International Journal of Advanced and Applied Sciences, 9(10) 2022, Pages: 81-93

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Optimized scheduling method in 6TSCH wireless networks

Ines Hosni [1, *], Ourida Ben Boubaker [2]

[1]*Department of Information Systems, College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia*
[2]*Department of Computer Sciences, College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia*

ARTICLE INFO

Article history:
Received 15 January 2022
Received in revised form
6 May 2022
Accepted 30 June 2022

Keywords:
6TSCH scheduling
Minimum scheduling function
IEEE 802.15.4e
Clustering

ABSTRACT

IEEE802.15.4e-TSCH is a mode exploited by the Internet of Things. Time Slotted Channel Hopping (TSCH) presents an upgrade to the IEEE 802.15.4 to build a Medium Access Control (MAC) for low power and loss network applications in IoT. This norm defines the concept of TSCH based on channel hopping and reservation of bandwidth to achieve energy efficiency, as well as consistent transmissions. Centralized approaches have been proposed for planning TSCH. They have succeeded in increasing network efficiency and reducing latency, but the scheduling length remains not reduced. However, distributed solutions appear to be more stable in the face of change, without creating a priori assumptions about the topology of the network or the amount of traffic to be transmitted. A distributed scheduling allowing neighboring nodes to decide on a coordination system operated by a minimal scheduling feature is currently proposed by the 6TiSCH working group. This scheduling allows sensor nodes to determine when data is to be sent or received. However, the details of scheduling time intervals are not specified by the TSCH-mode IEEE802.15.4e standard. In this work, we propose a distributed Optimized Minimum Scheduling Function (OMSF) that is based on the 802.15.4e standard TSCH mode. For this purpose, a distributed algorithm is being implemented to predict the scheduling requirements over the next slotframe, focused on the Poisson model and using a cluster tree topology. As a consequence, it will reduce the negotiation operations between the pairs of nodes in each cluster to decide on a schedule. This prediction allowed us to deduce the number of cells needed in the next slotframe. Clustering decreases, the overhead processing costs that produce the prediction model. So, an energy-efficient data collection model focused on clustering and prediction has been proposed. As a result, the energy consumption, traffic load, latency, and queue size in the network, have been reduced.

1. Introduction

With the adoption of new technologies, Industry 4.0 is appeared, employing IoT, Cyber-Physical Systems (CPS), and Cloud technologies (Kurunathan et al., 2018). Although there are multiple IoT architectures, which improve communication efficiency, the requirements and the urgency demand greater efficiency, speed, and reliability, known as Quality of Service (QoS). To address this problem, Wireless Sensor Networks (WSNs) have emerged and become one of the most important network infrastructures. These stringent qualities of service requirements of communication protocols have traditionally been addressed by making modifications to the IEEE 802.15.4 standard and proposing external mechanisms. The main challenge of industrial WSNs is to minimize the latency time given the critical nature of the physical events detected by the sensor nodes. In other words, in order to evaluate their bandwidth needs, the challenge is to decrease the rate of control packets exchanged between nodes. The TSCH mode is planned to permit sensor nodes to endure a wide range of applications, including industrial applications, by implementing the 802.15.4 protocol (Papadopoulos et al., 2017). This mode consists of a technique for accessing the communication medium using time synchronization between nodes in order to attain an operational level with low power consumption. On the other hand, TSCH implements

* Corresponding Author.
Email Address: itabbakh@ju.edu.sa (I. Hosni)
https://doi.org/10.21833/ijaas.2022.10.011
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-3349-5786

frequency hopping to allow the network to achieve an advanced level of reliability. The 802.15.4e amendment is the latest standard proposed by IEEE for low-power WSNs. This standard is implemented in an industrial environment with high requirements in terms of reliability, availability, and security. In this environment, deploying sensors in parallel with metallic equipment results in signal degradation due to interference, thus blocking the use of a single channel for communication. However, TSCH mode, implemented in the 802.15.4e standard, allows greater agility when using communication resources by providing greater reliability to the network. TSCH mode only focuses on MAC layer operation. As a result, the 6TiSCH Working Group was formed by the IETF Standardization Organization to link the IEEE802.15.4e TSCH-mode capabilities to previous standardization efforts and recommendations such as 6LoWPAN and ROLL (Routing Over Low-power and Lossy networks) (Verma et al., 2016). The 6TiSCH group offers an architecture based on open source standards with the objective of achieving high performance at the industrial level with regard to latency, reliability, and power consumption. However, the group does not define how to plan and schedule the sending and receiving of frames between nodes in the network and during the time slots that define the scheduling (Palattella et al., 2015).

This work aims to develop a scheduling algorithm able to meet the needs of industrial WSNs in terms of latency and energy. In other words, it aims to reduce the amount of control traffic exchanged between nodes to determine their requirements of bandwidth. The proposed solution is based on a distributed algorithm that allows two nodes in a cluster to calculate the number of cells required to exchange their packets. Depending on the mode of operation of the network, the chain and sequence of events were modeled as a Poisson process. The algorithm is based on a probability calculation that aims at predicting in a later stage the number of cells needed for a pair of nodes in cluster topology. This will minimize the control packets number exchanged and thus reduce the latency time. Also, clustering can reduce the energy consumption used for data collection. So, there will be a proposal for an energy-efficient data collection model focused on clustering and prediction. The remainder of this paper is structured as follows. We present an overview of 6TiSCH via TSCH 802.15.4e mode in Sections 2 and 3. In the two following sections, we discuss some context details and related work. Sections 6 and 7 explain the nature of the system. In section 8, we present the results of the simulation and conclude the article in section 9.

2. IEEE802.15.4e standard

IEEE 802.15.4 standard presents the benchmark for wireless sensor networks. This describes the service of low-speed local area wireless networks (LR-WPAN) and identifies the physical layer and the layer of media access control (Vallejo et al., 2013). These networks were very established, though retaining a simple and versatile protocol stack, for their ease of implementation, extremely low cost, short-range operation, reliable data transfer, and fair battery life (Mohamadi et al., 2020). The efficiency of the IEEE 802.15.4 standard network in the WSNs has been analyzed by many studies. However, many drawbacks and flaws have been found, making this standard inadequate for sensitive applications. These, typically operating in harsh environments, have strict criteria in terms of reliability, latency, power efficiency, and scalability. Lack of reliability, infinite latency, a built-in frequency hopping technique, and poor power management are among these limitations (Mohamadi and Senouci, 2018). These flaws render the IEEE 802.15.4 standard inappropriate for several systems, particularly where the reliability and latency of these systems are very demanding. In 2012, as an update to the IEEE 802.15.4 standard, the IEEE Standards Association Council accepted the IEEE 802.15.4e (Palattella et al., 2012b). Standard in order to better serve the numerous industrial fields of implementation. Additional features such as low power consumption, information elements, improved tags, MAC performance metrics, and quick pairing are supported by the new standard.

3. Time slotted channel hopping TSCH

The TSCH mode is configured primarily to schedule the data communication to the network nodes and their respective connections (Zand et al., 2012). The communication between the nodes takes place according to a schedule, thus the neighboring nodes, whose transmissions can interfere, will not be scheduled to transmit on the same time slot (slot offset) and channel offset (Channel Offset). The scheduling is projected as a matrix composed of a slot offset and channel offset, where each cell represents a specific link and can be reserved for a single link or be shared between multiple links. A system called backoff is defined by the IEEE 802.15.4e MAC protocol in order to resolve this conflict. In addition, for shared slots, the slotted CSMA/CA algorithm is used by the nodes of the PAN -TSCH (Accettura et al., 2012). The standard IEEE 802.15.4e TSCH-mode defines just how the MAC layer is run but does not explain how scheduling is managed, designed, and modified, or how to adjust to network traffic constraints (Chang et al., 2015).

3.1. Slotframe structure

The nodes synchronize in TSCH mode on a regular frame (called a slotframe) consisting of a certain number of time intervals (called timeslots). A slotframe is a sequence of a set of timeslots. Communications may be contention-based (i.e. using CSMA-CA) or non-contention-based in either time slot (Choudhury et al., 2020). The duration of each timeslot makes it possible to send a packet and obtain notification of reception. A slotframe size is determined by the number of time slots in the slotframe (Papadopoulos et al., 2017). Each slotframe occurs cyclically, thus creating a schedule for communication. Every node receives timing, channel hopping, timeslot, and slotframe data from Enhanced Beacons (EBs) that are sent to the network regularly by other nodes. It synchronizes on the network when a node receives a valid EB (Papadopoulos et al., 2017), initializes the slotframe, and sends its own beacons. From that moment on, the slotframe restates dynamically itself according to the notion of time shared by the nodes and does not necessitate beacons to start interactions.

3.2. Channel hopping

TSCH mode multi-channel communication relies entirely on channel hopping. TSCH mode can use up to 16 communication channels defined by a channel offset. In TSCH mode, the communication link between a pair of nodes is identified by the pair [n, channel offset]. This is an assignment of consecutive" n" timeslots and channel offset for pairwise contact.

You may describe the frequency used for communication by the f function.

$$f = F(ch_{offset} + ASN) mod\ N_{ch} \quad (1)$$

In Eq. 1, Nch is identified as the number of channels used for the current network, since using all 16 channels is not mandatory. Some channels may not be used in an attempt to improve energy consumption or worsen its consistency. As noted previously, ASN helps calculate the timeslots number that has occurred since the network was entered by the node. From Eq. 1, note that for an increased ASN, another channel can be introduced with the same offset, i.e., the channel hopping technique can be used on the same link with various frequencies.

4. Scheduling in 802.15.4TSCH

Networks as stated in the literature, the scheduling issue has indeed increased concern on TDMA networks (Ergen and Varaiya, 2010). Despite this, most current multi-channel scheduling schemes for TSCH networks are not appropriate. They have not been developed for nodes with limited bandwidth, do not enable packet channel hopping, and are not effective for channel use. New scheduling schemes developed for TSCH networks were also developed by the researchers. There are different approaches, which could be used to establish the scheduling. The IEEE 802.15.4-2015 standard in TSCH mode allows the upper layers to create delays that all nodes must respect. This allows nodes to communicate with each other in a multihop, thus making the information flow from the initial point to the collection point easier and faster. A node implements global and local scheduling by allocating cells to each respective flow to aid in the sharing of cells containing information between clients.

5. Related works

The design of a schedule is unique to the application and many scheduling schemes for planning TSCH networks have been identified. In order to determine the schedule, creative approaches may be used. It is possible to identify them as centralized and distributed. In a centralized approach, a single coordinator node is responsible for planning and building all communications, as well as maintaining network scheduling. A centralized machine called the Path Computation Element may be the scheduler (PCE) (Farrel et al., 2006). The Traffic Aware Scheduling Algorithm (TASA) is a centralized scheduling algorithm for IEEE802.15.4e TSCH-mode networks proposed in (Palattella et al., 2012a). This approach considers the topology of a tree structure and reflects on a converged cast scenario where the coordinating node must be supplied with various amounts of data. TASA's primary objective is to establish the best schedule, reducing the number of slots required to send all information to the coordinator. This scheduling may be achieved through the process of matching and coloring. The TASA method applies the matching algorithm at every phase to pick a collection of suitable schedule links in the same time slot. Then, for each connection selected in the previous step, a vertex coloring algorithm applies the various channel offsets. In addition, the authors found that the use of more channels would increase network efficiency, reduce latency and improve energy efficiency considerably. Soua et al. (2012) planned the MODESA (Multi-channel Optimized Delay Time Slot Assignment) method. Unlike TASA, MODESA aims for coherent traffic conditions where the same number of packets are generated by all nodes. Conflict-free scheduling in TASA is constructed using an iterative method. TASA picks a set of links at each step and arranges their transmissions in the same timeslot, using several channel offsets, if required (Mohamadi and Senouci, 2018). The MODESA method selects a single node and chooses a single channel to support one of its mandatory transmissions. Additionally, by first scheduling nodes that provide more packets in their queues, MODESA decreases queue congestion, while TASA does not take queue congestion into consideration. This method was enhanced and built-in (Soua et al., 2013), in order to maintain diverse traffic, as well as several coordinators.

Jin et al. (2016) studied an adaptive, centralized, and multi-hop (AMUS) scheduling method based on the TSCH mode is proposed. At the PCE unit situated network, the authors introduced their approach and used a flexible application layer protocol (the CoAP protocol) to gather the data needed to measure the schedule. The AMUS approach enables a multihop planning sequence (MSS) to offer low latency and distributes extra resources to susceptible connections in order to significantly minimize the delay created by conflict or collisions. This approach exceeds TASA in improving contact efficiency and also achieving exceptionally low delay. Ojo and Giordano (2016) formulated the scheduling issue as a throughput maximization issue and the delay as a minimization issue. They recommended using the theory of charts based on the correspondence theory to solve the problem of maximizing throughput centrally. The results indicate that a very high velocity is obtained by the proposed system. The same issue was developed as the problem of optimizing energy efficiency (Ojo et al., 2017). In this work, the authors have presented an Energy Efficient Scheduler (EES) that works better than the Round Robin Scheduler (RRS) in terms of energy consumption, although guaranteeing an improved data transfer rate.

Unlike centralized scheduling, distributed solutions appear to be more stable in the face of change, without creating a priori assumptions about the topology of the network or the amount of traffic to be transmitted. When implementing a distributed approach, each node needs to negotiate with neighboring nodes and define locally which links to plan with them. Decentralized Traffic Aware Scheduling DeTAS is the distributed version of the TASA method (Accettura et al., 2013). This method is intended for networks of multi-coordinators. Therefore, to build the overall schedule, it utilizes mixed micro-scheduling. All micro-scheduling uses a collection of specific channels to prevent interference. TASA has been compared to the DeTAS approach and the results obtained indicate that the former offers better management of queues. Besides, DeTAS guarantees a high duty cycle, end-to-end delay, and packet loss ratio efficiency (Packet Loss Ratio) (Accettura et al., 2015).

The DiSCA solution (Distributed Scheduling for Converge cast in Multi-channel Wireless Sensor Networks) recognizes two types of transmission: without an acknowledgment of receipt and with an acknowledgment of receipt (Soua et al., 2015). On each iteration in this algorithm, a node schedules a transmission following a set of rules. Each iteration provides microscheduling. This algorithm is likely to interlock in order to minimize the total slot number. The authors compared DiSCA to Soua et al. (2012) and to Soua et al. (2016) and the results obtained indicate that DiSCA is very similar to optimal scheduling with a limited number of control messages.

Wave for IEEE 802.15.4e convergecast networks is a distributed scheduling algorithm (Soua et al., 2014). Each node in the network is aware of its nodes in conflict and of its parent node. The network is filled with a number of waves, the first of which is caused by a START message transmitted by the coordinator. When a node has precedence over its conflicting nodes and receives this message, a cell in the wave is allocated to it and notifies its conflicting nodes by submitting an ASSIGN message. This process repeats till cells are selected by all nodes. The coordinator transmits a REPEAT message to trigger a second wave after the first wave has been transmitted, which strengthens the first one. This process repeats once all nodes could schedule all the packets that are in their queues. The results of the simulations made by the authors demonstrate that, compared to DiSCA, the Wave method reduces scheduling length (Soua et al., 2015) and holds out as an important algorithm for distributed scheduling.

Demir and Bilgili (2019) suggested a distributed diverge cast scheduling algorithm called DIVA. Unlike converge cast traffic, diverge cast traffic flows in all ways and not just from root nodes. Each node begins by distributing a request for a CON-REQ connection. If the CON-ACK connection acknowledgment packet receives a response, a link between these two nodes will be created. Both nodes in the network perform this process until the full slotframe duration is reached. This strategy has been compared to that proposed by Tinka et al. (2010) Unfortunately, DIVA does not boost Aloha, but accepts all slotframe sizes.

6. Proposed system

6.1. Requirements

The 6top (6p) protocol enables a 6TiSCH network's neighboring nodes to add or delete cells from their schedules. It is part of the 6TiSCH IEEE802.15.4e sublayer of operations, which provides frameworks in this type of network for performing distributed orchestration. It is the scheduling feature that determines when cells should be added or removed, so 6p is used to efficiently assign resources. The 6p protocol performs a so-called 6p transaction when new cells need to be added or removed, which includes the negotiation of adding or removing cells between a pair of nodes. The nodes hold their own orchestrations in the case of networks with distributed scheduling. In terms of signaling duration between nodes, this guarantees better performance. Less signaling also suggests, however, that nodes are less knowledgeable about the network, making it more difficult to generate efficient scheduling.

6.2. Design constraints

During the design phase of the scheduling algorithm, we defined some operating constraints, which are essential when running our solution. Each node keeps changing the number of resources assigned to its neighbor nodes depending on both its current allocation of resources and its own resource needs. The minimal scheduling function does not take into consideration the recurring traffic load, which means that each allocated cell reiterates at each slotframe and subsequently wastes assets in the event that the generation of packets in the network is not as frequent. In addition, during the phase of determining the bandwidth required for communication between a pair of nodes, the number of exchange messages must be controlled. This will reduce the waiting time during end-to-end transmission and also minimize the packet number in the memory queue.

6.3. Network model

In TSCH mode networks, defined by 802.15.4e, when a node senses a sudden fluctuation of a physical event, it creates and queues a large flow of data packets in the memory of the sensor node. This node checks to see if it has enough bandwidth to transmit these packets to its parent node. Otherwise, the node checks, in the next slotframe, if it has enough reserved cells with its parent and compares if it is able to forward those packets during those cells. Checking the bandwidth with the parent node triggers a high number of transactions per packet, thus generating a considerable increase in the use of energy resources. When the limited number of transmissions and retransmissions is exceeded, there will be a loss of packets which results in the depletion of the node's power resources. We suggest a new scheduling algorithm based on the minimal scheduling feature provided in the IETF draft (Chang et al., 2020) and applied it to cluster topology to minimize communication failures, the number of packet losses, and end-to-end latency. The developed algorithm consists of two principal processes: calculating the average number of packets each node produces and predicting the number of cells needed in the next slotframe. We present these descriptions:

- Description 1: Data are sent to receivers in an upstream transmission mode.
- Description 2: The network topology is presented by the form G=(N, L), where N represents the set of nodes in the network and L denotes the set of communication links.
- Description 3: A model of a network made up of a coordinator and N nodes. Nodes may be full-function devices (FFDs) or reduced-function devices (RFDs) according to their capability and available resources. FFDs receive packets from sensor nodes held by their respective clusters, also they forward traffic to the higher level in the tree till reaching the sink. In a cluster-tree topology, by organizing RDF and FFD at various hierarchical levels, the tree depth is obtained (Fig. 1).

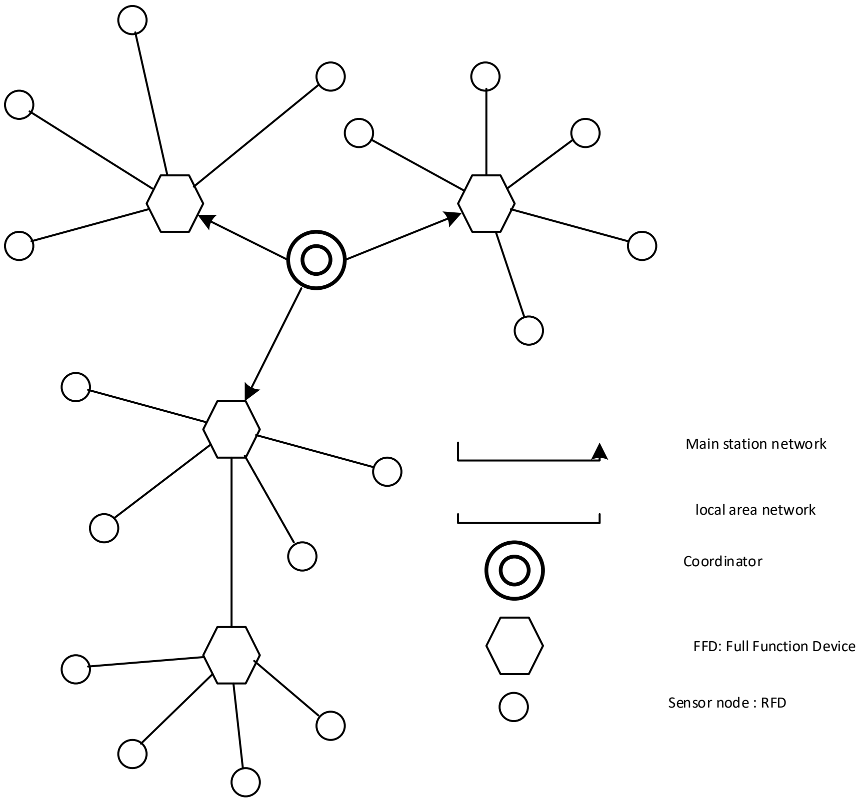


Fig. 1: Cluster tree topology

- Description 4: The cluster tree topology consists of a routing tree of a node n containing FFD(n), tree(n), and RFD(n). Where, FFD(n) is the cluster head and represents the parent node of node n, tree(n) is the subtree of the routing tree nested at node n, and finally RFD(n) represents the child nodes of a given parent (FFD(n)).
- Description 5: In a collection frame from any node n ∈ G, we denote by G(n) the number of packets sent by RFD(n). We also denote by T(n) the total of all transmitted packets, comprising those sent by RFD(n), and the number of the received packet by FFD(n). So, we present T(n) by Eq. 2:

$$T(n) = \sum_{n \in Tree(n)}^{\infty} G(n) \qquad (2)$$

- Definition 6: We denote Q(n) as the number of queued packets of RFD(n) that are to be transmitted to the parent FFD(n). We also denote C(n) the number of allocated cells between the cluster head F FD(n) and a son RFD(n).

- Definition 7: After the execution of the scheduling algorithm, an add, delete, or maintain cell transaction is triggered in the next slotframe S_{i+1}.

7. Optimized minimum scheduling function

7.1. Predicting data amount

OMSF is a distributed scheduling protocol based on the MSF protocol which aims to determine when to increase or decrease the bandwidth between two neighboring nodes (add/remove cells) by interacting with the 6top sublayer. Unlike the MSF protocol, which retrieves statistics from the 6top sublayer to finally make decisions about adding or removing cells, OMSF is based on a statistical calculation that is done at the MAC layer. This reduces the number of control packets exchanged between a parent node and its child node in the cluster.

In the 802.15.4e TSCH networks, a node can transmit two types of packets: periodic denoted by D^w_i (n) and event denoted by D^v_i (n), as shown in Fig. 2.

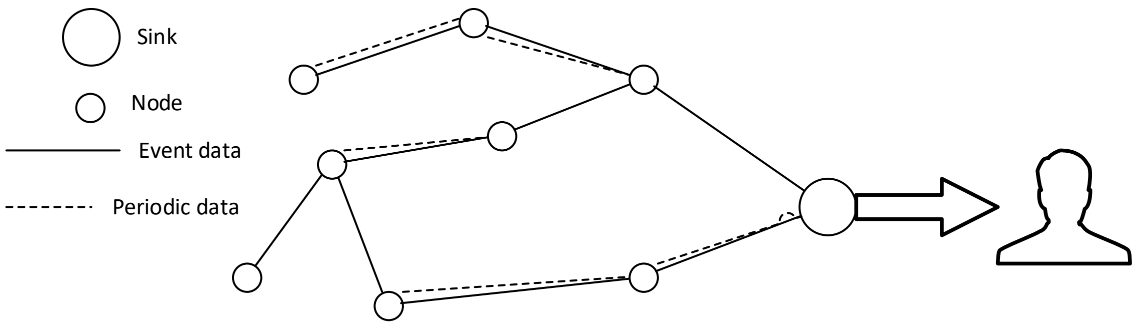


Fig. 2: Data types in the TSCH network

Periodic packets are made up of beacons called enhanced beacons containing ASN data of the slotframe, its length in seconds, etc... This information is used when a node tries to join the network and also for maintaining synchronization between nodes in the network. Event data is sent when a physical event is detected by the node. The algebraic flow generated by a node, not by D^T_i (n) is illustrated as follows:

$$D_i^T(n) = D_i^W(n)+D_i^W(n) \quad (3)$$

which can be developed as follows:

$$D_i^T(n) = \sum_{i\neq k} D_{i-sin\,k} + \sum_{k=1}^{i-1} D_{k-sin\,k}\text{ , i<1<sink} \quad (4)$$

this can be translated to Eq. 5:

$$D_i^T(n) = \sum_{i-1}^{i} Tree(n) + \sum_{i-1}^{i} G(n)\text{ , i<1<sink} \quad (5)$$

7.2. Poisson process generation model

In our model, we consider TSCH-mode 802.15.4e sensor networks, which transmit packets only when they detect a change or deviation from a physical event taking into consideration the nature of the way packets are sent and received in the network. We may follow a Poisson process model to formulate data packet generation in a cluster tree network.

7.3. Mathematical formulation

First, we represent by E (t) the number of events that happen in an interval of time (0, t], and we undertake that E(0) = 0. The process E (t); t ≥ 0 is the counting process and it verifies the following conditions: (Paul and Baschnagel, 2013).

∀t ≥ 0, E(t) ∈ E t → E(t) is increasing ∀0 < a < b, E(b) − E(a) denotes the number of events that have taken place in time interval [a, b]. In our model, the counting process is considered to be incrementally independent because the events (transmission/reception of data packets) that occur in a disjoint time interval (the duration of a slotframe) are independent. In addition, we adopt the following conditions:

- Condition 1: Events that repeat in disjoint time intervals are independent; a couple of nodes begin the exchange of packets during slotframes if and only if an event is detected. Therefore, for any choice of real numbers 0 ≤ t1 < t2 < ... < tn, the random variables E(t2) − E(t1), E(t3) − E(t2), ..., E(tn)- E(tn − 1) are mutually independent.
- Condition 2: A pair of nodes transmit packets regardless of their state during the previous slotframe. We can then conclude that for any positive real number t and h, the number E(t + h) − E(t) of events that occur during a time period (t, t + h] is independent of the value t and depends only on the interval length h.
- Condition 3: In the 802.15.4e TSCH standard mode, the maximum length of a slotframe must not exceed 101-time slots with a duration of 15 milliseconds each. This means that the length of h of the time interval over which we count the events is reduced, so we can deduce that the probability of observing more than one event is almost zero. g (h) here is a small order function of h.

$$\Pr[E(t+h) - E(t) \geq 2] = g(h) \quad (6)$$

$$\Pr[E(t+h) - E(t) \geq 1] = \lambda h + g(h) \quad (7)$$

The allocation of the packets number, E(t), designed by each network node along the slotframe length is calculated as follows:

$$\Pr(E(t) = n) = \frac{\lambda t^n}{n!} = e^{-t} \quad (8)$$

where, λ represents the average value of the number of packets generated and transmitted between a pair of nodes in a cluster since they have been synchronized to the network scheduling. Once the nodes are synchronized, the 802.15.4e standard implements the minimal scheduling function which ensures minimal network operation. Mathematically, λ is defined as follows:

$$\lambda(n) = \frac{\sum_{i=1}^{T} Pkt_i(n)}{S_T - S_i} \quad (9)$$

where, T represents the current time, $Pkt_i(n)$ are the generated packets number by node n at time T = i, S_T represents the slotframe number at T and S_i represents the slotframe number at T=i. Let λ = S_i – S_T is the total of numbers preceding past slotframes. In order to have a precise value of γ, the 802.15.4e standard implements the scheduling function until slotframe number 10.

7.4. Add/remove cells (6top)

The 6TiSCH Operation (6top) sublayer is the layer immediately above the IEEE Std 802.15.4 TSCH communication media access control layer. The roles of the 6top underlay are as follows:

- Allows neighbor nodes to communicate to add/remove cells from each other.
- Execute one or more 6top scheduling functions, which precise the measures determining when to add/delete cells. Once a node joins a 6TiSCH network, it will be able to add, delete or move cells with its preferred parent node for the following three reasons:
- Adapt the connection layer's resources to the traffic between the node and its chosen parent.
- Manage the change of preferred parent (triggered by the RPL protocol).
- Manage a collision in the scheduling.

We only focus on where there will be a change in the scheduling to accommodate network traffic. The proposed algorithm uses the minimal scheduling function in the event of a collision or a change of parent. Our solution provides statistics on bandwidth usage at the 6top sublayer so that you can make decisions about adding or removing cells. From slotframe γ = 10, each node (parent or child) executes the algorithm described in the next paragraph. The aim is to estimate the packet number that will be created in the next slotframe.

In order to reduce the resource use made by every node, the algorithm terminates its execution when achieving a maximum probability of packets producing λ. By identifying the packet number that will be produced in the next slotframe, a node may estimate the number of cells required to share data with its FFD.

A node may also do a 6p transaction with its parent to update (increase/decrease) cells to the TSCH schedule between the two nodes, depending on the performance of the algorithm (Fig. 3).

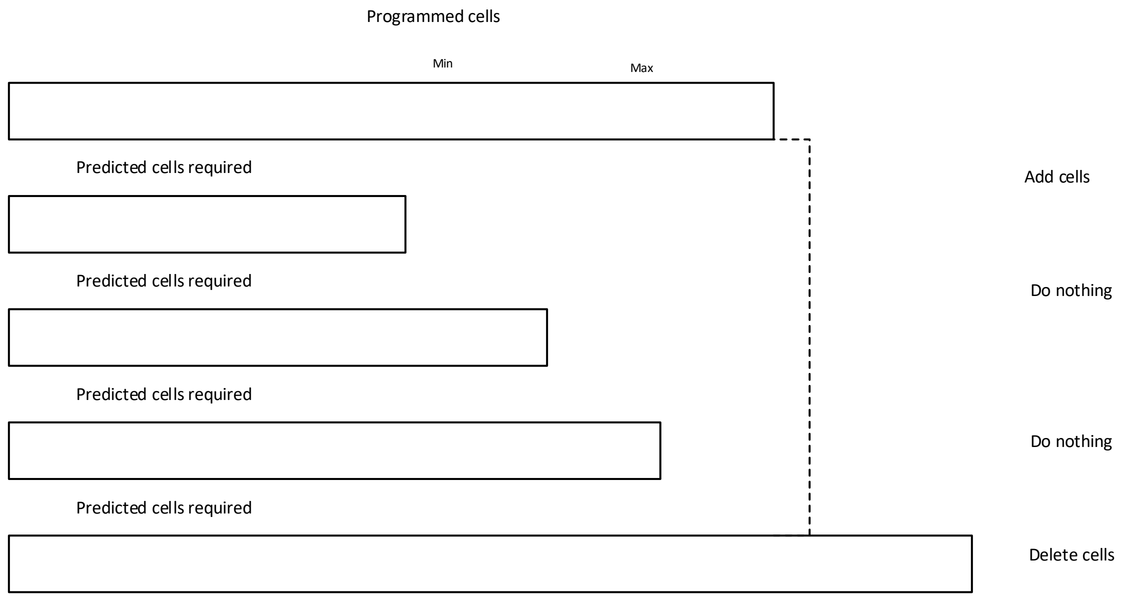


Fig. 3: Add/delete cells

7.5. Scheduling algorithm

In the algorithm below, each node will perform certain calculations based on statistics collected from its parent in order to predict how many cells it will need during the next slotframe. This prediction replaces the method used in the minimal scheduling function which is based on an exchange of control packets to calculate the bandwidth required for communication between a pair of nodes. As mentioned previously, this prediction takes place at a single node level and will therefore avoid a packet exchange.

Algorithm 1: *Prediction algorithm applied on Slotframe S*

Require*: cluster node n, G(n): the number of packets sent by node n*
Ensure:
For *each cluster* ***do***
For *S=S11 to simulation time;* S_{i++} ***do***
Calculate λ from Eq. 9
$\lambda \leftarrow n$
while pr_{max} ***do***
pr←Calculate(Pr E(t), λ): packet generation probability
If *(pr ≥*pr_{max}*)* ***then***
Maxi←pr
End if
End while
Return pr: Maximum probability
If *(pr< Pr(Cells(n))) : (Cells(n) the present allocated cells between the parent FFD(n) and the node RFD(n))* ***then***
6p_REMOVE(Cells)
Else
If *pr > Pr(Cells(n))* ***then*** *6p_ADD(Cells)*
end if
end if
end for
end for

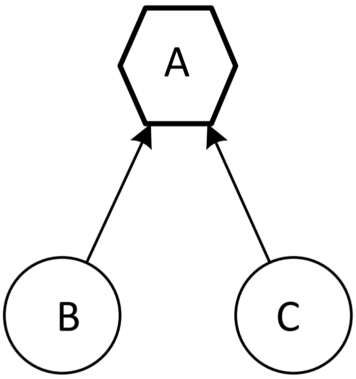
7.5.1. Calculating the mean

In order to have a precise value of the average generated packets between a pair of nodes, the algorithm uses Eq. 9 from the slotframe S = γ = 11. This manipulation is repeated at each start of a slotframe until the node becomes desynchronized from the network.

7.5.2. Predicting the number of packets

Take as a simple example, a cluster containing 3 nodes. After synchronization between nodes, Node B (RFD) is considered to have Node A as its preferred parent (FFD).

Node A begins to calculate the probability of having 1 single packet based on the generated packet average number between the pair of nodes during the last 10 slotframes until a maximum probability value is found (Fig. 4). With the objective of saving computational resources and based on the probability mass function of a Poisson process, the algorithm stops the computation when it reaches this maximum probability.



$$Pr(E(10)=1)=\frac{(\lambda(\Upsilon)T)^1}{1!}e^{-\lambda(\Upsilon)T}$$
$$Pr(E(10)=2)=\frac{(\lambda(\Upsilon)T)^2}{2!}e^{-\lambda(\Upsilon)T}$$
$$Pr(E(10)=3)=\frac{(\lambda(\Upsilon)T)^3}{3!}e^{-\lambda(\Upsilon)T}$$
$$\vdots$$
$$Pr(E(10)=i)=\frac{(\lambda(\Upsilon)T)^i}{i!}e^{-\lambda(\Upsilon)T}$$

Fig. 4: A Poisson prediction model

7.5.3. Add/delete cells

Once the prediction of the next number of packets, which will be generated between the child node and its preferred parent in a cluster, is calculated and based on the number of cells already allocated, a node can add or remove or even keep the same number of allocated cells. These results are sent to the 6top sublayer which will be responsible for sending the following requests.

- 6p–addCell(Max): If the number of cells already allocated for the pair of nodes is lower than the prediction calculated by the algorithm, the request takes as a parameter the number of cells (Max) that will be added during the next slotframe.
- 6p–removeCell(Max): If the number of cells already allocated for the pair of nodes is greater than the prediction calculated by the algorithm, the request takes as a parameter the number of cells (Max) that will be deleted during the next slotframe.
- Do not send anything if Max is equal to the number of cells already allocated.

8. Performance evaluation

We run our simulations on OpenWSN. It is an open-source simulator for WSNs, which supports the IoT-based protocol stack with support for IEEE802.15.4 TSCH, 6LoWPAN, RPL, and CoAP. To evaluate our proposed solution, we have developed several simulation scenarios under different conditions. We have simulated a number of nodes that varies between 2 and 100 using a cluster tree topology. The packet delivery rate between each pair of nodes (PDRs) has been set at 100%. Each node generates a random number of packets during each slotframe. In the 802.15.4 physical layer configuration, we have considered that all communication channels are available and have the same physical characteristics. Table 1 shows simulation parameters.

8.1. Error rate

With each simulation, we increased the number of nodes by 10 distributed equitably between the different clusters. At each increase, we nested the responses from the 6p transactions, which contain an 8-bit sub-register containing an error code (defined in the 6top protocol). We have taken into account all types of errors that a 6p transaction can return. Then, we calculated the average of these error transactions according to the number of nodes.

Table 1: Simulation parameters

Parameter	Value
Number of nodes	2-100
Available channels	11-26
Slotframe length	101 timeslots
Time slot duration	15 ms
Data packet size	127 bytes
Topology	Cluster tree
MAC layer	IEEE 802.15.4-TSCH
Type of cells	softcells
Maximum MAC attempts	4
Maximum length of the queue	5
Transmission period	200 ms
Routing protocol	RPL
Routing metric	ETX

From Fig. 5, we show the negotiation error rate increases with increasing network density, in case we implemented MSF, the error rate increases dramatically from 1.9% up to 14.3%, whereas by implementing OMSF the rate increases from 1.2% up to 3.3%. The proposed mechanism, OMSF, largely outperforms MSF and maintains a negotiation error rate of less than 3.3% for all network densities. This is attributable to the replacement of the mechanism for measuring bandwidth, necessary for communication between a pair of nodes by the prediction calculation performed independently in each node at each cluster and which is implemented in OMSF. Otherwise, OMSF helps reduce the number of control packets exchanged over the network, allowing the node to send specific transactions of adding or removing cells.

8.2. Energy savings

A limited network duty cycle ratio (DCR) can be given by the OMSF algorithm, defined as the proportion of timeslots in which a node is operating.

We observe that a very low duty cycle ratio (Fig. 6) is reached, often below 2%, even with 100 nodes. Clearly, when there is more traffic on the network for forwarding, the ratio of active time slots increases. This is related to the role of the distributed method to complete local data processing and local prediction. Within any cluster, the cluster head gathers data. Then, on data distribution, the cluster head should finish the local prediction. Predictions must be executed by the members of each cluster and predicted data must then be transmitted to the head of the cluster. Therefore, each cluster head has a true vision of all sensor data, via the cluster. Thus, the energy consumption significantly decreased.

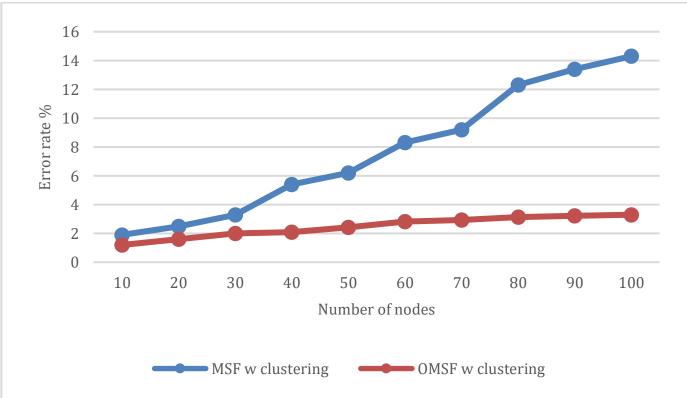


Fig. 5: 6p error rate

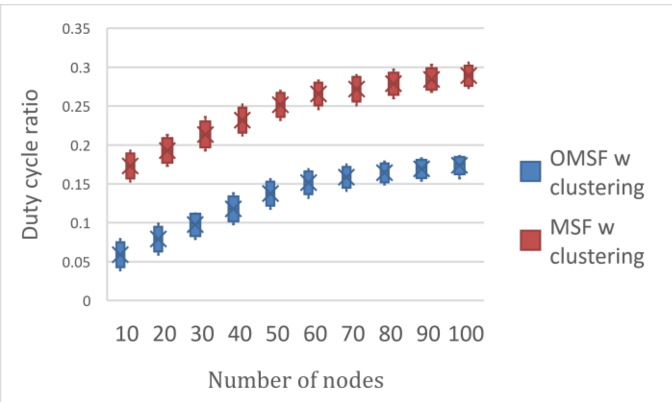


Fig. 6: Duty cycle

8.3. Traffic load

In the following simulation, we calculated the number of control and measure packets that a pair of nodes transmits in order to decide the number of required cells in the next slotframe. We have gradually increased the number of nodes in the network from 10 to 100 deployed in a cluster tree topology, in intervals of 10. The MSF protocol implements the bandwidth estimation algorithm to translate the needs of the nodes into a number of cells. The latter monitors the amount of data sent between each pair of nodes. When this quantity becomes large or small (reaches a determined threshold) compared to the number of cells already allocated, MSF asks 6top to add or delete cells with the designated node. This process generates the transmission of certain control packets, which leads to an additional traffic load.

Fig. 7 displays the overhead load of traffic (measured in bytes) used by nodes to share network information. We note that the number of messages exchanged increases with the number of nodes deployed. This is attributed to the negotiation phases made between nodes to decide which 6p transaction to deploy in scheduling. We also find that the OMSF protocol retains a nearly constant number of transmitted packets. This is because of predicting the number of cells needed during the next slotframe for each pair of nodes at each cluster. It eliminates the transmission of transactions exchanged between a pair of nodes in the MSF protocol to determine the necessary bandwidth. The OMSF protocol prevents overloads from being sent and maintains a constant average over the network life of control packets.

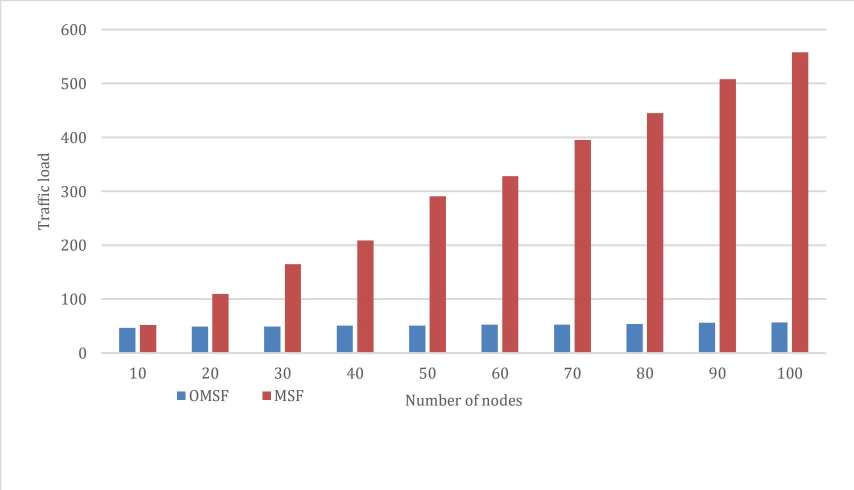
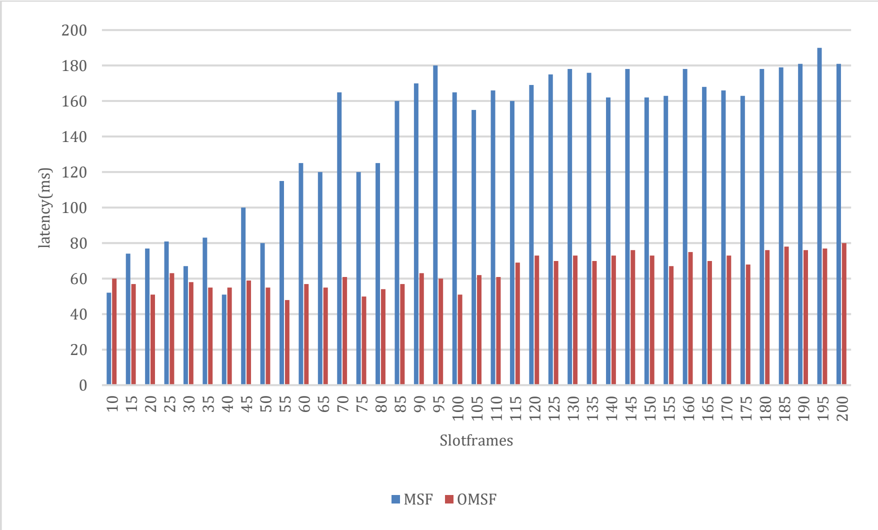


Fig. 7: Traffic load

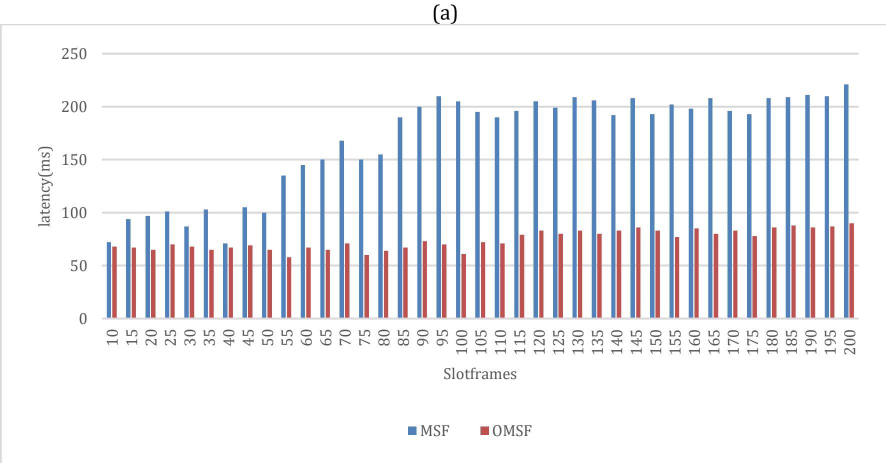
8.4. Latency

In order to study the end-to-end latency of our OMSF proposal, we simulated a network consisting of 50 nodes distributed over 5 clusters. Each node in the network generates, during the first slotframes, constant data traffic equal to 2 packets for each slotframe. Then a transient data flow varying from 2 to up to 8 packets per slotframe is created by each node in each cluster. We simulated the same network under the same conditions by implementing the MSF protocol in order to trust OMSF in terms of latency.

Fig. 8a presents a comparison between MSF and OMSF in terms of latency, where the source node is chosen from a level 1 cluster which is the nearest to the Destination Acyclic Graph (DAG) root. Fig. 8b presents also the same comparison, but the source is chosen from a level 4 cluster. We calculated the latency as follows: Each packet is time stamped from the time it was generated at the source node until it hits the DAG root node. When a packet reaches 4 retransmission attempts and the node's queue memory becomes full, the packet will be significantly discarded. However, if a packet is transmitted on several occasions, an increase in the latency time may occur. Each attempt to retransmit a packet generates an increase in latency since each packet takes longer than its allotted time to reach its destination. The latency was almost constant during the first slotframes due to the stable data traffic flow that has been generated by the nodes of the network. Thereafter, the latency varies between cycles due to failed exchanges. In Figs. 8a and 8b, we note that the OMSF protocol maintains an end-to-end latency of less than that achieved by the protocol MSF. This is due to the reduction in negotiation errors as well as collisions, which the scheduling algorithm of the OMSF protocol guarantees based on the prediction calculation. We also note that for the two functions the latency is less when the source belongs to the cluster closest to the DAG root. This is due to greater variability in the neighborhoods of the clusters, especially at the lowest tree levels. OMSF protocol maintains an end-to-end latency of fewer than 80 milliseconds during all slotframes in the case of level1 cluster and end-to-end latency of fewer than 90 milliseconds in level 4 cluster. So, it always presents better results even by changing the level of the cluster. Therefore, the added traffic load is also reduced, causing a high PDR value when irregular data streams are generated.



(a)



(b)

Fig. 8: Latency, (a) level 1 cluster, (b) level 2 cluster

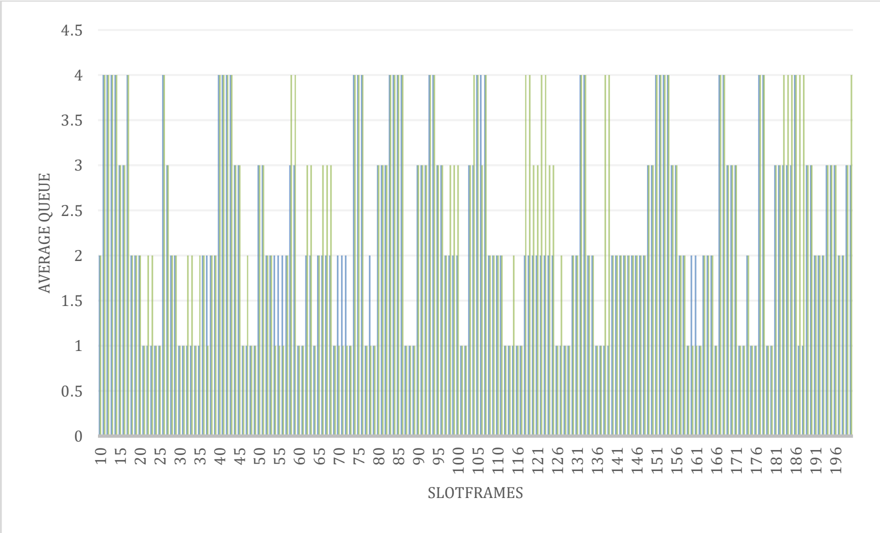
8.5. Queue size

The traffic flow in a TSCH 802.15.4e network varies depending on the location of the cluster, to which the node belongs, in the DODAG. Nodes belonging to a cluster near the root have a higher traffic load than nodes in end clusters. Following the detection of a sudden event by a group of nodes, a high traffic load will be transported in the network. When this traffic reaches nodes in the cluster near the root, there will be an accumulation of packets in their memory queues. As a result, each of these nodes will need multiple cells in a slotframe to dump packets from its queue. This will cause unwanted delays, negatively affecting the cumulative communication delay. When the client requires to provide vital data such as alarms, this delay is extremely inappropriate. In order to study the impact of the prediction algorithm proposed in the OMSF protocol on the size of the memory queue, we simulated a network of 100 nodes distributed over 5 clusters and we compared the result obtained by implementing the protocol MSF. The memory size of each node can hold a maximum of 5 packets in the queue. Once this memory is full, the first arrived packet in memory will be discarded first. During each slotframe, we determined the queued packets' average number of all nodes in the network.

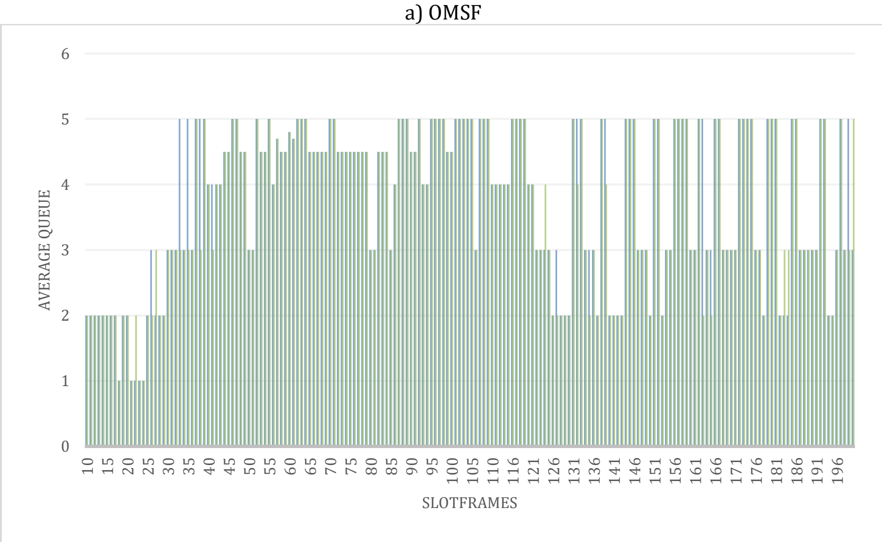
Fig. 9 shows the queued packets average number in the network. In Fig. 9, it seems to be that, in the case of MSF protocol Fig.9b in some slotframes, the queues in the memories of the nodes are almost complete and reach their limits, contributing to the missing packets. The OMSF protocol exhibits better performance Fig.9a by keeping a virtually moderate queue, showing an average of 0 to 4 packets in the queue during all slotframes. In particular, the nodes which are located in clusters near the DAG root node, over-allocate a number of cells greater than those which are located in clusters distant to it. The high packet flow through these nodes allows the prediction algorithm, proposed in the OMSF protocol, to reserve more cells. Therefore, this ensures sufficient bandwidth, which meets the fluctuating needs of the pairs of nodes.

9. Conclusion

In this work, we have formulated the packet transmission mode in the TSCH network based on the clustering and prediction model. A probability calculation is made between each parent node and its child using Poisson model, we were able to predict the number of packets that will be exchanged between this pair of nodes. This prediction allowed us to deduce the number of cells needed in the next slotframe. As a result, the traffic load circulating in the network, used to determine the scheduling, has been reduced considerably. In addition, data clustering and prediction methods can reduce the energy consumption of sensor nodes for data collection.



a) OMSF



b) MSF

Fig. 9: Queue size

In a future perspective, we intend to integrate into our model a cell selection method, which gives priority to the nodes closest to the root node. This will provide more bandwidth to the nodes that have more traffic, thus the latency will be further reduced.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Accettura N, Palattella MR, Boggia G, Grieco LA, and Dohler M (2013). Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things. In the IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks," IEEE, Madrid, Spain: 1-6. https://doi.org/10.1109/WoWMoM.2013.6583485

Accettura N, Palattella MR, Dohler M, Grieco LA, and Boggia G (2012). Standardized power-efficient and internet-enabled communication stack for capillary M2M networks. In the IEEE Wireless Communications and Networking Conference Workshops, IEEE, Paris, France: 226-231. https://doi.org/10.1109/WCNCW.2012.6215495

Accettura N, Vogli E, Palattella MR, Grieco LA, Boggia G, and Dohler M (2015). Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation. IEEE Internet of Things Journal, 2(6): 455-470. https://doi.org/10.1109/JIOT.2015.2476915

Chang T, Vučinić M, Guillén V, Dujovne D, and Watteyne T (2020). 6TiSCH minimal scheduling function: Performance evaluation. Internet Technology Letters, 3(4): e170. https://doi.org/10.1002/itl2.170

Chang T, Watteyne T, Pister K, and Wang Q (2015). Adaptive synchronization in multi-hop TSCH networks. Computer Networks, 76: 165-176. https://doi.org/10.1016/j.comnet.2014.11.003

Choudhury N, Matam R, Mukherjee M, and Lloret J (2020). A performance-to-cost analysis of IEEE 802.15. 4 MAC with 802.15. 4e MAC modes. IEEE Access, 8: 41936-41950. https://doi.org/10.1109/ACCESS.2020.2976654

Demir AK and Bilgili S (2019). DIVA: A distributed divergecast scheduling algorithm for IEEE 802.15. 4e TSCH Networks. Wireless Networks, 25(2): 625-635. https://doi.org/10.1109/ISNCC.2019.8909093

Ergen SC and Varaiya P (2010). TDMA scheduling algorithms for wireless sensor networks. Wireless Networks, 16(4): 985-997. https://doi.org/10.1007/s11276-009-0183-0

Farrel A, Vasseur JP, and Ash J (2006). A path computation element (PCE)-based architecture. No. rfc4655, Network Working Group, Washington, USA. https://doi.org/10.17487/rfc4655

Jin Y, Kulkarni P, Wilcox J, and Sooriyabandara M (2016). A centralized scheduling algorithm for IEEE 802.15. 4e TSCH based industrial low power wireless networks. In the IEEE Wireless Communications and Networking Conference, IEEE, Doha, Qatar: 1-6. https://doi.org/10.1109/WCNC.2016.7565002

Kurunathan H, Severino R, Koubaa A, and Tovar E (2018). IEEE 802.15. 4e in a nutshell: Survey and performance evaluation. IEEE Communications Surveys and Tutorials. 20(3): 1989-2010. https://doi.org/10.1109/COMST.2018.2800898

Mohamadi M and Senouci MR (2018). Scheduling algorithms for IEEE 802.15. 4 TSCH networks: A survey. In the International Conference on Computer Science and its Applications, Springer, Kuala Lumpur, Malaysia: 50: 4-13. https://doi.org/10.1007/978-3-319-98352-3_2

Mohamadi M, Djamaa B, and Senouci MR (2020). Industrial internet of things over IEEE 802.15. 4 TSCH networks: Design and challenges. International Journal of Internet Technology and Secured Transactions, 10(1-2): 61-80. https://doi.org/10.1504/IJITST.2020.10026294

Ojo M and Giordano S (2016). An efficient centralized scheduling algorithm in IEEE 802.15. 4e TSCH networks. In the IEEE Conference on Standards for Communications and Networking, IEEE. Berlin, Germany, 1-6. https://doi.org/10.1109/CSCN.2016.7785164 **PMid:26708953**

Ojo M, Giordano S, Portaluri G, Adami D, and Pagano M (2017). An energy efficient centralized scheduling scheme in TSCH networks. In the IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, Paris, France: 570-575. https://doi.org/10.1109/ICCW.2017.7962719

Palattella MR, Accettura N, Dohler M, Grieco LA, and Boggia G (2012a). Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks. In the IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications-(PIMRC), IEEE, Sydney, Australia: 327-332. https://doi.org/10.1109/PIMRC.2012.6362805

Palattella MR, Accettura N, Vilajosana X, Watteyne T, Grieco LA, Boggia G, and Dohler M (2012b). Standardized protocol stack for the internet of (important) things. IEEE Communications Surveys and Tutorials, 15(3): 1389-1406. https://doi.org/10.1109/SURV.2012.111412.00158

Palattella MR, Watteyne T, Wang Q, Muraoka K, Accettura N, Dujovne D, and Engel T (2015). On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks. IEEE Sensors Journal, 16(2): 550-560. https://doi.org/10.1109/JSEN.2015.2480886

Papadopoulos GZ, Matsui T, Thubert P, Texier G, Watteyne T, and Montavont N (2017). Leapfrog collaboration: Toward determinism and predictability in industrial-IoT applications. In the IEEE International Conference on Communications, IEEE, Paris, France: 1-6. https://doi.org/10.1109/ICC.2017.7997160 **PMid:27184731 PMCid:PMC7121526**

Paul W and Baschnagel J (2013). Stochastic processes: From physics to finance. 2nd Edition, springer, Berlin, Germany. https://doi.org/10.1007/978-3-319-00327-6

Soua R, Livolant E, and Minet P (2013). MUSIKA: A multichannel multi-sink data gathering algorithm in wireless sensor networks. In the 9th International Wireless Communications and Mobile Computing Conference, IEEE, Sardinia, Italy: 1370-1375. https://doi.org/10.1109/IWCMC.2013.6583756

Soua R, Minet P, and Livolant E (2012). MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks. In the IEEE 31st International Performance Computing and Communications Conference, IEEE, Austin, USA: 91-100. https://doi.org/10.1109/PCCC.2012.6407742

Soua R, Minet P, and Livolant E (2014). A distributed joint channel and slot assignment for convergecast in wireless sensor networks. In the 6th International Conference on New Technologies, Mobility and Security, IEEE, Dubai, UAE: 1-5. https://doi.org/10.1109/NTMS.2014.6813995

Soua R, Minet P, and Livolant E (2015). DiSCA: A distributed scheduling for convergecast in multichannel wireless sensor networks. In the IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, Ottawa, Canada: 156-164. https://doi.org/10.1109/INM.2015.7140288

Soua R, Minet P, and Livolant E (2016). Wave: A distributed scheduling algorithm for convergecast in IEEE 802.15. 4e TSCH networks. Transactions on Emerging Telecommunications Technologies, 27(4): 557-575. https://doi.org/10.1002/ett.2991

Tinka A, Watteyne T, and Pister K (2010). A decentralized scheduling algorithm for time synchronized channel hopping. In the International Conference on Ad Hoc Networks, Springer, Victoria, Canada: 201-216. https://doi.org/10.1007/978-3-642-17994-5_14

Vallejo M, Recas J, Del Valle PG, and Ayala JL (2013). Accurate human tissue characterization for energy-efficient wireless on-body communications. Sensors, 13(6): 7546-7569. https://doi.org/10.3390/s130607546 **PMid:23752565 PMCid:PMC3715232**

Verma PK, Verma R, Prakash A, Agrawal A, Naik K, Tripathi R, and Abogharaf A (2016). Machine-to-Machine (M2M) communications: A survey. Journal of Network and Computer Applications, 66: 83-105. https://doi.org/10.1016/j.jnca.2016.02.016

Zand P, Chatterjea S, Das K, and Havinga P (2012). Wireless industrial monitoring and control networks: The journey so far and the road ahead. Journal of Sensor and Actuator Networks, 1(2): 123-152. https://doi.org/10.3390/jsan1020123