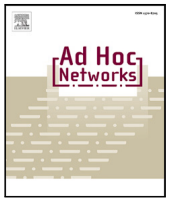
Ad Hoc Networks 152 (2024) 103268

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Survey paper

# Dy-MAC: Implementation of dynamic MAC stack for IEEE 802.15.4e-TSCH

Deeksha P. Rao *, Shrutkirthi S. Godkhindi, Abhigna V. Kumar, T.V. Prabhakar

*Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore, India*

ARTICLE INFO

*Keywords:*
Frequency hopping
IEEE 802.15.4e
IIoT
TSCH
Time-synchronization
Channel hopping
Node re-joining
SPI
Information element (IE)
Finite state machine (FSM)
API

ABSTRACT

This paper presents the lessons learnt in the implementation of IEEE 802.15.4e-Time Slotted Channel Hopping (TSCH) Medium Access Control (MAC) protocol on commercially available hardware. Our MAC Application programming Interface (API) called the Dy-MAC is agnostic to the PHY layer and is coded on a ARM cortex M4. MAC functions such as TDMA, FDMA, frame counter, channel access, synchronization and PAN formation are abstracted as part of this API. Dy-MAC is demonstrated for 2.4 GHz with O-QPSK modulation and 868 MHz with SUN FSK modulation. Furthermore, the API supports star, mesh and tree network to provide flexible topologies in factory environment. Our experiments show that for a 76 byte payload, the TSCH Dy-MAC provides ~99% packet delivery ratio (PDR) for 26 ms timeslot with SUN FSK PHY and ~97% PDR for 10 ms timeslot with O-QPSK PHY. To maintain synchronization over the entire network, we implemented algorithms to successfully communicate between fast and slow nodes. Resilience to interference and range with packet delivery ratio is evaluated. We consolidate our implementation in terms of 10 lessons that were learnt in building the Dy-MAC protocol.

## 1. Introduction

In Industry 4.0, there is an unprecedented ubiquity of interconnected smart devices, which is predicted to grow exponentially in the near future. Hence, to support such gigantic and time critical network applications, an efficient, scalable, noise resistant, reliable and low power communication protocol is required.

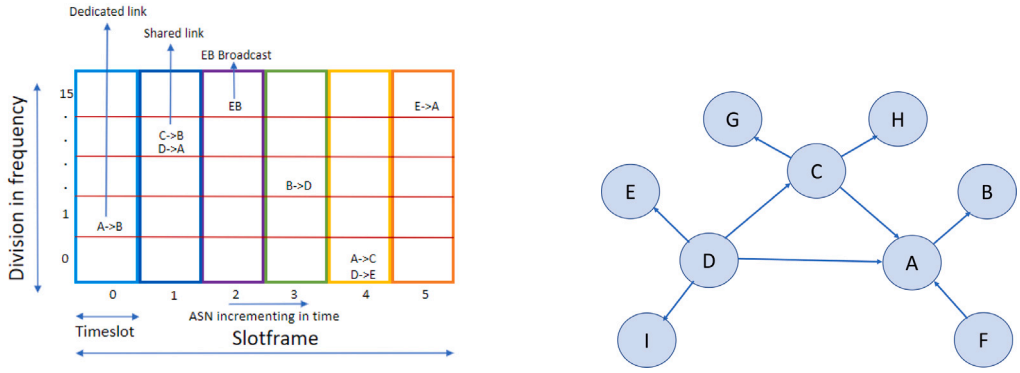
The IEEE 802.15.4 wireless standard [1] based radio systems are widely applied in commercial and industrial environments owing to their noise resistant features. For instance, the radio supports Offset Quadrature Phase Shift Keying (O-QPSK) modulation with Direct Sequence Spread Spectrum (DSSS). A few commercial applications include medical fields, environment monitoring, bio-diversity mapping, etc. The industrial applications include monitoring production lines, smart grids, real time condition monitoring and several other critical applications. The IEEE 802.15.4e [2] is an evolving protocol with various Medium Access Control (MAC) modes and provides noise resistance, power efficiency and deterministic behavior making it the future protocol for the IoT eco-system.

The IEEE 802.15.4e standard provides different MAC modes suitable for various applications. Time Slotted Channel hopping (TSCH) mode facilitates multi-hop multi-channel network, supporting various topology. The provision for channel hopping allows the nodes to communicate over a range of frequencies imparting resistance towards channel interference and fading. The Time Division Multiple Access (TDMA) fortifies the unrestrained delay and provides communication reliability. Hence, the integration of TDMA and channel hopping imparts the deterministic characteristic to the TSCH mode. In our previous work [3] we have described the six lessons learnt in hardware implementation of the MAC layer for the IEEE 802.15.4e-TSCH mode. The MAC layer was developed on the ARM Cortex-M4, the nRF52840 [4] micro-controller platform and the PHY layer functionality of the ADF7242 [5] radio transceiver chipset was employed.

In this current extended version, perhaps for the first time ever, we have implemented IEEE 802.15.4e-TSCH for three network topologies viz., Star, Mesh and Tree to support short and long range communication. We explored implementation of TSCH Dynamic MAC (Dy-MAC) stack for different PHY layer. A consolidated module of lessons learnt while implementing and deploying TSCH Dy-MAC with Smart Utility Networks Frequency Shift Keying (SUN FSK) PHY and O-QPSK PHY is conferred. The remainder of the paper is as follows. First, the design of TSCH and key features of Time Slotting and Channel hopping is discussed in Section 2. Then, Section 3 provides an overview of performance of IEEE 802.15.4e-TSCH for Industrial Internet of Things (IIoT) applications and other related works. In Section 4, we discuss our design and implementation of TSCH Dy-MAC with O-QPSK and SUN FSK PHY. Experimental settings and evaluation of TSCH Dy-MAC is presented in Section 5 and Section 6 provides the conclusions.

* Corresponding author.
*E-mail addresses:* deekshaprao24@gmail.com, deeksharao@iisc.ac.in (D.P. Rao).

https://doi.org/10.1016/j.adhoc.2023.103268
Received 15 May 2023; Accepted 2 August 2023
Available online 7 September 2023




(a) Illustration of Time and Frequency division (b) TSCH Topology

**Fig. 1.** Topologies.

**Table 1**
MAC PIB attributes for *macSlotframeTable*.

| Term | Description as per standard |
|---|---|
| *macSlotframeHandle* | Identifier for a slotframe |
| *macSlotframeAdvertise* | Determines if the slotframe has to be advertised in the EB |
| *macSlotframeSize* | Number of timeslots in the Slotframe |

**Table 2**
MAC PIB attributes for *macLinkTable*.

| Term | Description as per standard |
|---|---|
| *macLinkHandle* | Identifier for a link |
| *macTxType* | Tx Link |
| *macRxType* | Rx Link |
| *macSharedType* | Shared Link |
| *macLinkTimekeeping* | Link with time keeping parent node |
| *macPriorityType* | Priority Link |
| *macLinkType* | Advertise link or normal link |
| *macSlotframeHandle* | Identifier of the slotframe to which the link belongs |
| *macNodeAddressMode* | Addressing mode used. |
| *macNodeAddress* | Address of the neighbor node |
| *macTimeslot* | Timeslot of the Link |
| *macChannelOffset* | Channel Offset |
| *macLinkAdvertise* | Advertise the link |

## 2. Background

The IEEE 802.15.4e-TSCH mode shown in Fig. 1(a) is characterized by time slotting and channel hopping. The horizontal axis represents the division in time and the vertical axis depicts the division in frequency. The time slotting mechanism (TDMA) has the timeslot as its integral part which when combined together forms a slotframe. This slotframe structure repeats periodically. Each timeslot is uniquely identified by an Absolute Slot Number (ASN). The ASN is a 5-byte number that is initialized by the Personal Area Network (PAN) coordinator during the network initiation. For every timeslot the ASN increments by 1, which helps in maintaining a track of number of timeslots since the start of network. As depicted in Fig. 1(a), each timeslot can have a dedicated link (or dedicated channel) or a shared link (or shared channel) between two nodes to communicate. To enable nodes to join the network, a few links (or channels) are used to send out a special packet called the Enhanced Beacon (EB). The EB contains the Information Elements (IEs) related to network parameters.

To determine the channel hopping frequency, we have implemented the hopping Eq. (1), which provides the frequency of operation based on the ASN. This frequency is set in the radio before the node enters either Tx or Rx state.

$$f = F[(ASN + channel\_offset)\ \%\ number\_of\_channels] \quad (1)$$

The function $F$ is implemented as a lookup table that contains a list of frequencies. The operating frequency for a particular timeslot is found by taking modulus of ASN added with channel offset and number of channels. PAN coordinator assigns different channel offset to allocate different channels for a timeslot. This channel hopping sequence is shared with the node as Channel Hopping IE through EB during node joining. The Clear Channel Assessment (CCA) is optional in TSCH mode, enabling CCA will improve the communication reliability in the presence of interference.

### 2.1. MAC PIB attributes

Section 6.2.6.1 of the standard [2] provides the general structure of the slotframe. Section 8.4.3.3.2 of the standard [2] defines MAC PAN Information Base (PIB) Attributes for *macSlotframeTable*, which contains description of the slotframe structure as shown in Table 1. A node can contain multiple slotframes, where each slotframe is identified by a unique slotframe handle. In our implementation, we have demonstrated a single slotframe. Multiple slotframes can be created by calling instances of the defined structure. These slotframes can have soft cells, which are editable by upper layer and hard cells which are fixed and can only be read by upper layer.

Each link in a slotframe is described by MAC PIB for *macLinkTable* as per Section 8.4.3.3.3 of the standard [2]. The attributes for *macLinkTable* are as described in Table 2. The MAC PIB attributes supported for Timeslot structure follow Section 6.5.4.2 of the standard. The terms used in the standard timeslot structure are as explained in Table 3. The attributes shown in the table are transmitted as part of the TSCH timeslot IE. The values of the MAC PIB attributes for the defined timeslot template ID are listed in Section 5.3.

**Topologies supported by TSCH:** The TSCH Dy-MAC implementation supports various topologies. A node can be a Full-Function Device (FFD) or a Reduced-Function Device (RFD). An FFD is capable of acting as a coordinator or a PAN coordinator which can act as Tx node and a Rx node in a network. RFDs are the end devices which can either do sense and send (RFD-Tx) or receive and actuate (RFD-Rx). These are the end devices with very minimal activity. The TSCH network supports both star as well as peer-to-peer topology. In a basic star topology an FFD (PAN coordinator) establishes its own network with other RFD. While peer-to-peer involves many FFD and RFD nodes supporting tree, mesh and partial mesh network as shown in Fig. 1(b).

**Table 3**
Standard timeslot terminologies associated with MAC PIB timeslot template ID.

| Term | Description as per standard |
|---|---|
| *macTsCcaOffset* | Waiting time till node can enter CCA state. |
| *macTsCca* | Duration for CCA. |
| *macTsRxTx* | Time to transit to TX after CCA. |
| *macTsMaxTx* | Time required to send a maximum length packet. |
| *macTsRxAckDelay* | Waiting time to start RX state for Ack reception. |
| *macTsAckWait* | Time between end of packet transmission and start of Ack reception. |
| *macTsRxOffset* | Minimum waiting time for a node before entering RX state. |
| *macTsRxWait* | Minimum waiting time for Packet. |
| *macTsTxAckDelay* | Time between end of packet reception and start of Ack transmission. |
| *macTsMaxAck* | Time to transmit a Maximum length Ack. |
| *macTsTimeslotLength* | Total length of a timeslot. |

**Table 4**
PHY layer parameter for different Transceiver.

| Range | O-QPSK PHY | SUN FSK PHY |
|---|---|---|
| Frequency Band | 2.4 GHz | 868 MHz |
| Modulation | O-QPSK | SUN FSK |
| Data Rate | 250 kbps | 50 kbps |
| No. of channels | 1–16 | 1-69 |
| Channel Spacing | 5 MHz | 0.1 MHz |
| Radio Used | ADF7242 | ADF7030-1 |

## 3. Related works

The authors of [6] have surveyed and outlined the major enhancement made to the MAC layer in 802.15.4e from 802.15.4 standard. Given that the IEEE 802.15.4e-TSCH communication protocol provides deterministic delay guarantees, it is ideally suited for IIoT systems. There have been several simulation studies and a few hardware implementations as well. From the work reported in this paper, one can characterize the latency from application layer to the physical layer. Data transferred to the radio chip via the Serial Peripheral Interface (SPI) protocol and the associated SPI clock frequency is available as a handle for users. Several works are as discussed further in this section. The works in [7] provide the performance evaluation of the IEEE 802.15.4e-TSCH standard based on the number of dedicated links and number of nodes in the network. The authors evaluate the network using Matlab simulations. With the increase in number of dedicated channels the reliability, throughput and energy consumption of TSCH network increases. Whereas, in our implementation we have set aside a few shared links within the slotframe structure. The Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol implementation for these links is pending an implementation. [8] demonstrates the existence and reliability of the TSCH network in a noisy environment such as an aircraft cabin. Here three Wi-Fi access points operating in three different channels, each provide services to about 19 to 20 clients. We conduct Wi-Fi interference measurements in a laboratory setting and enumerate our results from practice. Simulations of the standard have been performed on many open-source platforms like GNU radio companion [9], Open WSN [10], Contiki [11], etc. They provide detailed view on the development and performance of the IEEE 802.15.4e-TSCH MAC stack.

[12] provides the experimental performance analysis for Smart Utility Networks (SUN) for outdoor environment. The paper [13] provides a comparative study of different SUN modulation schemes tested for 99 days using 11 nodes. It was inferred that the highest PDR of 81.3% was given by SUN-FSK when compared to other SUN modulation schemes. In the paper [14], the performance IEEE802.15.4g was observed for different environmental scenarios such as Line of Sight (LoS), Smart Agriculture, Urban Canyon and Advanced Metering Infrastructure (AMI). The experimental result shows that FSK-FEC modulation scheme provides the longest radio link when compared to other SUN modulation schemes. Since the standard provides flexibility of implementing the scheduling algorithm and node joining process, large number of works have been proposed in these areas. A scheduling algorithm called Traffic Awareness Scheduling Algorithm (TASA) [15], is a centralized scheduling method based on the graph theory. The Decentralized version of TASA is called DeTAS [16]. Paper [17] provides a distributed scheduling algorithm, aiming to minimize the difference between the number of packets to be sent and number of scheduled cell. Since scheduling is a requirement for the PAN coordinator or some node equivalent of this functionality, it is strictly not in the purview of the standard. In our work, we have hand crafted the slotframe structure, timeslot, time schedule for shared channels, Enhanced Beacon (EB) broadcast slot. A node that wishes to join the network can pick the information available in the EB and automatically join the network. The work in [18] provides a node joining algorithm where the EBs are advertised in different channels to ensure faster joining. In [19], author demonstrates multi-PHY layer implementation within single TSCH network by modifying the timeslot structure. In order to increase the success rate of EB reception even by the nodes which are far from the PAN Coordinator, the author in [20] proposes Sparse Beacon Advertisement scheme. In this approach the nodes that has joined the network will re-transmit the Beacon, which helps in faster node association. Our implementation uses factory programmed single channel for EB reception and node joining.

### 3.1. Contributions of the paper

To the best of authors' knowledge this is the first ever implementation of IEEE 802.15.4e-TSCH protocol using a commercially available hardware. The major contribution of this work is as follows:

- A set of API functions adhering to IEEE 802.15.4e-TSCH is designed and tested. These functions are mapped to the MAC layer functionality, which are generic to the PHY layer. These APIs are employed to demonstrate IEEE 802.15.4e-TSCH with two different PHYs.
- A time synchronization algorithm as proposed in the standard is implemented to ensure synchronization between nodes extending to a multi-hop network.
- A node rejoining algorithm is designed and implemented to ensure a faster re-connection of the node.
- A concise module of lessons learnt for implementing and deploying TSCH Dy-MAC is provided.
- We present most favored set of features required in TSCH FSM which addresses deterministic transmit opportunity as well as support for energy efficient IEEE 802.15.4e-TSCH radio.

### 3.2. Motivation for TSCH Dy-MAC

In the 2015 revision of the standard, a total of six PHY and one MAC amendment was defined. List of several PHY defined in [2] includes O-QPSK and SUN FSK PHY. Since our goal is to design and implement a Dynamic MAC for short and long range communication, we employed two transceiver radios with O-QPSK PHY and SUN FSK PHY. The parameters of these radios is given in Table 4. For short range, our

choice is the popular Direct Spread Spectrum based O-QPSK modulated communication. Whereas for long range, we chose the SUN FSK for its robustness and ability to offer multiple over the air data rates.

## 4. Implementation

This section explains the algorithms and techniques followed to develop the TSCH Dy-MAC from scratch, as defined in the IEEE 802.15.4e-TSCH standard. The Dy-MAC stack is developed on ARM cortex M4 (nRF52840) micro-controller platform. The various functionalities implemented are available as API as explained in the following subsections. We provide a detailed discussion on PAN formation. To test the MAC functionality, the controller is interfaced with commercially available radios which provide base-band and PHY stack functionalities. The Dy-MAC is designed to serve for both short range (O-QPSK based ADF7242 transceiver) and long range (SUN FSK based ADF7030-1 transceiver) communication. A brief description of these radios are defined in Section 4.4.

### 4.1. Lessons from time synchronization

For any micro-controller to operate, it ought to have a clock source. Generally, a crystal clock oscillator is the preferred choice. The crystal's accuracy is measured in parts per million (ppm). The nRF52840 works on a crystal that has an accuracy of ±40 ppm, implying that given two nRFs with 64 MHz clock frequency, the frequency deviation between them will be ±256 Hz i.e. their frequency will vary in the range of 63.999744 MHz to 64.000256 MHz. Since the oscillator circuit is implemented within the controller, due to temperature variations, clock drifts result in frequency variation.

To maintain synchronization among the nodes in a TSCH network, the standard suggests two synchronization methods viz., Frame based synchronization and Ack based synchronization. In Frame based synchronization the receiving node will correct its clock based on the packet arrival time, whereas in Ack based synchronization the time correction information is sent as a Time Correction IE to the transmitter node to correct its clock. Although, the standard suggests the two-time synchronization algorithms, it does not specify the application scenario for the two methods. The ***first lesson*** we learnt is that, the nodes in a TSCH network requires both Ack based and Frame based synchronization because in the IEEE 802.15.4e-TSCH network, there can be multiple transceivers with faster and slower clocks. For example, consider a network where node A is a parent of node B and C. It is now possible that node B invokes frame based synchronization with A, whereas node C invokes Ack based synchronization with A. Therefore, every node should be equipped with both the algorithms.

Irrespective of the topology, we have employed a parent–child synchronization algorithm, where a parent's clock is assumed to be precise and stable. It is reasonable to assume that parent is equipped with a GPS module [21]. In a multi-level network such as tree structure, a child may serve as a parent to the lower levels and in such scenarios, the synchronization shall start from a precise and stable parent. When child is a Rx node, it performs Frame based synchronization and if the child is a Tx node, it performs Ack based synchronization. For example, from Fig. 1(b) node D will be equipped with the GPS module. Node A shall perform Frame based synchronization with node D (for D->A slot). Further, node F to node A is Ack based synchronization, and node B to node A supports Frame based synchronization. In our ***second lesson***, we learn from our implementation that the PAN coordinator should be equipped with a precise and stable clock to ensure the time propagates outwards. Further, the time keeping link should always be observed from child to parent.

In order to achieve Frame based and Ack based synchronization, the child has to evaluate time correction offset and compensate this offset with a time source neighbor. It is possible that a time source neighbor is its own parent. Prior to a child synchronizing, the parent would have synchronized to a precise and stable source. The parameter *macLinkTimeKeeping* as shown in Table 2, assists in identifying the link on which a time source neighbor is accessible. While there is no explicit timeslot for synchronization in the slotframe structure, during a link with time source neighbor (here parent) the time correction offset is calculated. The time correction offset is the difference in expected arrival time and actual arrival time. The expected arrival time is calculated using the MAC PIB Attributes and the actual arrival time is calculated when the Start Frame Delimiter (SFD) interrupt is received from the radio. While the child is an Rx node, the offset is calculated at child itself (Frame based) and when the child is a Tx node, the time offset is evaluated at parent and sent to child as Time Correction IE via Ack (Ack based).

A positive value of offset indicates that the child clock is slower, while a negative value indicates child clock is faster than the parent. If the child detects that it is slower than the parent, it initiates the timeslot early by a value equal to time correction offset to re-synchronize with the parent. If the child is faster, child delays the start of timeslot to match its parent clock. Further, for a multi-level network (tree structure), to ensure synchronization with lower levels, all connected nodes are informed about the upcoming time synchronization via Time correction IE appended to Packet or Ack. The whole synchronization process is provided in Algorithm 1.

**Algorithm 1:** Synchronization

```
begin
if macLinkTimeKeeping==1 then
    if macRxType==1 then
        Time_Correction_Offset=Expected_Arrival_Time−Actual_
         Arrival_Time
    else
        if macTxLink==1 then
            Time_Correction_Offset= Ack.Time_Correction_IE
        else
            Time_Correction_Offset= 0
        end
    end
    if |Time_Correction_Offset| > 100μs then
        Advertise to it's neighbors via Time Correction IE.
        Perform compensation.
    end
end
end
```

In our TSCH Dy-MAC implementation, the timeslot design has a provision to accommodate a clock drift of 250 μs, which is safe in margin of $\pm macTsRxWait/2$ threshold as defined in standard [2] Subsection 6.5.4.1. Since the effect of compensation is experienced in the next slotframe, within which a drift accumulation might occur, our experimental results suggest a threshold of 100 μs is appropriate. While a drift of 10 μs compensation is also possible, our drift threshold improves the battery life. Here, we present our ***third lesson*** on design consideration for timeslot period. If the clock of the chosen hardware has a lesser accuracy, due to higher clock drift the $macTsRxWait$ has to be chosen appropriately.

### 4.2. Lessons from PAN formation-node joining

While our node joining requires two slotframes, our re-joining process is limited to a single slotframe. An advertising device (PAN coordinator or coordinator), periodically transmits EBs to advertise the presence of its network. In our TSCH Dy-MAC implementation, the PAN Coordinator broadcasts one EB per slotframe restricted to a single channel. This EB includes: (a) TSCH Synchronization IE. (b) TSCH Slotframe and Link IE. (c) TSCH Timeslot IE. (d) TSCH Channel

hopping IE. These IEs contain information about network parameters such as timeslot length, channel hopping sequence, total number of channels and slotframe structure. These parameters are extracted from the first EB and stored in the Non-Volatile Memory (NVM) and a flag (namely NVM_set) is set in the memory to indicate the same. The ASN data is extracted from the TSCH Synchronization IE of the second EB and is used to synchronize with the network. This two-beacon process for node joining is required because extracting and storing of the network parameters consume some time, which is hardware dependent. We learn in ***fourth lesson*** that, in case of a node rejoining, the flag information helps to extract the IEs directly from the NVM thus reducing two step process to a single step process. Algorithm 2 outlines the steps followed by a node to join or rejoin the network.

**Algorithm 2:** Node (Re)Joining algorithm

***begin***
**if** $NVM\_set == 1$ **then**
  Retrieve the following from NVM;
    1. *macSlotframeTable*
    2. *macLinkTable*
    3. Timeslot Length
    4. Channel hopping sequence
    5. Number of channels
**else**
  Receive EB_1;
  Extract and store the required parameters from EB_1 in NVM;
  Set NVM_set flag;
**end**
Receive EB_2 by following the timeslot period;
Extract and update the ASN;
Follow the Schedule as mention in the IEs to synchronize to the network.;
***end***

### 4.3. Correctness of implementation

In order to estimate the correctness of algorithm, the probability of incorrect output is defined as $P(I, O, R)$, where $I$ is the event of successful packet transmission, $O$ is the event of Ack reception and $R$ is the set of cases where event $O$ does not occur given that event $I$ has occurred. $R$ can be represented as:

$$R = \begin{cases} \text{Case 1: Either received packet or Ack is corrupted} \\ \text{Case 2: Time for packet transmission followed by} \\ \qquad \text{Ack reception} > \text{timeslot period} \\ \text{Case 3: Loss of Synchronization} \end{cases} \tag{2}$$

Correctness is the probability of all the cases except R and is given by $(1 - P(I, O, R))$. Case 1 of R is due to bad CRC, which can be due to interference or wrong evaluation of CRC. Case 2 can occur if the implementor does not account for hardware delay such as SPI delay. Case 3 is faulty implementation of network formation and time synchronization. We characterize the packet loss occurred in our implementation into the above mentioned categories. In our experiment we found the PDR to be 98.95%. From the total 1.05% loss of correctness, 1.037% is caused due to bad CRC (Case 1) and 0.013% is caused due to loss of synchronization. In our system case 2 did not occur, as our implementation ensures the packet transaction is completed well within the timeslot period.

### 4.4. Choice of radio

The general PHY requirements for IEEE 802.15.4 are specified in Section 10 of standard [2]. Further, the common requirements of a radio is to provide (a) flexible State Machine to adapt the timeslot structure, (b) support for accepting MAC Protocol Data Unit (PDUs), and (c) deliver the modulating scheme suitable for IEEE 802.15.4 communication. In this paper, we have considered O-QPSK PHY operating at 2.4 GHz for short range and SUN FSK PHY operating at 868 MHz for long range. The brief over-view of commercially available radios for IEEE 802.15.4 is provided below.

- **Short Range Radio:** Our survey for short range radio to implement TSCH Dy-MAC included CC2480 [22], CC2420 [23], MCR20AVHM [24], DW1000-1-TR-13 [25] and ADF7242 [5]. The Texas Instruments CC2480 chipset's current consumption was found to be higher than the rest of the chipsets surveyed. The DW1000-1-TR-13 has support restricted to Binary Phase Shift Keying (BPSK) and thus not suitable for our data rate requirement. One drawback among CC2480, CC2420, MCR20AVHM and DW1000-1-TR-13 is that they do not provide us the flexibility to implement the timeslot structure proposed in the IEEE 802.15.4e-TSCH standard. In contrast to the above hardware, ADF7242 with O-QPSK PHY with data rate of 250 kbps, emerges as a suitable candidate to demonstrate our implementation.
- **Long Range Radio:** We surveyed several radios operating in the sub-GHz range to demonstrate Long Range TSCH Dy-MAC implementation. This included, BRAVO-T868 [26], ADF7030 [27] and ADF7030-1 [28]. To demonstrate the long range TSCH Dy-MAC, we have chosen the radio ADF7030-1 [28] with a SUN FSK PHY layer, which operates in 863–870 MHz range. The chosen radio satisfies the required modulation scheme of 2-FSK. Additionally, ADF7030-1 provides multi-rate function in the range between 0.1 to 300 kbps. Furthermore, the ADF7030-1 has two power amplifiers and provides configurable transmit power starting from −20 dBm to +17 dBm, with a step size of 0.1 dB. A variant of this radio is the ADF7030. This radio operates in the range 169.4 to 169.6 MHz. Due to country specific regulatory requirements, we chose ADF7030-1.

Having chosen the two radios, some of the common features of ADF7242 and ADF7030-1 include low power, flexible register settings for configuring modulation schemes, packet size, and moldable state machine. The following sub-section provides the implementation of TSCH Dy-MAC with ADF7242 and ADF7030-1 radios.

### 4.5. Brief description of ADF7242 and ADF7030-1 radio

***ADF7242*:** The controller to the radio are a set of commands issued over SPI communication for: (a) state transition, (b) channel hopping, (c) setting modulation scheme and (d) packet configuration. The radio to controller in-turn is capable of generating an interrupt for different events. We have configured the radio to generate interrupts to improve the overall reliability of communication and hardware. For instance the "RC_READY" interrupt, indicates that the commanded state transition is complete and radio is ready for a new command. Similarly "SFD_RCVD" (SFD received) interrupt, indicates that SFD of the packet is received and "CCA_COMPLETE" interrupt, indicates CCA operation is completed and the result stored in "read_rssi" register and CCA flag in status byte is valid.

The register configurations such as interrupt enable, packet configuration and setting of modulation scheme require 3 bytes of data to be sent over SPI, which takes 125 μs. Along with these PHY layer settings, the data packets, EBs, and the enhanced acknowledgment (Ack) packets are created in the MAC stack i.e., the software running on the embedded platform and communicated to the ADF7242. This communication requires a minimum of 250 μs.

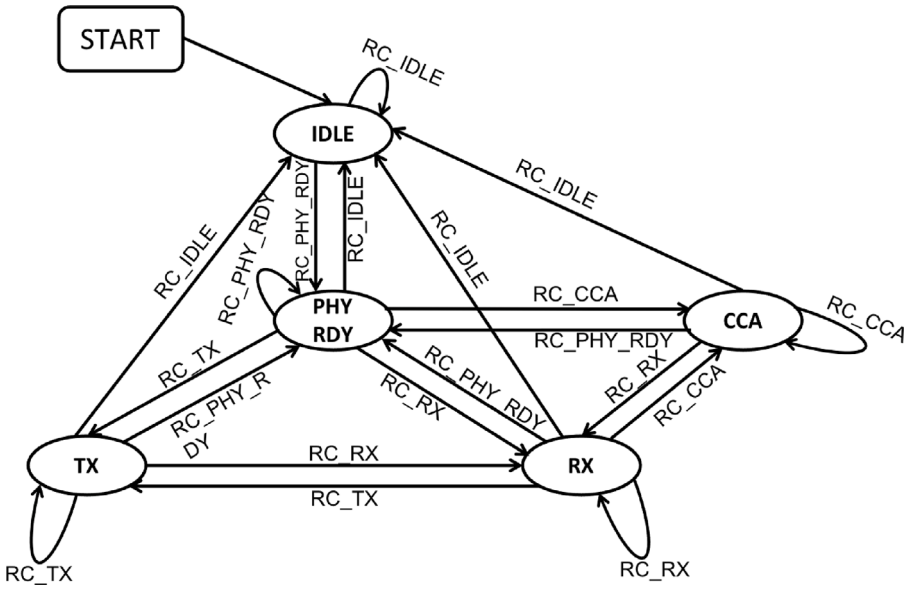



**Fig. 2.** Finite state machine of ADF7242.

*4.5.1. ADF7242 state machine*

The state transition employed in the implementation of TSCH Dy-MAC with ADF7242 is depicted in Fig. 2. The radio states utilized for TSCH Dy-MAC implementation are IDLE state, Physical ready (PHY_RDY) state, Transmit (TX) state, Receive (RX) state and Clear Channel Assessment (CCA) state. Unlike the state machine diagram indicated in the datasheet [5], the SLEEP and MEAS (Chip temperature Measurement) states are excluded. This choice of Finite State Machine (FSM) design is explained in Section 4.7.

The PHY_RDY state supports the RF frequency synthesizer block and system calibration, thus the communication frequency is set in this state. The ADF can be made to transit to this state using RC_PHY_RDY command. The transition to RX state and TX state can be achieved using RC_RX and RC_TX command respectively. Upon entering these states, the receiver and transmitter block of the radio are enabled. In RX state, the received data frame is stored in the RX_BUFFER of the radio, which can be read by the micro-controller and in the TX state the data frame which is written to TX_BUFFER is transmitted. In CCA state, the ADF will calculate the Received Signal Strength Indicator (RSSI) value and stores the result in "read_rssi" register, which can be read by the controller. The CCA state also provides information whether the channel is busy or otherwise. To maximize the battery life, all radio registers are configured in IDLE state using the RC_IDLE command. This state consumes 1.8 mA and is 8.2 mA lower than PHY_RDY state and 17.2 mA lower than RX state. The TX state consumes the maximum current of 21.5 mA.

*4.5.2. TSCH timeslot structure mapped to ADF7242's FSM*

The state transition of ADF7242 are bound to be completed within the time frame of *macTsTimeslotLength*. A single state transition process is executed as following.

1. Check if the radio is ready to accept a state transition command.
2. Check the current state of the radio.
3. Verify if the destination state transition is allowed by the FSM.
4. Transfer the corresponding state transition command via SPI.
5. Once the commanded state transition is completed an "RC_READY" interrupt is issued by the radio.

This process of state transition takes a minimum of 250 μs of SPI communication. If the radio is not ready to accept a state transition command or if FSM does not allow the desired state transition, the process is terminated. The state transitions are sequenced to complete the Tx or Rx timeslots as shown in Figs. 3 and 4. The blue shaded section of Figs. 3 and 4 depicts ADF7242's state machine mapped to the standard's Tx and Rx timeslot structure.

Fig. 3 shows Tx timeslot, the ADF7242 is initially in the IDLE state, which will transit to PHY_RDY state in the time *macTsCcaOffset*. In *macTsCca*, the radio transits to CCA state and measures the channel energy and transits back to PHY_RDY state. During *macTsRxTx*, the ADF will transit to TX state and within the period of *macTsMaxTx*, the ADF will transmit the data frame. During *macTsRxAckDelay*, the turnover from TX state to RX state is performed and the ADF waits for the Ack for the duration of *macTsAckWait*. Soon after receiving the Ack, the ADF will transit back to PHY_RDY state followed by IDLE state within the *macTsMaxAck* period.

Fig. 4 shows the Rx timeslot. The ADF7242 will transit from Idle state to PHY_RDY state during *macTsRxOffset*. The transition to RX state and data packet reception is completed during *macTsRxWait*. Over *macTsTxAckDelay*, the ADF transits to TX state from RX state. Transmission of the Ack, transition to PHY_RDY state and finally to IDLE state is completed in *macTsMaxAck*. The experimental values for these MAC PIB attributes are discussed in Section 5.3, also depicted in Figs. 9 and 10.

***ADF7030-1*:** The radio to controller set of interrupts provide "Packet_Sent", "Packet_Received" and "CCA_complete" for improved performance. Since there are a significant number of configuration registers to be set from the controller to the radio, a tool is used to generate a configuration file and SPI communication is used to configure the registers from this file. The radio profile configuration registers consists of features such as CCA threshold, low power mode, GPIO (General Purpose Input Output) and channel configuration. The packet configuration registers sets the payload location, sync word and the payload format. While memory access commands allow us to read and write the registers, there is a provision to also read and write from the internal memory. The radio commands are used to enforce state transitions and typically require 250 μs.

*4.5.3. ADF7030-1 state machine*

Fig. 5 depicts the state transition employed for TSCH Dy-MAC implementation. The radio states utilized are PHY_OFF, PHY_ON, PHY_TX (Transmit), PHY_RX (Receive), CCA and a configuring state. In PHY_OFF state the radio oscillator is enabled and allows memory access. This state can be configured using the command CMD_PHY_OFF or by a cold start. While the default current consumption is 1.9 mA, subsequent

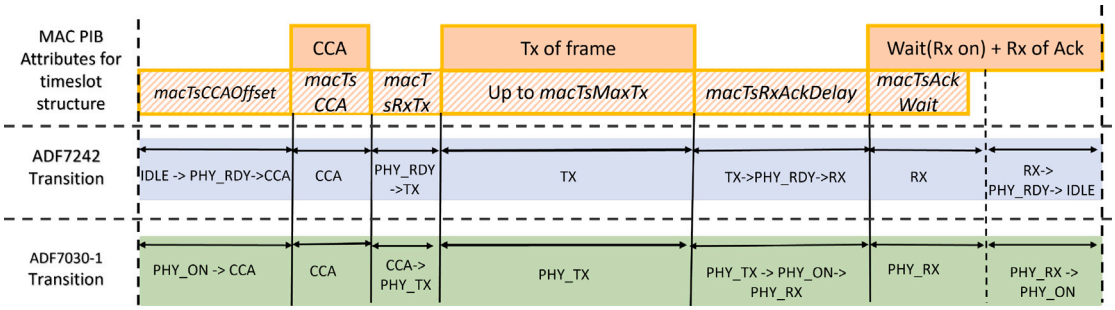



**Fig. 3.** Transmitter timeslot structure of acknowledged transmission defined in Section 6.5.4.2 of the standard. The implemented finite state machine is mapped to this structure.

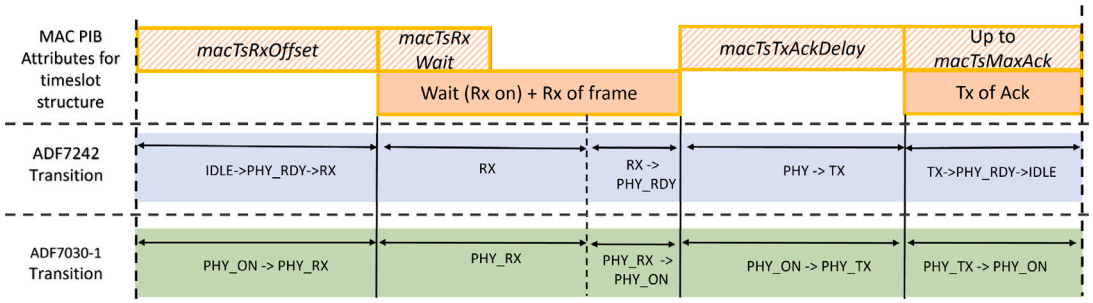



**Fig. 4.** Receiver timeslot structure of acknowledged transmission defined in Section 6.5.4.2 of the standard. The implemented finite state machine is mapped to this structure.

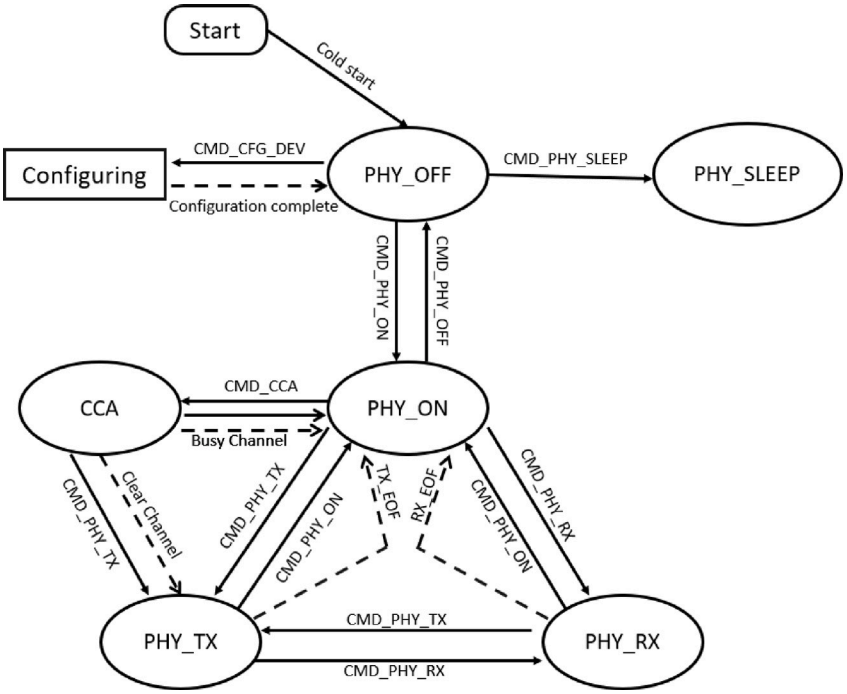



**Fig. 5.** Finite state machine of ADF7030-1.

to programming the registers from the configuration state, the current consumption in PHY_OFF is now at 3.7 mA. This increase can perhaps be attributed to memory consumption. An additional 0.3 mA is consumed by a clock source and regulators. Once powered up they can only be switched off either in power down or in PHY_SLEEP thus, the PHY_OFF state consumes 4 mA.

The PHY_ON state is an intermediate state between the idle state (PHY_OFF) and active states. During the first transition to PHY_ON from PHY_OFF the internal regulators are powered up. PHY_ON allows transitions to PHY_TX, PHY_RX and CCA using CMD_PHY_TX, CMD_PHY_RX and CMD_CCA command respectively. Transition to PHY_ON state can be made using the command CMD_PHY_ON or by enabling radio's auto-transitions such as end of frame transmission (TX-EOF), end of frame reception (RX-EOF) and Busy Channel. In PHY_TX the radio transmits the data stored in packet memory section, and in PHY_RX it stores the received frame in packet memory section. The CCA state is used to evaluate the channel and uses RSSI threshold to transit to either PHY_TX or PHY_ON.

**Table 5**
MAC sublayer functions to API mapping.

| MAC functions defined in Table D.6 of the standard | API function |
|---|---|
| Transmission of data | timeslot_tx() |
| Reception of data | timeslot_rx() |
| Time-stamping of incoming data | Get_time() |
| Transmit Beacons | Enhanced_Beacon() |
| Receive Beacons | Receive_Beacon() |
| TSCH Channel Access | Channel_Write() |
| TSCH Frame counter | ASN |
| Synchronization in TSCH PAN | Ack_based_synchronization() & Frame_based_synchronization() |
| Channel Hopping | Channel_Write() |

#### 4.5.4. TSCH timeslot structure mapped to ADF7030-1's FSM:

The ADF7030-1's acknowledgment based communication has to transit several states within the user-defined *macTsTimeslotLength* time. Each of these state transitions are performed as follows:

1. Read the radio's MISC_FW register to obtain the current state of the radio.
2. Check if the radio is not transiting between two states or in the middle of a frame communication. This is a required step to ensure the radio commands issued by the controller is synchronized with the timeslot structure.
3. Verify if the transition to a state is allowed by FSM.
4. Poll the radio for transition status until the state transition is complete.

The above four steps requires upto 500 μs of SPI communication. As discussed in Section 4.5.2, the process is terminated if FSM is violated or radio is busy. The mapping of these ADF7030-1 state machines to TSCH timeslot structure is shown by green shaded section of Figs. 3 and 4.

In Tx timeslot shown in Fig. 3, the radio starts with PHY_ON to CCA during *macTsCCAOffset* and performs CCA in the period of *macTsCCA*. It transmits the packet in the period of *macTsMaxTx*. In the duration of *macTsRxAckDelay*, radio transits from PHY_TX to PHY_RX via PHY_ON. The maximum wait for Ack is defined by *macTsAckWait* and finally the radio transits to PHY_ON state. In Rx timeslot shown in Fig. 4, the radio is initially in PHY_ON state, it transits to PHY_RX state during *macTsRxOffset*. It waits for the packet for a period of *macTsRxwait* in RX state. After packet reception, radio enters PHY_ON state followed by transition to PHY_TX which corresponds to *macTsTxAckDelay*. During *macTsMaxAck* the radio transmits the acknowledgment and transits to PHY_ON state.

During our TSCH timeslot implementation for ADF7242 and ADF7030-1, we learn our ***fifth lesson,*** that the state transition command should be issued after verifying the on-going state and associated status of the radio. This is to ensure that the radio is in synchrony with controller's flow of commands. As an outcome, if the status of radio is erroneous, we can terminate the current timeslot process to perform a radio reset and ensure the communication reliability.

### 4.6. API description

From our in-depth implementation of IEEE 802.15.4e-TSCH standard across two PHY layer radio systems, we abstracted and put together an API. The API supports configuring the radio, managing the FSM states, frame transfer between radio and controller, handling interrupt from radio to controller (R_to_C), TDMA & FDMA, Tx and Rx of Packet, Tx and Rx of EB, Synchronization, Slotframe design and scheduling. Table 5 depicts the mapping of the implemented API functions to the standard defined MAC functions as shown in Table D.6 of standard [2].

1. **Config_all():** This function is used to configure the registers of radio according to the requirement of TSCH Dy-MAC.
2. **FSM():** The state transition process in both the radios are implemented as a function. This function requires a single destination state as the argument. It ensures all steps of a state transition as discussed in Sections 4.5.2 and 4.5.4.
3. **read_address() & write_address():** These functions are used to read and write packets, ack and EB to the radio's memory.
4. **R_to_C_Interrupt_Handler():** This function is used to analyze the interrupt generated by the radio. It includes (a) clearing the generated interrupt, (b) categorizing the interrupt based on the source and (c) providing the time-stamp of interrupt arrival.
5. **Get_time():** It utilizes the internal timer of the nRF52840 to access the local timestamp of the node.
6. **timeslot():** This function utilizes the services of nRF52840's application timer to provide TDMA slotting. At the beginning of a timeslot, the timer is initialized to the timeslot period. Once the timer interrupt occurs the timeslot is incremented. Based on the *macTxType* and *macRxType* entries in the *macLinkTable*, timeslot_tx() or timeslot_rx() is invoked.
7. **channel_write():** This function implements the channel hopping as given in Eq. (1).
8. **timeslot_tx():** The whole Tx structure is executed by this function. This function maps the radio's state to standard Tx timeslot structure as shown in Fig. 3.
9. **timeslot_rx():** The whole Rx structure is executed by this function. This function maps the radio's state to standard Rx timeslot structure as shown in Fig. 4.
10. **Enhanced_Beacon():** This function will broadcast the EB carrying all IEs. This function is specific to PAN coordinator.
11. **Receive_Beacon():** This function is specific to new nodes trying to join the network. The node-joining process is executed by this function as explained in Section 4.2.
12. **Time_correction_offset_Calculation:** This function calculates the time correction offset by utilizing the timestamp taken during packet received interrupt as explained in Section 4.1.
13. **Frame_based_synchronization() & Ack_based_synchronization:** These functions ensure synchronization between child and parent node as explained in Section 4.1
14. ***macSlotframeTable* and *macLinkTable*:** These structures helps in modifying the slotframe design according to the requirement.

We analyzed execution time for the created APIs and found that the timeslot_tx and timeslot_rx execution time is 1420 μs for a controller running at 64 MHz. The ***sixth lesson*** we learnt is that even if we employ a controller running at 100 MHz instead of 64 MHz, our calculations show that the execution times reduces to 910 μs. Clearly, SPI communication delay overwhelms the execution delay.

### 4.7. Design lessons for implementors

As our ***lesson seven*** is related to order of implementation, it is recommended to start the implementation process with framing the timeslot structure, followed by TDMA and FDMA construction. The MAC functions such as PAN Formation and Time synchronization are built on the basis of this timeslot structure. The timeslot structure


**Fig. 6.** Experimental setup depicting TSCH Dy-MAC network.

design starts with the mapping of radio state and activities to the standard structure.

Our ***eighth lesson*** is related to radio calibration. Since the IEEE 802.15.4e-TSCH rests on time synchronization, temperature variation (ambient and chip related) can affect the oscillator. The ADF7030-1 radio supports a command to perform radio calibration and this might have to be issued frequently in factory floor and other industrial environments. This step typically requires 5 s for calibration and 690 ms for re-calibration. A sleep state erases the firmware module completely and thus calibration is mandatory. Regarding ADF7242 the register configuration are completely lost in a sleep state. The reconfiguration in case of ADF7242 would consume around 1.5 ms. Given this background, an industrial radio has minimal usage of the sleep state in the state machine. Our calculations shows that sleep state is useful if the active time of the node is less than 10% of the slotframe period. Our assumption here is that the slotframe period is 50 s.

Our ***ninth lesson*** is related to state machine design. While implementing the TSCH Dy-MAC on two different radios, we came across some features in each radio which aided in implementation. Inspired by these features, we propose an ideal FSM structure. The features of ideal FSM are discussed as follows.

- *A low power state with configuration retention*: The IDLE state as present in ADF7242 is desirable because it is low power and also retains the configuration, and thus assists in design of a battery efficient radio. This is unlike the PHY_OFF in ADF7030-1, which retains the configuration, but consumes more power.
- *An automatic CCA transition based on channel assessment:* This feature is desirable and similar to ADF7030-1 radio that provides auto-transition consequent to the CCA state. The absence of this feature in ADF7242 adds extra SPI overhead of approximately 300 μs to read and direct the transit to next state.
- *A status byte returning both state and status of the radio:* Before a state transition, it is crucial to confirm the current state of the radio. While the ADF7242 provides a status byte, ADF7030-1 the state of the radio is obtained by reading the MISC_FW register which adds 250 μs SPI communication time.
- *Should provide state transition complete interrupt:* After a state transition is commanded the radio should provide an acknowledgment, ensuring its completion. This interrupt is used to define the precise value for MAC PIB attributes. The chosen ADF7242 does this by providing an interrupt, whereas in ADF7030-1 radio the status of transition is verified by polling.
- *Interrupt enabled state transitions*: This features helps us to remove the SPI delay for state transition. The ADF7030-1 provides state transition based on interrupt.

## 5. Experimental settings and results

For our factory floor environment experiments, we interfaced an analog pressure sensor - M3256 [29] with controller's ADC. Since sensing and conversion requires 56 μs, this process can be accomplished during *macTsCCAOffset*, thus a sense and send process can be completed within the designed timeslot period. Fig. 6 captures the network with the sensor nodes placed on the machines to monitor their health. The transmit power was set to +3 dBm. Our slotframe implementation comprises of four timeslots of 10 ms for ADF7242 and 26 ms for ADF7030-1. We have a Time Slotted Channel Hopping implementation to accommodate 76 bytes of payload. This experimental testbed was used to validate our implementation for the following features:

1. Time Synchronization of all the nodes present in a TSCH Dy-MAC network.
2. The node joining time — We evaluate our node joining algorithm, and compute the estimated time required to join a productive network.
3. Analysis of TSCH MAC PIB attributes for *macTimeslotTemplate* - Here, we define each state transition and compare the same to standard's timeslot template.
4. Packet Delivery Ratio (PDR) - To check the accuracy of our implementation that incorporates frequency hopping, slotframe structure, timeslot design and sensor data.
5. Range Performance: To characterize the range of 2.4 GHz and 868 MHz TSCH MAC radios under in-situ condition.
6. Effect of interference — We test the performance of TSCH Dy-MAC under wireless interference in realistic settings. A wideband signal covering the entire radio spectrum is introduced as a source.
7. Network Topologies: To test the performance for TSCH Dy-MAC for star, tree and mesh network topologies. For a given number of nodes, we provide an estimate of number of timeslots required in a slotframe.

These parameters are as discussed in detail in the following subsections.

### 5.1. Time synchronization

Fig. 7 illustrates the packet arrival time at the receiver with and without synchronization. The x-axis of the graph indicates ASN (timeslot number) and y-axis indicates the Rx node timeline. The Radio's RX state begins soon after *macTsRxOffset*. The 'Green' horizontal line is the ideal expected arrival time. As explained in Section 4.1, there are two possible cases, when two IIoT nodes communicate with each

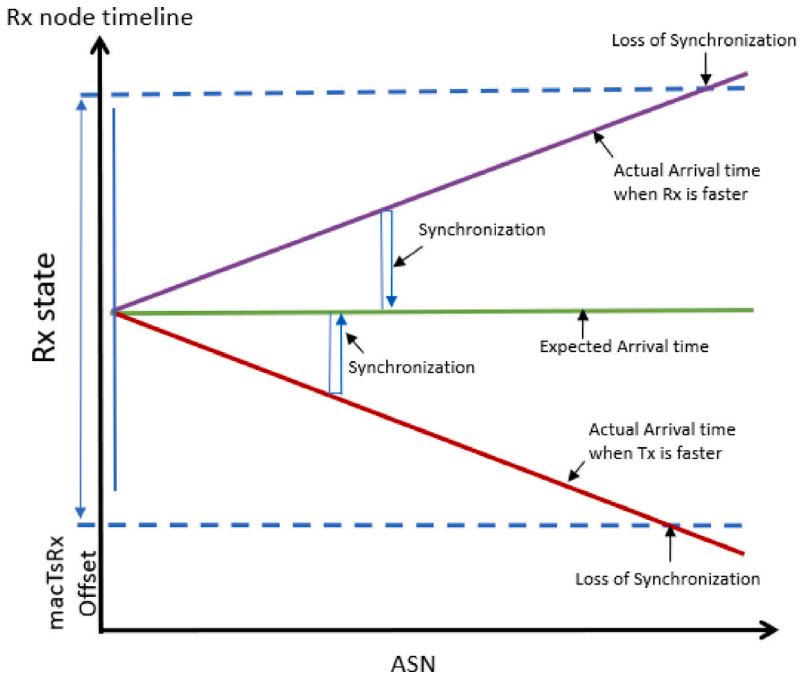



**Fig. 7.** Frame and Ack Based Synchronization to handle variation in packet arrival time.

**Table 6**
Effect of synchronization on Packet Error Ratio.

| Number of Packets transmitted | Number of Packets lost | PER (%) |
|---|---|---|
| 6660 (Without Synchronization) | 102 | 1.531531 |
| 6660 (With Synchronization) | 7 | 0.105105 |

**Table 7**
Joining time for ADF7242 & ADF7030-1.

| Condition | New node joining time (ms) |
|---|---|
| ADF7242 | 126 |
| ADF7030–1 | 116 |

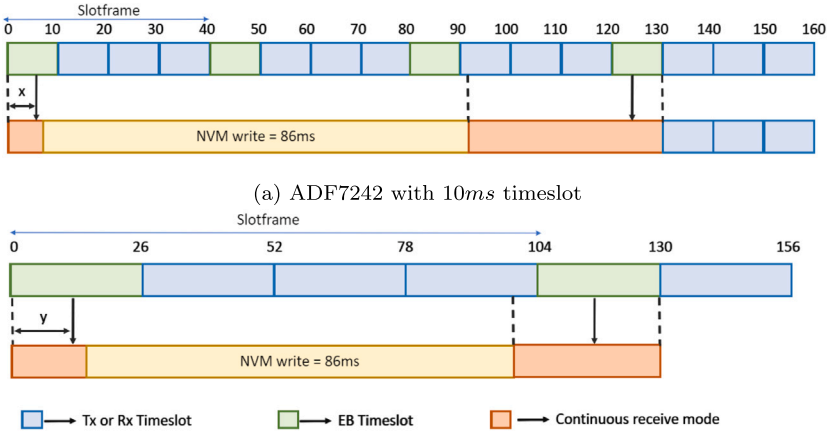
other: (1) the Rx might be faster than the Tx, or (2) Rx might be slower than the Tx. Consider a scenario where child is the Rx node, the early arrival time (red line) of packet is indicative of Tx being faster, and delayed arrival time (purple line) indicates Rx is faster. This is corrected in real-time by the Frame based synchronization. This ensures the Rx child to be in sync with the Tx parent. Similarly, when the child is Tx node the offset is corrected by Ack based synchronization. Even the slightest deviation from the expected arrival time is corrected using these synchronization algorithms as discussed in Algorithm 1. The packet arrival time is computed by taking SFD_Arrival time (beginning of packet arrival). For ADF7242, SFD_Arrival time was taken for 6660 timeslots each of 10 ms spread over a total of 26648 timeslots with a inter timeslot gap of 30 ms. Similarly for ADF7030-1 the SFD_Arrival time was taken for 5000 timeslots each of 26 ms spread over a total of 20008 timeslots with a inter timeslot gap of 78 ms. Table 6 indicates communication between a pair of ADF7242 nodes have lost 102 packets out of 6660 packets. This evaluates to a packet error ratio of 1.53% without enabling Ack based or Frame based synchronization. With our synchronization algorithms, the packet loss ratio is significantly reduced to 0.1%. These numbers are comparable for ADF7030-1.

### 5.2. Node joining time

Our implementation assumes that the network originates with the presence of the PAN coordinator and sensor nodes join the network sequentially. The node joining time majorly depends on the frequency of broadcast and the successful reception of EB. In our implementation, the EBs are broadcast once in a slotframe and nodes perform passive scanning in a pre-defined channel, awaiting an EB. The last byte of the received beacon is checked to ensure that all IEs required by the node are received successfully. If the node fails to receive and analyze all IEs from the EB broadcast, then it waits for another EB. Fig. 8 depicts the number of timeslots required for node joining. In Figs. 8(a) and 8(b), $x$ and $y$ are the EB arrival time, which is ~4 ms for ADF7242 and ~11 ms for ADF7030-1 respectively. The process of writing the IEs in the NVM, requires 86 ms for our chosen controller. The node joining is accomplished by receiving the successive beacon in the following slotframe. Table 7 shows the total time consumed for node joining from the first beacon reception.

### 5.3. Analysis of TSCH MAC PIB attributes for macTimeslotTemplate

Figs. 9 and 10 depicts the values for MAC PIB attributes obtained from the radios. As mentioned earlier, in our implementation, the communication between the micro-controller and the radio chip is through the SPI bus. Our experiments revealed that a certain amount of delay is required for reliable SPI communication. The evaluated SPI overhead for state transitions in ADF7242 is 250 µs. Unlike the previous work [6] the timeslot period achieved is 10 ms, irrespective of CCA. In order to achieve this standard timeslot of 10 ms, we support a smaller payload of length 76 bytes. The transition to CCA, performing the channel assessment, storing the result and finally transiting back to PHY_RDY consumes around 450 µs. In our ***tenth lesson*** for optimized battery usage, this time is added in the low power IDLE state at the Rx node. Fig. 9 characterizes the timing values for MAC PIB Attributes of transmitter timeslot as explained in Section 4.5.2. Likewise, Fig. 10 depicts the timing values for MAC PIB Attributes of receiver timeslot. The maximum wait for packet or Ack is configured to 3000 µs and maximum transmission for packet is 2500 µs. These values were calculated for a 250 kbps data rate and payload size of 76 bytes. An additional 250 µs margin is given at both the side of the receiver end to allow compensation for drift as explained in Section 4.1. The timing value of *macTsRxWait* & *macTsAckWait* are shown in Figs. 9 and 10. This figure includes overhead for packet or Ack wait and reading the same




(a) ADF7242 with $10ms$ timeslot

(b) ADF7030-1 with $26ms$ timeslot

**Fig. 8.** Node Joining time.

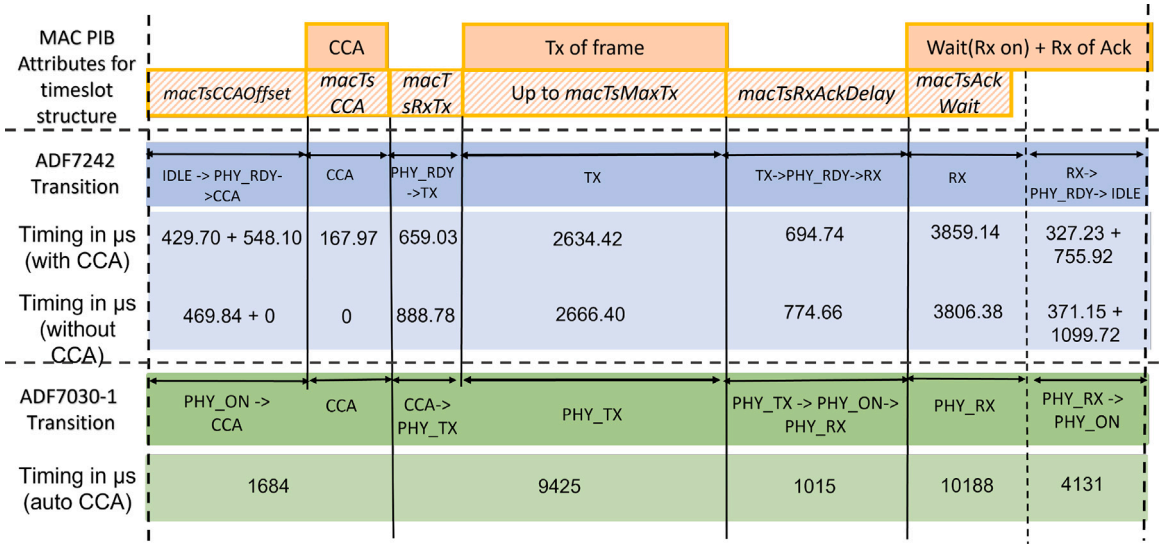



**Fig. 9.** Values for Tx MAC PIB attributes.

from radio to MAC via the SPI. Similarly, *macTsMaxTx* & *macTsMaxAck* corresponds to writing a packet or Ack frame from the controller to the radio over SPI for an ensuing transmission. With the implementation of time synchronization, the parameters *macTsRxWait* and *macTsAckWait* were tuned to reduced the drift. After transmission or reception of packet the radio automatically transits from TX or RX state to a less power consuming PHY_RDY state, which ensures low power operation. We enabled auto-transition feature of the radio, and 5 ms is gained.

In case of ADF7030-1, due to the chosen data-rate of 50 kbps and SPI overhead of 500 µs for each state transition, the timeslot period achieved for a 76 byte of payload is 26 ms. Due the auto-transitions available from CCA states, there is no additional overhead in performing CCA. Thus, we provide timing value with CCA implemented. For low power operation, radio is programmed to transit to PHY_ON state from PHY_RX and PHY_TX after packet reception or transmission.

### 5.4. Range performance and packet delivery ratio

Our goal is to experimentally determine the fade margin for the factory floor environment. The receiver sensitivity of ADF7242 is −95 dBm [5] and −106.4 dBm [28] for ADF7030-1. Unfortunately, since ADF7242 only supports CCA mode 1, we have considered energy detection based RSSI as the evaluation parameter to measure the range performance of the radio. In our experiment, we transmitted $10^6$, 76-byte packets over different times of the day, in a dynamic factory environment. Fig. 11 provides the plot for RSSI obtained for varying inter-node distance along with the variance bound. From Fig. 11, it can be inferred that reliable communication range for ADF7242 is ∼3 m and ADF7030-1 is ∼13 m for a fade margin of 20 dB. The obtained PDR within this range is 99.09% for ADF7030-1 and 97.63% for ADF7242. For a fade margin of 22 dB, the ADF7030-1 provides a PDR of 99.99%.

### 5.5. Effect of interference and PDR

To test the effect of interference on the TSCH Dy-MAC network, our noise source consists of a Universal Software Radio Peripheral (USRP) X310 transmitting a continuous wide-band signal. The transmit power was set to 33 dBm. Table 8 depicts the PDR observed under no interference and under the above explained condition. This setting was tested in different environments such as laboratory, factory floor and indoor hallway. We found that the worst case PDR for ADF7242 to be 66.66% when 25% of channel is blocked in factory setting and that for ADF7030-1 to be 36.71% for 60% of channel blocking. Clearly, our TSCH implementation appears to be functioning satisfactorily for the designed slotframe structure.

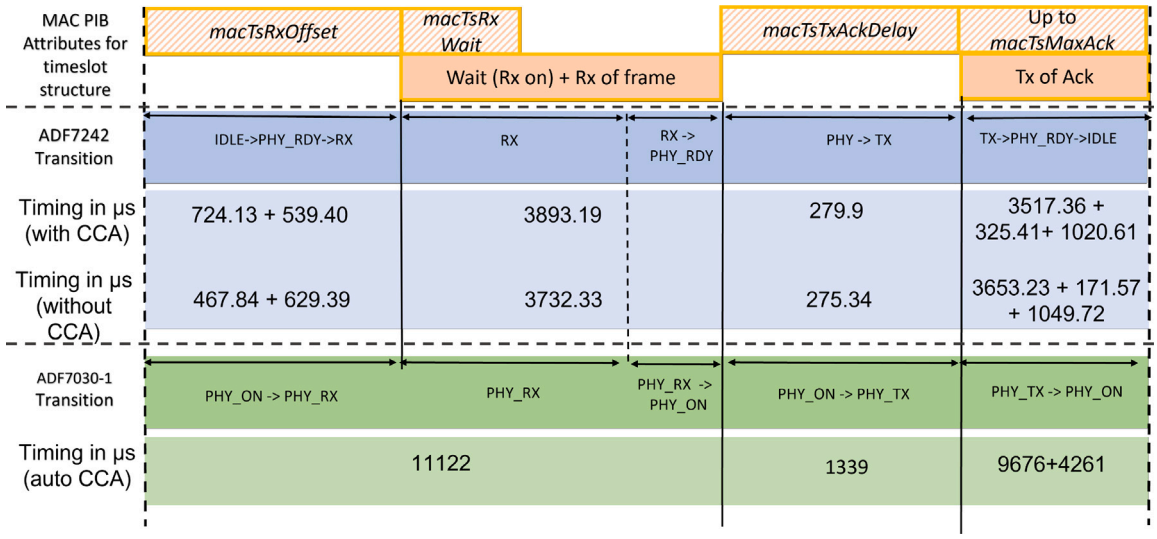



**Fig. 10.** Values for Rx MAC PIB attributes.

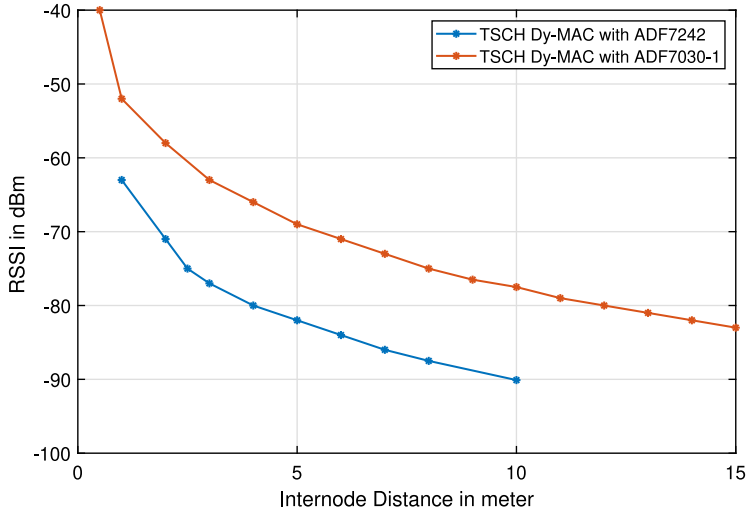



**Fig. 11.** RSSI for varying inter-node distance in a factory floor environment.

**Table 8**
Effect of interference on packet delivery ratio.

| Condition | PDR of ADF7242 | PDR of ADF7030-1 |
|---|---|---|
| No injected interference | 97.15% | 99.09% |
| Injected interference in Channel | 66.66% | 36.71% |

### 5.5.1. Effect of interference on PDR for multi-hop network

We can observe that PDR is directly related to percentage of injected interference from Table 8. Consider a two-hop network with probability of successful packet delivery being $P_1$ and $P_2$ at each hop respectively. The packet drop events are assumed to be independent, hence the total probability of packet delivery from source to destination $P_{total}$ is:

$$P_{total} = P_1 \times P_2 \tag{3}$$

If all the nodes in a network experience equal amount of interference, one may generalize the probability for a n-hop network being successfully delivered $P_d$ as:

$$P_{total} = P_d^n \tag{4}$$

### 5.6. Topologies implemented

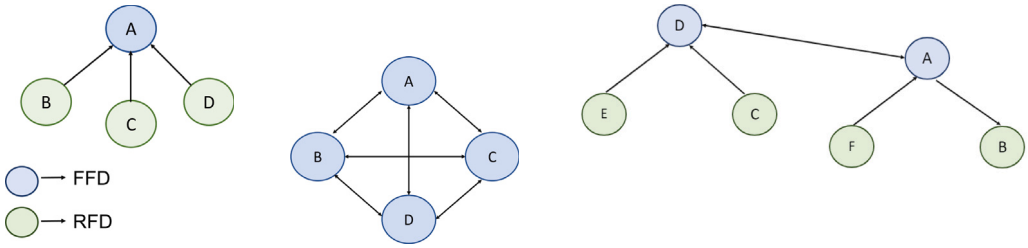
For any TSCH network, the design of slotframe is dependent on the network topology and the number of nodes. In this section, we provide an analytic study on the minimum number of timeslot required for TSCH Dy-MAC implementation for different topologies.

Consider the total number of nodes in a slotframe is $N$. Let $m, n \in N$ be the number of coordinators and the number of end nodes respectively. Since each coordinator is capable of broadcasting EBs individually, it is recommended that the coordinators to send EBs in different timeslots. This number of EBs to be broadcasted is determined by the design of a network. The minimum number of timeslots required for each topology for dedicated links are provided below.

For a ***Star network***, the minimum number of timeslot $T_{star}$ required as shown in Fig. 12(a), is given by Eq. (5)

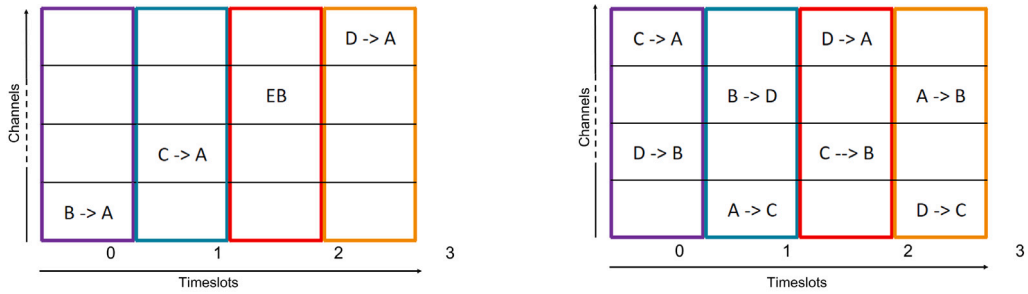
$$T_{star} = n + EB \tag{5}$$

For a ***Mesh network***, where all nodes can communicate with each other, $N = m$ as shown in Fig. 12(b). Let the total number of channels employed in a single timeslot be $x_c$, and in a timeslot a maximum of $\lfloor m/2 \rfloor$ communication links can be established and hence in a network maximum number of simultaneous link in a timeslot possible are $x_n =$

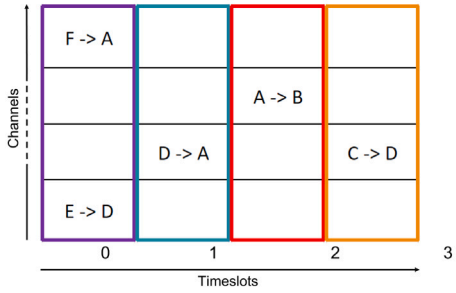



(a) Star (b) Mesh (c) Tree

**Fig. 12.** Network Topologies.




(a) Star (b) Mesh




(c) Tree

**Fig. 13.** Slotframe structure for different topologies.

$min(\lfloor m/2 \rfloor, x_c)$. Then the minimum number of timeslots required in a slotframe design is given by Eq. (6).

$$T_{mesh} = \frac{(m)(m-1)}{x_n} + EB \tag{6}$$

A ***Tree structure***, can be considered as a combination of both mesh and star structure as shown in Fig. 12(c). The minimum number of timeslots required to support such a tree structure is given by Eq. (7).

$$T_{tree} = max(n_x) + T_{mesh} \tag{7}$$

where $n_x$ is the number of end nodes communicating with $x$th coordinator.

Fig. 13(a) & 13(b) shows a simple TSCH slotframe designed star and mesh network with four nodes. The tree network with six nodes is shown in Fig. 13(c)

## 6. Conclusion

This work constructs the IEEE 802.15.e-TSCH MAC using commercially available ARM controllers and radio hardware. The implementation involved constructing and building several algorithms, code optimizations, testing the deployment etc. Our implementation exhaustively covers the IEEE standard document as well as adheres to several recommendations. We present a dynamic MAC stack which abstracts the state machine of the radio chip. The developed MAC is tested with two radios. Several lessons were learnt from problems faced during the Dy-MAC implementation. Our first three lessons are on implementing synchronization for TSCH Dy-MAC, lesson 4 provides optimized node rejoining method. From lesson 6, it is evident that the impact of using the higher accurate clock source on the timing values is not completely beneficial. Further, we present five lessons on state machine properties to achieve low power and list of desirable features for TSCH Dy-MAC deployment, which will ensure that a large number of platform hardware can be supported.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Prabhakar T V reports financial support was provided by The Ministry of Electronics and Information Technology ( MEITy).

## Data availability

No data was used for the research described in the article.

## Acknowledgments

We express our gratitude to the MeitY, Government of India (MITO-003) for funding this work. A part of funding was also received from

Centre for Networked Intelligence (CNI), Indian Institute of Science, Bangalore. Thanks to our partners, CDAC-BLR, IIT-Hyderabad and IIT-Bhubaneswar.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.adhoc.2023.103268.

## References


[1] J. IEEE, IEEE standard for low-rate wireless networks, in: IEEE Std 802.15. 4-2015 (Rev. of IEEE Std 802.15. 4-2011), 2016, pp. 1–709.

[2] IEEE standard for low-rate wireless networks, in: IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015), 2020, pp. 1–800, http://dx.doi.org/10.1109/IEEESTD.2020.9144691.

[3] I. Sanyal, D.P. Rao, R. Gunasekaran, S. SM, T.V. Prabhakar, Lessons learnt from the implementation of the IEEE 802.15.4e-TSCH MAC, in: 2023 15th International Conference on COMmunication Systems & NETworkS, COMSNETS, 2023, pp. 658–666, http://dx.doi.org/10.1109/COMSNETS56262.2023.10041267.

[4] nR.F. 52840, Nordic semiconductors, 2022, https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.7.pdf. (Accessed 14 April 2023).

[5] ADF7242, Analog, 2022, https://www.analog.com/media/en/technical-documentation/data-sheets/ADF7242.pdf. (Accessed 14 April 2023).

[6] D. De Guglielmo, G. Anastasi, A. Seghetti, From IEEE 802.15. 4 to IEEE 802.15. 4e: A step towards the internet of things, in: Advances onto the Internet of Things: How Ontologies Make the Internet of Things Meaningful, Springer, 2014, pp. 135–152.

[7] S. Touloum, L. Bouallouche-Medjkoune, M. Yazid, M. Moktefi, A. Djamil, Performance modeling of the IEEE 802.15. 4e TSCH enabling both shared and dedicated links in industrial WSNs, Computing 104 (4) (2022) 859–891.

[8] M. Gürsu, M. Vilgelm, S. Zoppi, W. Kellerer, Reliable co-existence of 802.15. 4e TSCH-based WSN and Wi-Fi in an aircraft cabin, in: 2016 IEEE International Conference on Communications Workshops, ICC, IEEE, 2016, pp. 663–668.

[9] I. Sanyal, V. Prabhakar, IEEE 802.15. 4e-TSCH MAC stack extensions to GNU radio, in: 2022 14th International Conference on COMmunication Systems & NETworkS, COMSNETS, IEEE, 2022, pp. 199–203.

[10] OPENWSN, Intia, 2022, https://aio.inria.fr/berkeleys-openwsn. (Accessed 14 April 2023).

[11] Contiki, Github, 2021, https://github.com/contiki-ng/contiki-ng/wiki. (Accessed 14 April 2023).

[12] C.-S. Sum, M.-T. Zhou, F. Kojima, H. Harada, Experimental performance evaluation of multihop IEEE 802.15. 4/4g/4e smart utility networks in outdoor environment, Wirel. Commun. Mob. Comput. (2017).

[13] P. Tuset-Peiro, R. Gomes, P. Thubert, E. Cuerva, E. Egusquiza, X. Vilajosana, A dataset to evaluate IEEE 802.15.4g SUN for dependable low-power wireless communications in industrial scenarios, Data 5 (2020) 19, http://dx.doi.org/10.3390/data5030064.

[14] J. Munoz, T. Chang, X. Vilajosana, T. Watteyne, Evaluation of IEEE802.15.4g for environmental observations, Sensors 18 (2018) 3468, http://dx.doi.org/10.3390/s18103468.

[15] M.R. Palattella, N. Accettura, M. Dohler, L.A. Grieco, G. Boggia, Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks, in: 2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, IEEE, 2012, pp. 327–332.

[16] N. Accettura, E. Vogli, M.R. Palattella, L.A. Grieco, G. Boggia, M. Dohler, Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation, IEEE Internet Things J. 2 (6) (2015) 455–470.

[17] Y. Zhang, C. Chen, S. Zhu, An adaptive distributed scheduling algorithm for IEEE 802.15. 4e TSCH protocol, in: 2019 3rd International Symposium on Autonomous Systems, ISAS, IEEE, 2019, pp. 193–198.

[18] I. Khoufi, P. Minet, E. Livolant, B. Rmili, Building an IEEE 802.15. 4e TSCH network, in: 2016 IEEE 35th International Performance Computing and Communications Conference, IPCCC, IEEE, 2016, pp. 1–2.

[19] M. Brachmann, S. Duquennoy, N. Tsiftes, T. Voigt, IEEE 802.15. 4 TSCH in sub-GHz: Design considerations and multi-band support, in: 2019 IEEE 44th Conference on Local Computer Networks, LCN, IEEE, 2019, pp. 42–50.

[20] K. Singh, Addressing Energy and Performance Related Challenges in Networked Embedded Systems (Ph.D. thesis), 2021.

[21] K.Y. Koo, D. Hester, S. Kim, Time synchronization for wireless sensors using low-cost GPS module and arduino, Front. Built Environ. 4 (2019) http://dx.doi.org/10.3389/fbuil.2018.00082.

[22] Z-Accel 2.4 GHz ZigBee® Processor, No. CC2480 data sheet, Texas Instruments, 2022.

[23] 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, No. CC2420 data sheet, Texas Instruments, 2022.

[24] Low Power, High-Performance 2.4 Ghz IEEE 802.15.4 Compliant Transceiver with Connectivity, No. MCR20AVHM Data sheet, NXP Semiconductors, 2016, Rev. 3.2.

[25] IEEE 802.15.4-2011 UWB Transceiver Module, No. DWM1000 data sheet, Decawave, 2016.

[26] T. 868, Bravo, 2012, https://www.rfsolutions.co.uk/downloads/62a7144acc70b250DS-BRAVO-9.pdf. (Accessed 29 April 2023).

[27] ADF7030, Analog devices, 2015, https://www.analog.com/media/en/technical-documentation/data-sheets/ADF7030.pdf. (Accessed 29 April 2023).

[28] ADF7030-1, Analog devices, 2016, https://www.analog.com/media/en/technical-documentation/data-sheets/adf7030-1.pdf. (Accessed 29 April 2023).

[29] M3256-00000B-016BG, TE connectivity, 2021, https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7FM3200%7FA19%7Fpdf%7FEnglish%7FENG_DS_M3200_A19.pdf%7F20010084-00. (Accessed 29 April 2023).


**Deeksha P Rao** received her bachelor's degree from Electronics and Communication Engineering department RNS Institute of Technology, VTU in the year 2022. She is working as Project Associate in Department Electronic Systems Engineering, Indian Institute of Science Bengaluru. Her areas of interest include MAC protocol for low power low data rate network, Embedded systems, Wireless Communication, IoT (Internet of Things), Cryptography, and WSN (Wireless Sensors and Network) and Signal Processing.

**Godkhindi Shrutkirthi S.** received her Ph.D. from the Department of Electronics and Communication Engineering, National Institute of Technology Karnataka, India in 2021 and her Master's degree in Wireless Communication from Gujarat Technological University in the year 2015. Currently, she is working as a Project Scientist in Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore. Her areas of interest are: Internet of Things, MIMO wireless communications, space–time coding, Error control coding, Cooperative communication systems, MAC protocols for low power, low data rate networks, indoor localization, BLE Mesh networks, RFID communication and LoRa communication.

**Abhigna V Kumar** received his bachelor's degree from Electronics and Communication Engineering department of RNS Institute of Technology, VTU in the year 2023. He is working as a Project Associate in Department Electronic Systems Engineering, Indian Institute of Science Bengaluru. His areas of interest are MAC protocol for low power low data rate network, Embedded system, Edge AI, Communications, IoT and Networking.

**Prabhakar T V** is a Principal Research Scientist at IISc. His main area of research is Networked Embedded Systems and IoT. The focus is on hardware system design, power management algorithms, IoT data management and technologies to build smart IoT applications. Some recent work is related to RFID localization, PUF based device ID, Fast MAC access and Tactile networking, Fast and Slow loop algorithms for IoT data management, edge intelligence systems, application of powerline modems for smart sensing and machine learning for speech to text systems. His laboratory has published over 75 papers in IEEE and ACM conferences. Prabhakar holds M.Sc. (Physics), M.Sc. (Engg) and Ph.D. degrees.