
step 1
Procedure: make TBR for all the nodes in the network
if TBR succesful then
    return value : go to step 2 ;
else if case of identical ranks then
    Place the elements in a same subset;
else if case priority indicator then
    assign the priority transmission in the full initial
    timeslot;
else
    The transmissions are invalid
end
step 2
Procedure: place the subset with the highest rank
adjacent to each other
Assign adjacent row slots till $subset1 \rightarrow null$
Assign subset 2 in the next row of the first column
if constraint not satisfied then
    place the transmissions in the first row ;
else if constraint satisfied then
    continue placing the transmissions till
    $allthesubsets \rightarrow null$;
else
    The transmissions are invalid
end
```

Algorithm 2: time-frequency scheduling algorithm



C→D	C→A	A→B				
B→E	B→D	D→F				
F→A	E→F					

Fig. 11: optimal schedule solution

the node the freedom to adapt its channel in case of any link quality deterioration, thus improving the robustness of the system.

Compared with simulated annealing and exhaustive searching, the number of processing times is very low in the symphony algorithm. This is due to the ranking process and direct placement of a set of nodes in the timeslots using routing without any exhaustive constraint checking process.

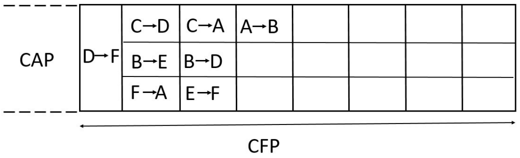
However, the optimality of the algorithm is checked by equation 7, this can be true only in certain cases. The anomalies are listed as follows,

anomaly 1: use of a priority indicator

When used a priority indicator, A priority based transmission is given all the elibility to occupy use the initial timeslot till it is allocated. In this case the optimal solution varies to the following:

$$NT = \lceil (n/R) \rceil + t_{priority-timeslot} \tag{8}$$

anomaly 2: Rank $\geq$ NT



D→F	C→D	C→A	A→B				
	B→E	B→D					
	F→A	E→F					

Fig. 12: example of priority based scheduling

In this case, if the rank of a specific node or a set of nodes is higher than that of the optimality condition. Then, a single node shall occupy the entire length of the timeslots, in such a case the optimality condition will not be met. Despite these anomalies, our algorithm meets the optimal solution reached by any exhaustive search technique and the ILP problem.

VII. PERFORMANCE EVALUATION

The average transmission delay can be calculated for successfully transmitted GTS frames in the multisuperframe.

$$\delta = \sum_{i=0}^{\infty} P^{f}_{(i,m)}(i(MI)) \tag{9}$$

Considering the schedule for routing is carried our every multisuperframe, $P^{f}_{(i,m)}$ is the probability that the GTS is transmitted in the i^{th} superframe of the multisuperframe m. MI is the summation of all the individual BIs (Beacon Interval) within the multisuperframe. To calculate this probability let us take two parameters X^{s}, the total number of GTS that is successfully transmitted, and $X^{S}_{(i,m)}$, the number of GTS that have to wait i superframes within a multisuperframe for its successful transmission. The probability will be 1, if considered all the superframes in case of delayed transmission. Using these parameters the probability $P^{f}_{(i,m)}$ can be formulated as:

$$P^{f}_{(i,m)} = X^{S}_{(i,m)} / X^{S} \tag{10}$$

The first set of GTS frames based on the symphony schedule that gets successfully placed in the initial attempt need not wait the next, let us consider as $X^{S}_{(o,m)}$. These include all the transmissions in all the available channels (m) of the initial superframe.

$$\begin{gathered} X^{S}_{(o,m)} = K(1-P_e), \\ where\ m = (0-16)\ and\ K\ \epsilon\ (0,1) \end{gathered} \tag{11}$$

X can be denoted as the guaranteed transmissions that were not able to transmit along the first timeslot. The value of X will be incrementing as with the failures to accommodate a successful transmission. Now the the GTS superframes that get transmitted successfully in the adjacent superframe can be denoted by $X_((1,m))^{S}$, this value can be formulated as:

$$X^{S}_{(1,m)} = HK(1-P_e) \tag{12}$$

where, H is the probability of failure to get accommodated within the initial transmission.The value of H can be given as $P_e e^{-BI \cdot m \cdot i\lambda}$, this probability is with an assumption that all the transmissions shall be carried out within the multisuperframe with i superframes and m channels with a GTS arrival rate of λ. Generalizing for all the i superframes, the successful transmissions can be denoted as:

$$X^S_{(i,m)} = H^{(i)}K(1-P_e) \tag{13}$$

The value of the successfully transmitted GTS can be formulated as:

$$X^S = \sum_{i=0}^{\infty} H^{(i)}K(1-P_e) \tag{14}$$

using the aforementioned equations, the probability to be transmitted in the i^{th} superframe can be calculated as:

$$P^f_{(i,m)} = (1-H)\cdot\ H^i \tag{15}$$

and the overall average delay of the network can be given as:

$$\delta = \sum_{i=0}^{\infty}(1-H)\cdot\ H^i(\epsilon + i(MI)) \tag{16}$$

For the analysis we consider a multisuperframe with 2 superframes over 3 channels, we consider three arrival rates for the mathematical analysis. From the Fig 13, it can be understood that lower the values of λ the delay is significantly decreased, this is due to the decrease of the delay in the inter-arrival rates, this will also increase the overall throughput of the network. The possibility of multichannel in DSME also contributes to lesser delay, in our case we have considered an ideal case of symphony in which all the timeslots in the channels are accommodated without any failure.

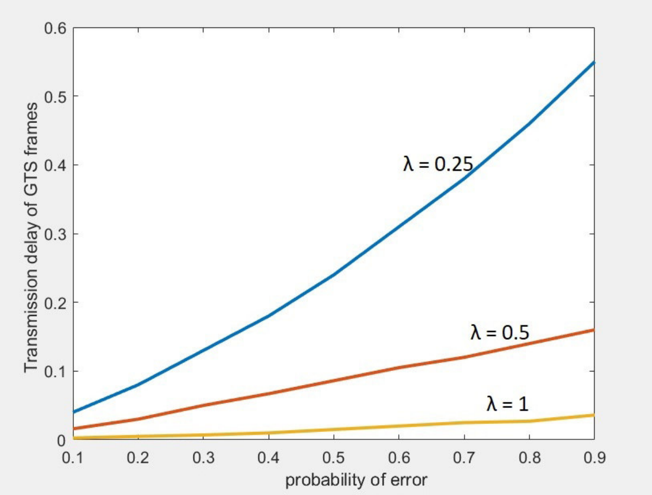


Fig. 13: Pe vs GTS delay

We use the same formulation to calculate the delay of schedule placement within a superframe. For this very special case, we take the value of H and replace with H_{tslot} which is the probability of failure to accommodate within the initial timeslot. This aforementioned value can be expressed as:

$$H_{tslot} = P_e e^{-T_s \cdot m \cdot i\lambda} \tag{17}$$

For our analysis, let us consider that all the timeslots have an equal size for all the i superframes in the multisuperframe.

using this we can derive a formulation for the delay for single GTS that fails to occupy the first timeslot and moves to the next. utilizing the aforementioned details in Equation 1, we can derive the delay for a timeslot to be,

$$\delta_{timeslot} = P_e e^{-T_s \cdot m \cdot i\lambda}(T_l)/(1 - P_e e^{-T_s \cdot m \cdot i\lambda}) \tag{18}$$

For this analysis, we will consider the length of the timeslot T_l to be common symphony,MDT and brute-force FIFO algorithms taken for comparison. As the nodes start get accommodating in the channels and move from one timeslot to another, the value of T_l starts to increase with an unit. We calculate the transmission delay of the GTS frames for all the cases for every timeslot. The analysis shown below provides the Transmission delay of the GTS frames for a set of transmissions for different arrival rates and common probability error ratio of 0.5.

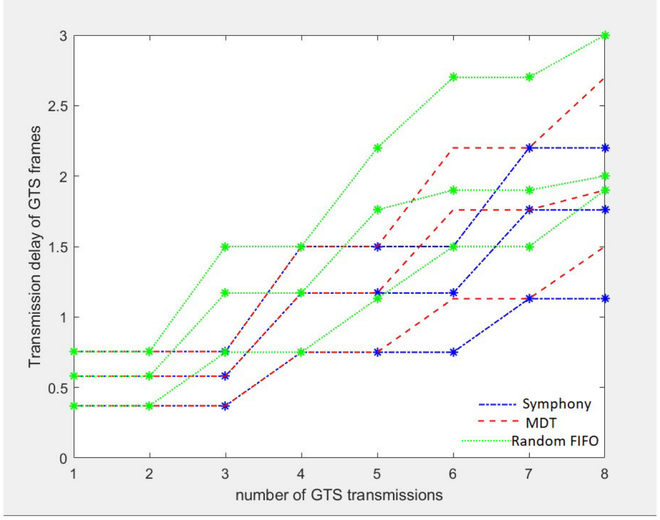


Fig. 14: number of transmissions vs GTS delay

Symphony provides better results when compared to the MDT and the random FIFO methods. MDT under performs in this scenario because it spares timeslots aiming better reliability of the network. Symphony on the other hand aims at filling all the timeslots channel wise thus eventually leading to lesser transmission delay.

ACKNOWLEDGMENTS

This work was partially supported by National Funds through FCT (Portuguese Foundation for Science and Technology) within the CISTER Research Unit (CEC/04234); also by FCT and the EU ECSEL JU under the H2020 Framework Programme, within project ECSEL/0002/2015, JU grant nr. 692529-2 (SAFECOP). The authors would like to thank the Robotics and Internet of Things (RIoT) Unit at Prince Sultan University's Innovation Center for their support to this work.

REFERENCES

[1] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of lora: Long range & low power networks for the internet of things,” *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[2] Z. Zhang, F. Yu, and B. Zhang, “A depth-based tdma scheduling for clustering sensor networks,” in *Frontier of Computer Science and Technology, 2009. FCST’09. Fourth International Conference on*. IEEE, 2009, pp. 261–266.

[3] J. Lee and W.-C. Jeong, “Performance analysis of ieee 802.15. 4e dsme mac protocol under wlan interference,” in *ICT Convergence (ICTC), 2012 International Conference on*. IEEE, 2012, pp. 741–746.

[4] G. Alderisi, G. Patti, O. Mirabella, and L. L. Bello, “Simulative assessments of the ieee 802.15. 4e dsme and tsch in realistic process automation scenarios,” in *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*. IEEE, 2015, pp. 948–955.

[5] P. Du and G. Roussos, “Adaptive time slotted channel hopping for wireless sensor networks,” in *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th*. IEEE, 2012, pp. 29–34.

[6] P. K. Sahoo, S. R. Pattanaik, and S.-L. Wu, “A reliable data transmission model for ieee 802.15. 4e enabled wireless sensor network under wifi interference,” *Sensors*, vol. 17, no. 6, p. 1320, 2017.

[7] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1567–1576.

[8] T. Van Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 171–180.

[9] N. Kushalnagar, G. Montenegro, and C. Schumacher, “Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals,” Tech. Rep., 2007.

[10] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, “From manet to ietf roll standardization: A paradigm shift in wsn routing protocols,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 688–707, 2011.

[11] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, “6tisch: deterministic ip-enabled industrial internet (of things),” *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, 2014.

[12] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust mesh networks through autonomously scheduled tsch,” in *Proceedings of the 13th ACM conference on embedded networked sensor systems*. ACM, 2015, pp. 337–350.

[13] R. W. Floyd, “Nondeterministic algorithms,” *Journal of the ACM (JACM)*, vol. 14, no. 4, pp. 636–644, 1967.

[14] T. Herman and S. Tixeuil, “A distributed tdma slot assignment algorithm for wireless sensor networks,” in *Algosensors*, vol. 3121, 2004, pp. 45–58.

[15] Z. Hanzálek and P. Jurcik, “Energy efficient scheduling for cluster-tree wireless sensor networks with time-bounded data flows: Application to ieee 802.15. 4/zigbee,” *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 438–450, 2010.

[16] Q. Weisheng, “N queens problem,” *Journal of Mathematics*, vol. 2, p. 002, 1986.

[17] P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing,” in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.