

## Article

# TSCH Multiple Slotframe Scheduling for Ensuring Timeliness in TS-SWIPT-Enabled IoT Networks

Dongwan Kim <sup>1</sup>, Jung-Hyok Kwon <sup>2,\*</sup>  and Eui-Jik Kim <sup>2,\*</sup> 

<sup>1</sup> Department of Electronics Engineering, Dong-A University, 37 Nakdong-Daero 550 beon-gil, Saha-gu, Busan 49315, Korea; dongwankim@dau.ac.kr

<sup>2</sup> School of Software, Hallym University, 1 Hallymdaehak-gil, Chuncheon, Gangwon-do 24252, Korea

\* Correspondence: jhkwon@hallym.ac.kr (J.-H.K.); ejkim32@hallym.ac.kr (E.-J.K.)

**Abstract:** This paper presents a time-slotted channel hopping (TSCH) multiple slotframe scheduling (TMSS) protocol to ensure the timeliness of energy harvesting and data transmission for sensor devices with different transmission periods in Internet of Things (IoT) networks enabled with time-switching simultaneous wireless information and power transfer (TS-SWIPT). The TMSS uses a modified three-step 6P transaction to allocate power and data cells within the slotframe. The sensor device sets the slotframe length equal to the transmission period and estimates the number of power and data cells for allocation in the configured slotframe and requests cell allocation to the hybrid access point (HAP). Upon request from a sensor device, the HAP executes a cell-overlapping prevention (COP) algorithm to resolve the cell-overlapping problem and responds to the sensor device with a candidate cell list. Upon receiving the response from HAP, the sensor device determines its power and data cells by referring to the cell list. We conducted experimental simulations and compared the TMSS performance to that of the legacy TSCH medium access control (MAC) with a single slotframe and the harvest-then-transmit-based modified enhanced distributed coordination function (EDCF) MAC protocol (HE-MAC). The results showed that TMSS outperforms legacy TSCH MAC and HE-MAC in terms of delay, effective throughput and energy utilization.

**Keywords:** cell overlapping prevention; energy harvesting; IEEE 802.15.4 TSCH; Internet of Things; timeliness; TS-SWIPT



**Citation:** Kim, D.; Kwon, J.-H.; Kim, E.-J. TSCH Multiple Slotframe Scheduling for Ensuring Timeliness in TS-SWIPT-Enabled IoT Networks. *Electronics* **2021**, *10*, 48. <https://doi.org/10.3390/electronics10010048>

Received: 11 November 2020  
Accepted: 24 December 2020  
Published: 30 December 2020

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As the Internet of Things (IoT) has become widespread, many types of sensor devices have been deployed in IoT networks to support various IoT applications, such as smart grid, smart cities and smart manufacturing [1–4]. In IoT applications, sensor devices periodically collect information from the surrounding environment and deliver it to the intended destination [5–7]. Sensor devices are generally powered by a battery with limited capacity; thus, most IoT applications suffer from a short device lifetime [8–10]. Energy depletion of certain sensor devices leads to unexpected malfunctions of IoT networks, which incur high maintenance costs [11,12].

Recently, simultaneous wireless information and power transfer (SWIPT) has been considered a promising solution to address this problem because it allows the sensor device to harvest energy from radio frequency (RF) signals and exchange data packets using the harvested energy [13–15]. In addition, SWIPT can be classified into power-splitting (PS)-SWIPT and time-switching (TS)-SWIPT according to the receiver architecture [16–18]. In PS-SWIPT, the receiver splits the received RF signal into two separate signal streams to harvest energy and obtain information at the same time. In TS-SWIPT, the receiver periodically switches its antenna between the energy harvesting and information decoding circuits to harvest energy and obtain information in a time-division manner. The TS-SWIPT is more suitable than PS-SWIPT for implementing IoT networks because of its simpler

receiver architecture [19]. However, PS-SWIPT can be impractical in a real environment because power loss may occur in the power-splitting process.

In the IoT network to which TS-SWIPT is applied (i.e., a TS-SWIPT-enabled IoT network), the sensor device harvests energy from a nearby hybrid access point (HAP) and then transmits data according to its transmission period (TP) [20]. Therefore, in TS-SWIPT-enabled IoT networks, the timeliness requirement must be ensured for both energy harvesting and data transmission. The medium access control (MAC) layer protocol determines when and how long the sensor devices conduct operations for energy harvesting and data transmission in a TS-SWIPT network [21–25]. Therefore, designing a MAC layer protocol specialized for TS-SWIPT-enabled IoT networks that includes heterogeneous sensor devices with different TPs is a challenging issue [26,27].

In the literature, many studies have been conducted to develop an efficient MAC layer protocol for TS-SWIPT-enabled IoT networks. Naderi et al., Kim et al. and Ha et al. focused on a MAC protocol based on a carrier-sense multiple access with collision avoidance (CSMA/CA), in which each sensor device competitively accesses the channel using the binary exponential backoff algorithm [28–30]. To ensure the timeliness of energy harvesting and data transmission, the authors differentiated the channel access priorities of sensor devices. Specifically, they adjusted the MAC parameters, such as the contention window and inter frame space, depending on the traffic type (i.e., power and data) and residual energy level. However, the CSMA/CA-based approach has intrinsic difficulty ensuring timely energy harvesting and data transmission due to unpredictable backoff delays and collisions between sensor devices. Furthermore, it may cause unnecessary energy consumption due to idle listening and overhearing [31]. In References [32–34], the authors proposed the time-division multiple access (TDMA)-based MAC protocol, in which time is divided into frames consisting of slots of the same length where the slots are allocated to individual sensor devices. Therefore, it can provide a deterministic delay for energy harvesting and data transmission and prevent collisions between sensor devices. However, in a TS-SWIPT-enabled IoT network including heterogeneous sensor devices, the use of a TDMA-based MAC protocol may decrease cell utilization because it generally uses a frame of a fixed length regardless of the different TPs and energy consumption of sensor devices.

In this paper, we propose a time-slotted channel hopping (TSCH) multiple slotframe scheduling (TMSS) protocol to ensure the timeliness of energy harvesting and data transmission for sensor devices with different TPs in TS-SWIPT-enabled IoT networks. To this end, TMSS uses a TSCH MAC standardized on IEEE 802.15.4-2015 as a baseline MAC scheme, which is a TDMA-based MAC protocol and allows the use of multiple concurrent slotframes with different lengths to support various types of traffic [35]. In other words, by managing multiple concurrent slotframes of the TSCH, TMSS supports timely data transmission for traffic from sensor devices transmitted at different periods, while maximizing power efficiency from a MAC layer perspective. The TSCH slotframe consists of several fixed-size cells, where each cell is represented by a time offset and channel offset. In TMSS, the sensor device sets a slotframe length equal to its TP. It estimates the numbers of power and data cells to be allocated in the configured slotframe, then requests cell allocation to the HAP with the slotframe length and numbers of power and data cells. Initially, the HAP maintains one slotframe and, upon receiving a request for cell allocation from a sensor device, it executes the cell-overlapping prevention (COP) algorithm to resolve the cell-overlapping problem with existing cells that are already allocated. After resolving the cell-overlapping problem via the COP algorithm, the HAP responds to the sensor device with a candidate cell list, which includes empty cells in the slotframe of the corresponding sensor device. Upon receiving the response from the HAP, the sensor device determines its power cells and data cells by referring to the cell list. To verify the superiority of TMSS, we conducted experimental simulations and compared the performance of TMSS to that of the TSCH MAC with a single slotframe and the HE-MAC. The results revealed that TMSS obtains an average of a 40.41% and 20.20% lower delay, 61.66% and 15.30% higher effective

throughput and 44.31% and 25.48% higher energy utilization compared to that of legacy TSCH MAC and HE-MAC, respectively.

The rest of this paper is organized as follows. In Section 2, we introduce the related works. Section 3 provides a system model of TMSS. The detailed design of TMSS is described in Section 4. In Section 5, the simulation configuration and results are presented. Finally, Section 6 concludes the paper.

## 2. Related Works

In TS-SWIPT-enabled IoT networks, the MAC protocol plays an important role in providing the timeliness of energy harvesting and data transmission. Recently, a number of MAC protocols have been proposed for TS-SWIPT networks and these can be categorized into two types; CSMA/CA-based approach and TDMA-based approach according to multiple-access methods.

Naderi et al. [28] proposed an RF-MAC which is a CSMA/CA-based MAC protocol for RF energy harvesting sensor networks. In RF-MAC, a sensor device broadcasts the request for energy (REF) packet when its residual energy level reaches a pre-set threshold and waits for the cleared for energy (CFE) pulses from energy transmitters. From the received CFE pulses, the sensor device determines their energy transfer frequencies and the charging duration and responds with an Ack packet. Then, the multiple energy transmitters set their energy transfer frequencies and concurrently transmit the energy signals to the sensor device during the allocated charging duration. To provide the timeliness for energy harvesting, the RF-MAC prioritizes the energy request over data exchange by adjusting CSMA/CA MAC parameters such as slot time, distributed coordination function (DCF) inter-frame space (DIFS) and short inter-frame space (SIFS). Kim et al. [29] proposed an RF energy auto-charging and harvesting (REACH) method, in which the energy transmitters participate in channel contention and automatically transmit the energy signals to charge the sensor devices during the idle time without energy request from sensor devices. In REACH, to provide the timeliness for data transmission of sensor devices, the energy transmitter maintains has a lower channel access priority than sensor devices. Ha et al. [30] proposed a harvest-then-transmit-based modified EDCF MAC protocol, HE-MAC, which manages both energy harvesting and data transmission. In HE-MAC, the sensor device first harvests energy using RF signals from the HAP and then uses the harvested energy to transmit data to the HAP. For timely energy harvesting and data transmission, the sensor devices in the network are classified into different access categories (ACs) depending on whether they have data to be transmitted to HAP. For each AC, the EDCF operational parameters such as arbitration inter-frame space (AIFS),  $CW_{\min}$  and  $CW_{\max}$  are differentiated.

The CSMA/CA-based approaches commonly suffer from unpredictable delays due to control packet exchange, random backoff and frequent collisions and as the number of devices in the network increases, the number of collisions increases, resulting in a long delay. Therefore, they have limitations in ensuring the timeliness for energy harvesting and data transmission in the TS-SWIPT-enabled IoT networks.

Ju et al. [32] proposed a TDMA-based harvest-then-transmit protocol that uses a common-throughput maximization approach to determine the data transmission time differently according to the amount of harvested energy of individual sensor devices. In Reference [32], time is divided into multiple blocks consisting of downlink and uplink periods and in each block, the HAP first broadcasts the energy signals via downlink period and then the sensor device transmits data to the HAP via uplink period. The uplink period is further partitioned into multiple block transmission times, each of which is sequentially allocated to the sensor device. Choi et al. [33] proposed a slotted ALOHA-based energy harvesting MAC protocol for wireless powered communication networks (WPCNs), which uses a frame structure consisting of beacon followed by multiple random access (RA) slots. In Reference [33], the device randomly selects one of the RA slots and harvests energy from the full-duplex HAP until it reaches the selected RA slot. It transmits uplink data at the selected RA and then goes to sleep mode for the rest of the frame. Cho et al. [34]

proposed a hybrid multiple access-based dual harvest-then-transmit (H-DHT) protocol for large-scale WPCNs, which uses a frame structure including the TDMA period and CSMA/CA period. In the former, the HAP allocates timeslots to each device for periodic traffic requiring transmission reliability and the latter is a duration for non-periodic traffic unconstrained from transmission reliability. With regard to energy harvesting, the H-DHT uses the dual-wireless energy transfer (WET) method; The main-WET is performed in the TDMA period and the sub-WET is performed during the idle time in the CSMA/CA period (i.e., space holes).

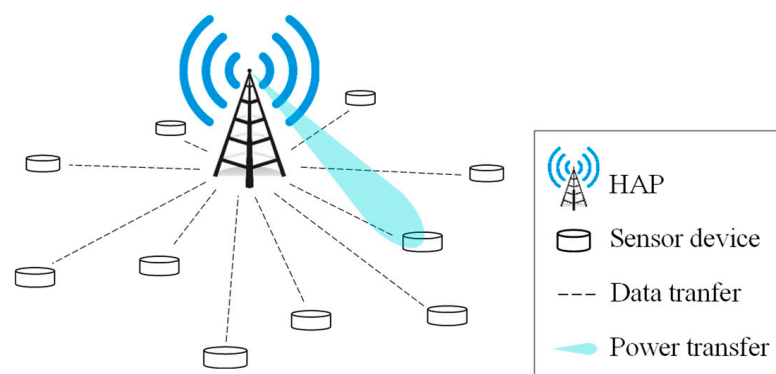
The TDMA-based approach is advantageous to ensure the timeliness for energy harvesting and data transmission of TS-SWIPT networks because it allows devices to use dedicated slots allocated without contentions. Nevertheless, in the TDMA-based approaches, the high complexity required to determine the frame structure for energy harvesting and data transmission of individual devices is a great burden on the operation of MAC protocol. Furthermore, when considering the TS-SWIPT-enabled IoT network environments including heterogeneous sensor devices with different TPs, the use of a fixed size of frame structure greatly degrades channel utilization and throughput performance.

### 3. System Model

To support various TPs of heterogeneous sensor devices in TS-SWIPT-enabled IoT networks, the TMSS uses the multiple concurrent slotframes of TSCH MAC standard, in which power cells and data cells are allocated using the 6top protocol (6P) of the IETF 6TiSCH standard [36]. In the following subsections, we describe the system architecture of TMSS, TSCH slotframe structure and cell allocation procedure in 6P.

#### 3.1. System Architecture

Figure 1 presents the system architecture of the TS-SWIPT-enabled IoT network in TMSS. In the figure, the TS-SWIPT-enabled IoT network consists of a single HAP and several heterogeneous sensor devices. Each device has a different TP and is deployed within the transmission range of the HAP. It is assumed that the sensor device and HAP are equipped with a single omnidirectional antenna and communicate with each other in a half-duplex manner. The power transfer is performed only from the HAP to the sensor device, whereas the data transfer is the opposite.



**Figure 1.** System architecture of TSCH multiple slotframe scheduling (TMSS).

The sensor device monitors the surrounding environment and periodically transmits the data packets to the HAP. The sensor device must first harvest the required energy from a specific RF signal transmitted from the HAP before performing sensing and data transmission. Therefore, at the beginning of the network operation, each sensor device first checks its TP and sets it as its slotframe length. Upon receiving the enhanced beacon (EB) from the HAP, the sensor device measures its received signal strength indicator (RSSI) and then checks the information contained in the EB, such as the timeslot length, number of slotframes, slotframe identifier, length of each slotframe, power and data cells

allocated in each slotframe and transmission (Tx) power. Then, it calculates the amount of expected energy consumption per slotframe and expected energy harvesting per timeslot to estimate the number of power cells required for each slotframe. The amount of expected energy harvesting per timeslot varies depending on the distance between the HAP and sensor device. Thus, the measured RSSI is used to obtain the distance in this estimation process. However, the number of data cells can be simply estimated using the number of data packets, size of the data packet and data rates. The sensor devices transmit the cell allocation request message containing the slotframe length and number of power cells and data cells to the HAP.

The HAP plays a role in coordinating the overall network and provides a power source for energy harvesting of sensor devices, for which it periodically advertises network information, such as the slotframe list and scheduled power and data cells, by broadcasting the EB. The slotframe list maintained by the HAP includes the slotframe of the HAP and the slotframes of the scheduled sensor devices, each of which has a different length and is indexed by a unique identifier. Initially, the HAP maintains one slotframe to broadcast the EB and exchange control packets (e.g., the cell allocation request, cell allocation response, etc.). Whenever receiving a cell allocation request including a new slotframe, the HAP adds the slotframe to the slotframe list. Then, it creates a candidate power cell list and a candidate data cell list by referring to the slotframe list, the information in the received message (i.e., the slotframe length and number of power and data cells) and the existing cells already allocated. The candidate power and data cell lists are transmitted to the sensor device for cell allocation operation of the sensor device.

### 3.2. TSCH Slotframe Structure

Figure 2 illustrates a slotframe structure of the TSCH MAC. In the figure, a slotframe comprises a fixed number of timeslots whose lengths are generally defined to accommodate the maximum size of the data packet. Therefore, the slotframe length is determined according to the number of timeslots. In the slotframe, the pair [timeslot offset, channel offset] is expressed as a cell. The timeslot offset and channel offset denote an index of the timeslot and channel, respectively. The cells are used for the transmission of a device pair. Specifically, the cell can be allocated to a specific device pair to perform their transmissions exclusively (i.e., a dedicated cell). In addition, it can be allocated to be shared by multiple device pairs (i.e., a shared cell). In this case, the devices compete to transmit data packets. The slotframe is repetitive; thus, the allocated cells continue to be maintained in subsequent slotframes [37].

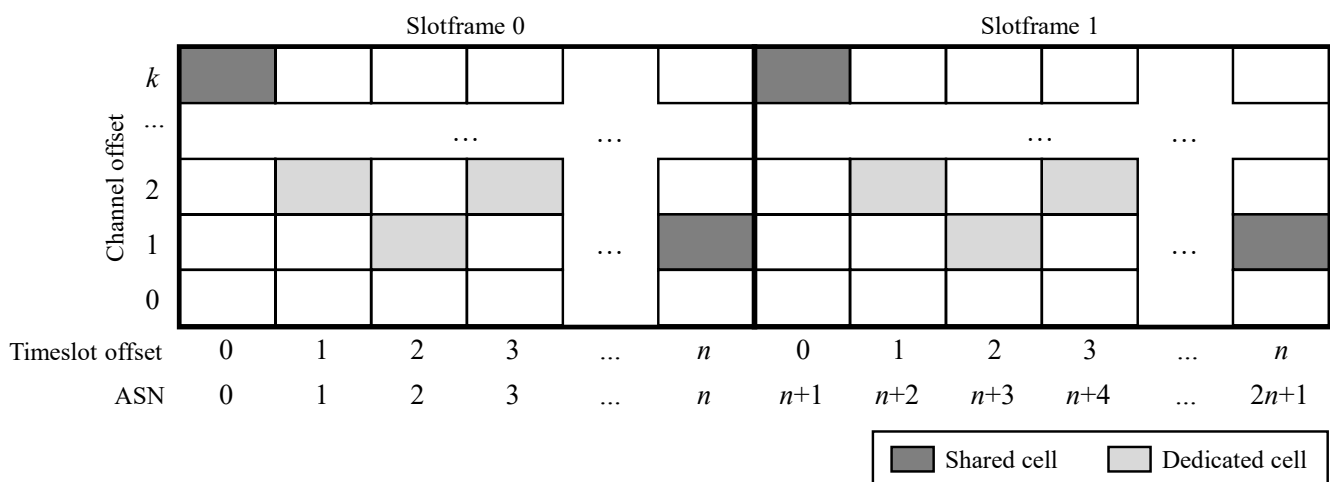


Figure 2. Time-slotted channel hopping (TSCH) slotframe structure.



As mentioned, the TSCH MAC standard allows the use of multiple concurrent slotframes, each of which has a different length. To establish a communication schedule considering the cells allocated to multiple slotframes comprehensively, all slotframes are overlapped. For this, the slotframes are aligned based on the timeslot boundaries and the absolute slot number (ASN) of the first timeslot within the first slotframe is set as zero. The ASN represents the total number of timeslots elapsed from the start of the network. When a cell with a specific ASN is simultaneously used by two or more sensor devices using slotframes with different lengths, cell overlapping occurs. To address this problem, the TSCH MAC standard uses the slotframe handle (i.e., *macSlotframeHandle*) that is an identifier for the slotframe to assign the transmission priority to each slotframe. In other words, the lower *macSlotframeHandle* has a higher transmission priority than the higher *macSlotframeHandle*. However, this approach degrades the delay performance because the sensor device using the lower *macSlotframeHandle* should defer its transmission when cell overlapping occurs. Therefore, to prevent cell overlapping, TMSS introduces the new algorithm COP, which creates candidate power and data cell lists consisting of only empty cells over all slotframes.

### 3.3. 6P Cell Allocation Procedure

The TSCH MAC standard does not specify how to establish and manage the TSCH schedule for communication between devices [38]. The IETF 6TiSCH working group has specified the 6TiSCH operation sublayer (6top) on top of TSCH MAC, in which the minimal 6TiSCH configuration establishes the initial TSCH schedule and the 6top protocol (6P) manages the TSCH schedule by cell allocation procedure [36,39]. For cell allocation within the slotframe, 6P supports two types of cell allocation procedures: two-step and three-step 6P transactions. In the former, the device requesting cell allocation (i.e., the sensor device) creates a candidate cell list, whereas, in the latter, the device receiving the request (i.e., the HAP) creates it. In this paper, the TMSS uses the three-step 6P transaction; thus, the HAP creates the candidate cell list.

Figure 3 displays an operational example of three-step 6P transactions. Type, SeqNum, NumCells and CellList indicate the message type, sequence number of the transaction, number of cells to add and set of cells, respectively. In the figure, Device A transmits a cell allocation request message indicating that it wishes to allocate two cells to Device B. Upon receiving the cell allocation request message, Device B creates a candidate cell list by randomly selecting empty cells within the slotframe and responds with it to Device A. The number of cells in the candidate cell list is set to be greater than the value of NumCells and the CellList of the cell allocation response message is set to be the same as the candidate cell list. Finally, Device A selects two cells from the candidate cell list and informs Device B of the selected cells through a cell allocation confirmation message. The existing three-step 6P transaction is designed to allocate only a single type of cell (i.e., a data cell). In contrast, if the existing 6P transaction is applied to the TS-SWIPT-enabled IoT network using both the power cell and data cell as is, the number of exchanged control messages increases rapidly, resulting in high overhead. Therefore, the TMSS uses a modified three-step 6p transaction to allocate power cells and data cells simultaneously.

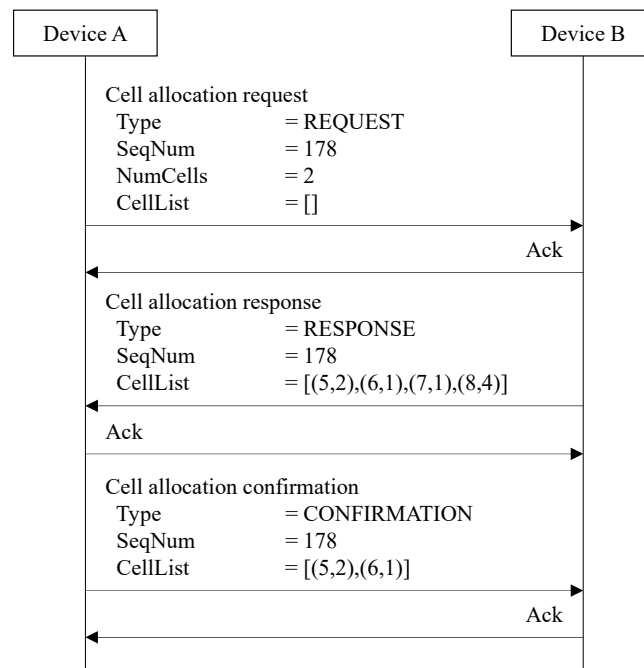


Figure 3. Example of three-step 6P transaction.

#### 4. Design of TMSS

In this section, we describe the design of TMSS. Figure 4 illustrates the overall operation of TMSS for a pair of devices: the HAP and sensor device. To allocate both power and data cells within the slotframe in TS-SWIPT-enabled IoT networks, the TMSS uses the modified three-step 6P transaction, in which NumPwrCells, NumDataCells, PwrCellList and DataCellList indicate the number of power cells to add, number of data cells to add, set of power cells and set of data cells, respectively. Moreover, TMSS uses the SlotframeLength field in the cell allocation request message, which represents the length of the slotframe requested for cell allocation, to support the use of TSCH multiple concurrent slotframes.

The sensor device first checks its TP and sets it as its slotframe length ( $L_{req}$ ). Then, it waits for the EB broadcasted from the HAP. Upon receiving the EB from the HAP, the sensor device measures the RSSI, from which the distance between the HAP and sensor device,  $d$  is calculated using Equation (1) [40].

$$\log(d) = \frac{-RSSI_d + RSSI_1}{10\alpha}, \quad (1)$$

where  $RSSI_d$  is the RSSI measured by the sensor device,  $RSSI_1$  is the reference RSSI measured at a distance of 1 m and  $\alpha$  is the path-loss parameter.

Then, the sensor device estimates the number of power cells and data cells for allocation within its slotframe. Figure 5 illustrates the timing diagram for the data transmission from the sensor device to the HAP, where TsTxOffset and TsRxAckDelay are the time intervals for preparing the transmission of the data packet and the reception of acknowledgement (Ack), respectively [41]. In these intervals, the sensor device is in the idle state.

To obtain the number of power cells to be added in the configured slotframe ( $N_{pwrCells}$ ), the sensor device first calculates the amount of expected energy consumption per slotframe ( $E_{slotframe}$ ), which is equal to the sum of the amount of energy consumed for sensing and data transmission per slotframe (i.e.,  $E_{slotframe} = E_{sensing} + E_{transmission}$ ). It is assumed that  $E_{sensing}$  is predefined as a constant value by the application of the sensor device. The  $E_{transmission}$  is given by Equation (2):

$$E_{transmission} = N_{pkt} \left[ \frac{Size_{pkt}Q_{tx} + Size_{ack}Q_{rx}}{Data\ rate} + (TsTxOffset + TsRxAckDelay)Q_{idle} \right], \quad (2)$$

where  $N_{pkt}$  is the number of data packets to be transmitted during a slotframe,  $Size_{pkt}$  and  $Size_{ack}$  are the size of the data packet and Ack and  $Q_{tx}$ ,  $Q_{rx}$  and  $Q_{idle}$  are the amounts of energy consumption per second in the Tx, Rx and idle states, respectively. Then, the sensor device calculates the amount of expected energy harvesting per timeslot ( $E_{timeslot}$ ). In the case of power transfer, the HAP waits for the  $TsTxOffset$  from the start of a timeslot and then transmits the RF signal until the timeslot ends. Thus,  $E_{timeslot}$  is calculated as shown Equation (3) [13,42]:

$$E_{timeslot} = (L_{timeslot} - TsTxOffset)Q_s = (L_{timeslot} - TsTxOffset) \frac{\eta P_{tx}}{1 + |d|^\alpha}, \quad (3)$$

where  $L_{timeslot}$  is the timeslot length,  $Q_s$  is the amount of energy harvesting per second,  $\eta$  is the efficiency factor of the energy harvesting process and  $P_{tx}$  is the transmission power of HAP, respectively. Finally,  $N_{pwrCells}$  is obtained using Equation (4):

$$N_{pwrCells} = \left\lceil \frac{E_{slotframe}}{E_{timeslot}} \right\rceil, \quad (4)$$

where  $\lceil \cdot \rceil$  represents the ceiling function that maps an input variable to the least integer greater than or equal to the input variable.

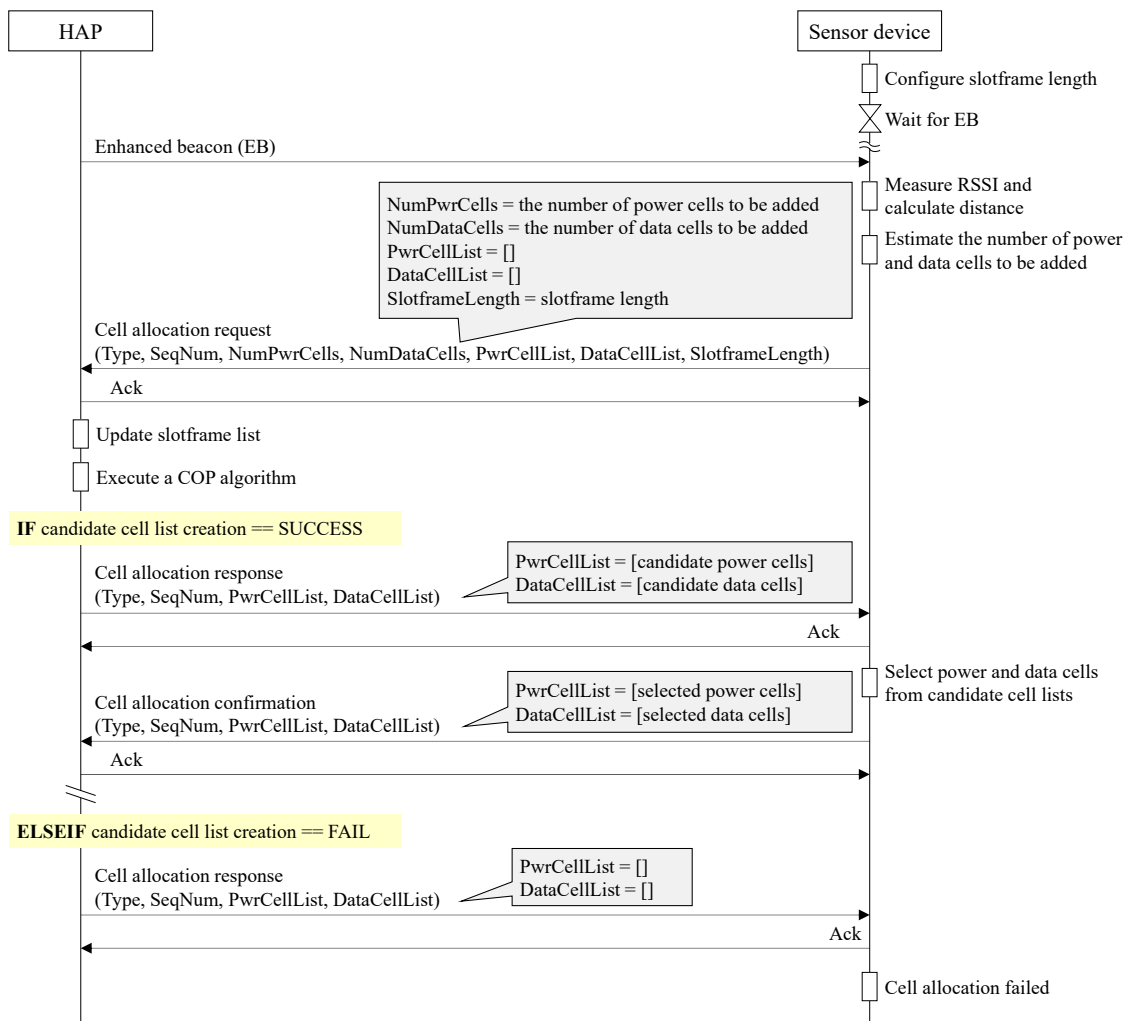


Figure 4. Overall operation of TMSS.



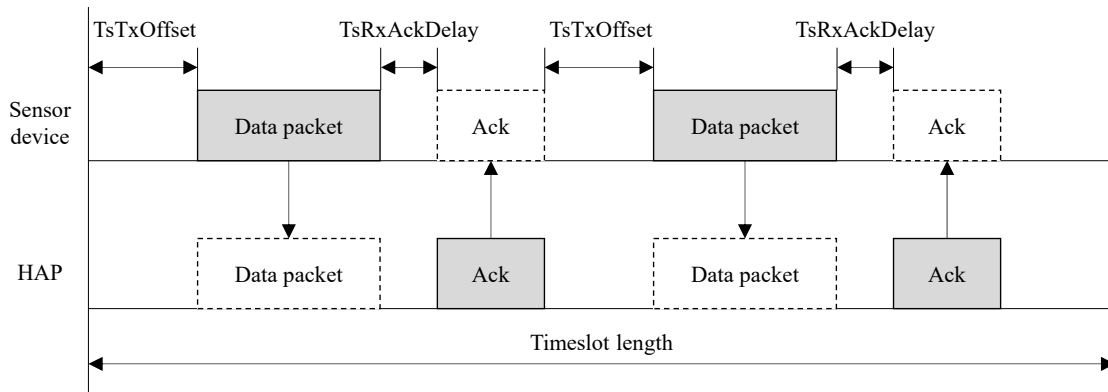


Figure 5. Timing diagram for a timeslot.

After obtaining  $N_{pwrCells}$ , the number of data cells to add in the configured slotframe ( $N_{dataCells}$ ) is estimated. To this end, the sensor device calculates the maximum number of data packets that can be transmitted in one timeslot ( $maxN_{pkt\_timeslot}$ ) via Equation (5) and then obtains  $N_{dataCell}$  using Equation (6):

$$maxN_{pkt\_timeslot} = \left\lfloor L_{timeslot} / \left( \frac{Size_{pkt} + Size_{ack}}{Data\ rate} + TsTxOffset + TsRxAckDelay \right) \right\rfloor, \quad (5)$$

$$N_{dataCells} = \left\lfloor \frac{N_{pkt}}{maxN_{pkt\_timeslot}} \right\rfloor, \quad (6)$$

where  $\lfloor \cdot \rfloor$  represents the floor function that maps an input variable to the greatest integer less than or equal to the input variable. After the estimation process,  $L_{req}$ ,  $N_{pwrCells}$  and  $N_{dataCells}$  are included in the cell allocation request message and the message is transmitted to the HAP.

Upon receiving the cell allocation request message from the sensor device, the HAP updates its slotframe list. If the received cell allocation request message contains  $L_{req}$  that is not contained in the slotframe list, a new slotframe is added to the slotframe list of the HAP. Otherwise, the existing slotframe list is maintained. Initially, the HAP maintains only one slotframe with an identifier of zero to broadcast the EB and exchange control packets. Whenever a new slotframe is added to the slotframe list, it is assigned a new identifier incremented by one. The slotframe list maintains information for the identifier, length and set of allocated power and data cells for each slotframe (i.e., **ID**, **L** and **SetCells**), which are represented as Equations (7)–(9), respectively:

$$\mathbf{ID} = [ID_{(0)}, ID_{(1)}, ID_{(2)}, \dots, ID_{(i)}], \quad (7)$$

$$\mathbf{L} = [L_{(0)}, L_{(1)}, L_{(2)}, \dots, L_{(i)}], \quad (8)$$

$$\mathbf{SetCells} = [\mathbf{Cells}_{(0)}, \mathbf{Cells}_{(1)}, \mathbf{Cells}_{(2)}, \dots, \mathbf{Cells}_{(i)}], \quad (9)$$

where  $ID_{(i)}$ ,  $L_{(i)}$  and  $\mathbf{Cells}_{(i)}$  are the identifier, slotframe length and set of power and data cells for the  $(i + 1)$ th slotframe, respectively.  $\mathbf{Cells}_{(i)}$  is given by Equation (10):

$$\mathbf{Cells}_{(i)} = [Cell_{(i,0)}, Cell_{(i,1)}, Cell_{(i,2)}, \dots, Cell_{(i,j)}], \quad (10)$$

where  $Cell_{(i,j)}$  indicates the  $(j + 1)$ th cell within the  $(i + 1)$ th slotframe, which can be represented by a pair of  $[TO_{(i,j)}, CO_{(i,j)}]$ . Moreover,  $TO_{(i,j)}$  and  $CO_{(i,j)}$  are the timeslot offset and channel offset for the  $(j + 1)$ th cell within the  $(i + 1)$ th slotframe, respectively.

As mentioned, when a certain cell with a specific ASN is allocated to multiple sensor devices, a cell-overlapping problem occurs. To prevent this, the HAP executes the COP algorithm to create the candidate power and data cell lists consisting of empty cells. Algorithm 1 describes the procedure of the COP algorithm in detail. In the algorithm, the HAP creates a virtual slotframe (i.e., a least common multiple (LCM) slotframe) having the same length as the LCM of the lengths of all slotframes included in its slotframe list. In addition, the HAP uses this to discover cells that are commonly empty in all slotframes of its slotframe list. The procedure of the COP algorithm consists of three steps as follows: (1) LCM slotframe creation, (2) empty timeslot discovery and (3) candidate cell list creation. In the first step, an LCM slotframe is created and in the second step, the LCM slotframe is divided into several partitions and the timeslot offsets of empty cells are discovered in all partitions. Finally, in the third step, the candidate power and data cell lists are created by referring to the timeslot offsets of empty cells.

The slotframe list (i.e., **ID**, **L** and **SetCells**) and the given channel list (i.e., **Channel**) are used as the inputs for the COP algorithm. At the beginning of the COP algorithm, the variables (i.e.,  $L_{LCM}$ ,  $numRep$ ,  $N_{partition}$ ,  $index_{empty}$  and  $CellListFlag$ ), which are required to create the candidate power and data cell lists, are initialized. In addition,  $L_{LCM}$  is the LCM slotframe length and  $numRep$  is the number of repetitions of each slotframe in the HAP slotframe list within the LCM slotframe. Moreover,  $N_{partition}$  is the number of partitions for the LCM slotframe, which indicates how many times the slotframe requested by the sensor device is repeated within the LCM slotframe. Additionally,  $index_{empty}$  denotes the index of the elements included in the set of timeslot offsets for empty cells ( $\mathbf{TS}_{empty}$ ). Finally,  $CellListFlag$  indicates whether the candidate cell lists are successfully created.

In the first step, the HAP calculates  $L_{LCM}$  through  $lcm()$ , a function that outputs the LCM for input variables and then creates an LCM slotframe (**LCM**) consisting of  $(L_{LCM}/L_{timeslot})$  cells, as shown in Equation (11):

$$\mathbf{LCM} = \left[ lcm_{(0)}, lcm_{(1)}, lcm_{(2)}, \dots, lcm_{(l)}, \dots, lcm_{(L_{LCM}/L_{timeslot}-1)} \right], \quad (11)$$

where  $lcm_{(l)}$  indicates whether the  $(l + 1)$ th cell within the LCM slotframe has already been allocated another sensor device. Therefore, its value is 1 if, for all slotframes in the HAP slotframe list, a specific cell is already allocated to another sensor device. Otherwise, it is zero.

In the second step, the LCM slotframe is divided into several partitions according to the length of the slotframe requested by the sensor device. Further,  $\mathbf{LCM}_{(l)}$  represents the  $(l + 1)$ th partition in the LCM slotframe and  $extract(\mathbf{INPUT}, x : y)$  is a function that extracts the elements from the  $x$ th column to the  $y$ th column from the **INPUT**. All partitions are overlapped to discover the timeslot offset of the cells with a value of zero (i.e.,  $\mathbf{LCM}_{overlap}$  is created). Then,  $\mathbf{TS}_{empty}$  is created by sequentially listing the discovered timeslot offsets.

In the last step, the HAP compares the number of elements in  $\mathbf{TS}_{empty}$  with the sum of  $N_{pwrCells}$ ,  $N_{dataCells}$  and  $2N_{extraCells}$ . The number of elements in  $\mathbf{TS}_{empty}$  is obtained through  $size()$ , a function that counts the number of elements of an input variable and  $N_{extraCells}$  is the number of additional cells required to create the candidate cell list. When the number of elements in  $\mathbf{TS}_{empty}$  is greater than or equal to the sum of  $N_{pwrCells}$ ,  $N_{dataCells}$  and  $2N_{extraCells}$ , the HAP creates the candidate power cell and data cell lists (i.e.,  $\mathbf{Cells}_{pwr}$  and  $\mathbf{Cells}_{data}$ ) and sets  $CellListFlag$  to TRUE. To perform the power transfer prior to the data transmission, the timeslot offset for the candidate power cells is selected from  $\mathbf{TS}_{empty}$  to be smaller than that for the candidate data cells. In the opposite case, the HAP makes both lists empty and sets  $CellListFlag$  to FALSE. The channel offset for the cells is randomly selected from the given channel list (**Channel**). Then, the algorithm returns  $\mathbf{Cells}_{pwr}$ ,  $\mathbf{Cells}_{data}$  and  $CellListFlag$ .

**Algorithm 1.** Cell-overlapping prevention (COP) algorithm.

---

```

1: INPUT ID, L, SetCells, Channel // Input the slotframe list and channel list
2: INITIALIZE  $L_{LCM}$  to 0, numRep to 0,  $N_{partition}$  to 0,  $index_{empty}$  to 0,  $CellListFlag$  to FALSE // Initialize variables
3: /* ===== Step 1. LCM slotframe creation ===== */
4:  $L_{LCM} \leftarrow lcm(L)$  // Calculate the length of LCM slotframe
5: FOR each slotframe,  $i, i \in [0, n_{slotframe} - 1]$  //  $n_{slotframe}$  is the number of slotframes within the slotframe list
6:   numRep  $\leftarrow L_{LCM}/L_{(i)}$ 
7:   FOR each repetition,  $k, k \in [0, numRep - 1]$ 
8:     FOR each cell,  $j, j \in [0, n_{cell,i} - 1]$  //  $n_{cell,i}$  is the number of cells within the  $i$ th slotframe
9:        $LCM_{(i,j)} \leftarrow TO_{(i,j)} + numRep \times L_{(i)} \leftarrow 1$ 
10:    ENDFOR
11:  ENDFOR
12: ENDFOR
13: /* ===== Step 2. Empty timeslot discovery ===== */
14:  $n_{partition} \leftarrow L_{LCM}/L_{req}$  // Calculate the number of partitions in LCM slotframe
15: FOR each partition,  $l, l \in [0, n_{partition} - 1]$ 
16:    $LCM_{(l)} \leftarrow extract(LCM, l \times (L_{req}/L_{timeslot}) : (l + 1) \times (L_{req}/L_{timeslot}) - 1)$ 
17:    $LCM_{overlap} \leftarrow LCM_{overlap} \cup LCM_{(l)}$ 
18: ENDFOR
19: FOR each timeslot,  $m, m \in [0, L_{req} - 1]$  // Search timeslot of empty cells in  $LCM_{overlap}$ 
20:   IF  $LCM_{overlap}[m] == 0$ 
21:      $TS_{empty}[index_{empty}] \leftarrow m$ 
22:   ENDIF
23:    $index_{empty} \leftarrow index_{empty} + 1$ 
24: ENDFOR
25: /* ===== Step 3. Candidate cell list creation ===== */
26: IF  $size(TS_{empty}) \geq (N_{pwrCells} + N_{dataCells} + 2N_{extraCells})$ 
27:   FOR each empty cell,  $p, p \in [0, N_{pwrCells} + N_{extraCells} - 1]$ 
28:      $Cells_{pwr}[p] \leftarrow [TS_{empty}[p], rand(Channel)]$ 
29:   ENDFOR
30:   FOR each empty cell,  $q, q \in [N_{pwrCells} + N_{extraCells}, N_{pwrCells} + N_{dataCells} + 2N_{extraCells} - 1]$ 
31:      $Cells_{data}[q] \leftarrow [TS_{empty}[q], rand(Channel)]$ 
32:   ENDFOR
33:    $CellListFlag \leftarrow TRUE$ 
34: ELSEIF
35:    $Cells_{pwr} \leftarrow []$ 
36:    $Cells_{data} \leftarrow []$ 
37:    $CellListFlag \leftarrow FALSE$ 
38: ENDIF
39: RETURN  $Cells_{pwr}, Cells_{data}, CellListFlag$ 

```

---

After executing the COP algorithm, the HAP checks  $CellListFlag$ . If  $CellListFlag$  is TRUE, the HAP includes the selected candidate power and data cells in the cell allocation response message and transmits the message to the sensor device. Upon receiving the message, the sensor device randomly selects the required power and data cells from the candidate power and data cells and then transmits the cell allocation confirmation message including the selected cells to the HAP. Meanwhile, when  $CellListFlag$  is FALSE, the HAP transmits the cell allocation response message including an empty candidate cell list to the sensor device. In this case, the sensor device determines that the cell allocation has failed.

## 5. Performance Evaluation

To evaluate the performance of TMSS, the experimental simulation was conducted under the IEEE 802.15.4 physical (PHY)/MAC layers environment. The simulation results for the performance of TMSS were compared to that of the TSCH with a single slotframe

(i.e., legacy TSCH) [35] and the HE-MAC [30] to verify the effectiveness of TMSS. In the following subsections, the simulation setting and results are described in detail.

### 5.1. Simulation Setting

In the simulation, we considered two simulation scenarios in which the TP of the sensor device was set to be shorter or longer (or equal) than the length of the slotframe used for the EB broadcasting and control packet transmission (i.e., the EB slotframe). In both scenarios, the length of the EB slotframe was set to 2 s. Because we set the timeslot length to 10 ms, the number of timeslots in the EB slotframe was equal to 200. In each scenario, three groups of sensor devices were randomly deployed and the sensor devices in each group have the same TP. Different lengths of TPs are used for each group. Specifically, the TP of each group was set to 0.5, 1 and 1.5 s in the first scenario (i.e., the short TP scenario) and the TP of each group was set to 2 s, 2.5 s and 3 s in the second scenario (i.e., the long TP scenario). Each group comprises multiple sensor devices and the number of sensor devices per group varies in the range of [2,20]. We assumed that all sensor devices were placed within the communication range of the HAP. The distance between the sensor device and the HAP (i.e.,  $d$ ) was randomly set from 1 to 4 m. The number of data packets to be transmitted to the HAP in a TP (i.e.,  $N_{pkt}$ ) and the packet size and data rate were set to 5, 125 bytes and 250 kbps, respectively. Thus, the number of data cells allocated to each sensor device is five. Meanwhile, the number of power cells allocated to each sensor device varies depending on  $d$  and TP. The parameters used to estimate the number of power cells, such as the amount of energy consumed in the Tx, Rx and idle states (i.e.,  $Q_{tx}$ ,  $Q_{rx}$  and  $Q_{idle}$ ), the transmission power of the HAP ( $P_{tx}$ ), the path-loss parameter ( $\alpha$ ) and the efficiency factor ( $\eta$ ) were set to 24.11 mW, 19.26 mW, 4.67 mW, 100 mW, 2.7 and 0.65, respectively. The performance of TMSS in terms of delay, effective throughput, energy utilization and accumulated overhead time was compared to that of legacy TSCH and HE-MAC. Moreover, we evaluated the cell utilization performance for TMSS and legacy TSCH. For legacy TSCH, we set the slotframe length of sensor devices in all groups to be the same as the EB slotframe length (i.e., 2 s), regardless of their TP. The number of data packets to be transmitted to the HAP within a slotframe length was set to  $\lceil N_{pkt} / (TP/2) \rceil$ . For HE-MAC, we set all sensor devices in a group to compete with each other to transmit  $N_{pkt}$  packets within the TP length. In HE-MAC, the  $CW_{min}$  and  $CW_{max}$  were set to three and seven and the duration of AIFS and SIFS were set to 50  $\mu$ s and 10  $\mu$ s. The size of request to send (RTS), clear to send (CTS), block Ack request (BAR) and block Ack (BA) packets were set to 160, 112, 200 and 125 bytes, respectively. The simulation was iterated 200 times. The detailed simulation parameters are listed in Table 1.

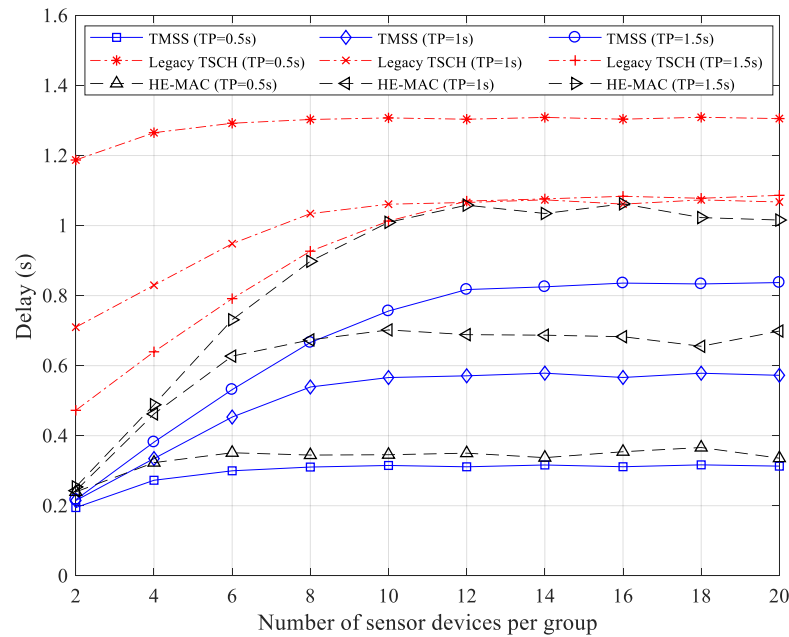
**Table 1.** Simulation parameters.

Parameter	Value	Parameter	Value
PHY/MAC	IEEE 802.15.4	TP	0.5 s–3 s
Number of sensor devices per group	2–20	$d$	1 m–4 m
Data rate	250 kbps	$N_{pkt}$	5
Packet size	125 bytes	$Q_{tx}$	24.11 mW
Ack size	25 bytes	$Q_{rx}$	19.26 mW
TsTxOffset	2120 $\mu$ s	$Q_{idle}$	4.67 mW
TsRxAckDelay	800 $\mu$ s	$P_{tx}$	100 mW
Timeslot length	10 ms	$\alpha$	2.7
Slotframe length	0.5 s–3 s	$\eta$	0.65

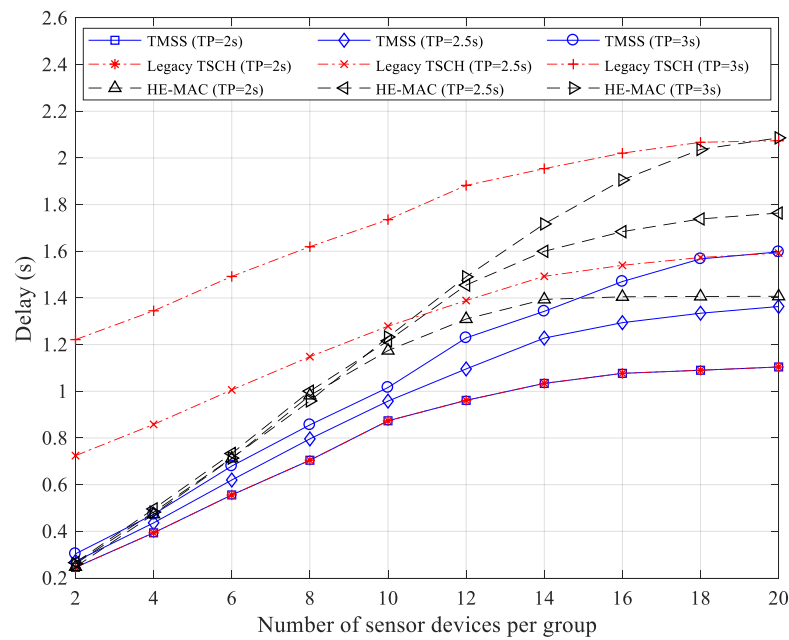
### 5.2. Simulation Results

Figure 6a,b show the delay performance of TMSS for short and long TP scenarios, respectively. In the simulation, the delay indicates the average time it takes a sensor device to deliver  $N_{pkt}$  data packets to the HAP. In both scenarios, the delay of TMSS is shorter than the TP length of each group regardless of the number of sensor devices per group

because the TMSS allows the sensor device to set the slotframe length equal to its TP. Thus, the TMSS ensures the timeliness of energy harvesting and data transmission. However, in the short TP scenario, the legacy TSCH does not ensure the timeliness when the TP is 0.5 s and when the TP is 1 s and the number of sensor devices is eight or more. In this case, some sensor devices are highly likely to select power and data cells after the TP within the slotframe or wait for the next slotframe to transmit data packets because the TP is shorter than the slotframe length (i.e., 2 s).



(a)



(b)

Figure 6. Delay: (a) short transmission period (TP) scenario; (b) long TP scenario.

As mentioned earlier, the HE-MAC is a CSMA/CA-based approach, which is inherently difficult to ensure the timeliness of data transmission because of its random access

feature. Therefore, in HE-MAC, we considered the delays only for sensor devices that completed transmission of  $N_{pkt}$  packets within the TP length (i.e., successful devices), for the convenience of performance comparison. For this reason, as shown in the figures, the delay of HE-MAC is always shorter than the TP length. Nevertheless, the TMSS shows better delay performance than HE-MAC in both scenarios and their difference increases as the number of sensor devices increases due to high contention, until it reaches a certain number. Table 2 lists the average number of successful devices, where  $n_{dev}$  is the number of sensor devices per group. In the table, the number of successful devices for HE-MAC increases as  $n_{dev}$  increases and then decreases again from a certain number. Therefore, the delay of HE-MAC does not increase any more even in high contention situations. Quantitatively, the TMSS achieves 54.32% and 18.87% shorter delays in short TP scenario and 26.49% and 20.87% shorter delays in long TP scenario, compared to legacy TSCH and HE-MAC, respectively.

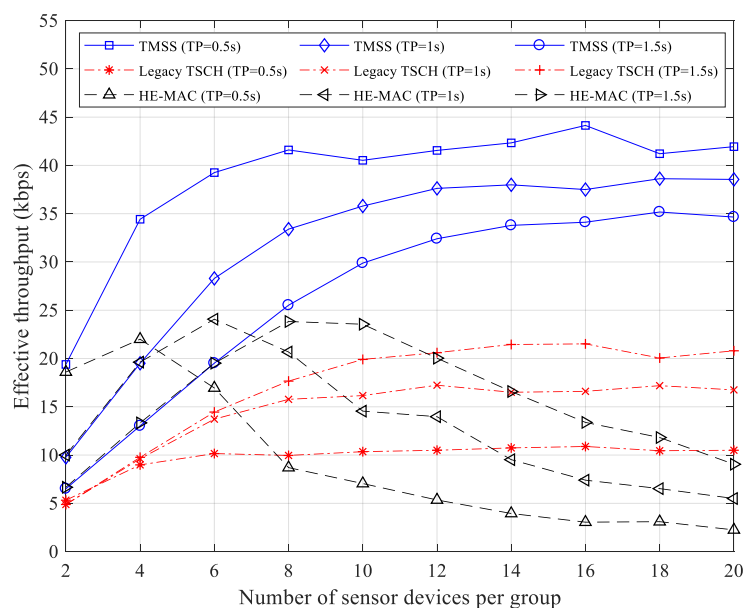
**Table 2.** Average number of successful devices.

	$n_{dev}$	TMSS			Legacy TSCH			HE-MAC		
		0.5 s	1 s	1.5 s	0.5 s	1 s	1.5 s	0.5 s	1 s	1.5 s
Short TP Scenario	2	1.99	2.00	2.00	0.55	1.00	1.50	1.86	2.00	2.00
	4	3.53	4.00	4.00	0.92	1.97	3.00	2.20	3.93	4.00
	6	4.02	5.80	6.00	1.04	2.81	4.44	1.70	4.82	5.86
	8	4.26	6.84	7.84	1.02	3.23	5.43	0.87	4.14	7.15
	10	4.15	7.33	9.18	1.06	3.31	6.12	0.71	2.91	7.07
	12	4.26	7.71	9.96	1.08	3.53	6.33	0.54	2.80	6.01
	14	4.34	7.78	10.38	1.10	3.38	6.59	0.40	1.91	4.98
	16	4.52	7.68	10.48	1.12	3.40	6.61	0.31	1.48	4.02
	18	4.22	7.91	10.81	1.07	3.52	6.16	0.31	1.31	3.55
20	4.30	7.90	10.65	1.08	3.43	6.39	0.23	1.10	2.72	
	$n_{dev}$	TMSS			Legacy TSCH			HE-MAC		
		2 s	2.5 s	3 s	2 s	2.5 s	3 s	2 s	2.5 s	3 s
Long TP Scenario	2	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	4	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
	6	6.00	6.00	6.00	6.00	6.00	6.00	5.99	6.00	6.00
	8	8.00	8.00	8.00	8.00	7.98	7.94	7.91	8.00	8.00
	10	9.85	9.97	10.00	9.85	9.64	9.78	9.32	9.97	10.00
	12	11.39	11.76	11.96	11.39	11.36	11.04	9.62	11.67	11.94
	14	12.36	13.24	13.70	12.36	11.44	11.34	9.18	12.56	13.73
	16	12.58	14.24	15.27	12.58	12.88	11.98	7.65	11.63	14.92
	18	12.94	14.69	15.71	12.94	12.66	12.36	6.52	10.38	15.14
20	13.38	14.51	16.78	13.38	12.80	12.64	5.61	9.26	13.67	

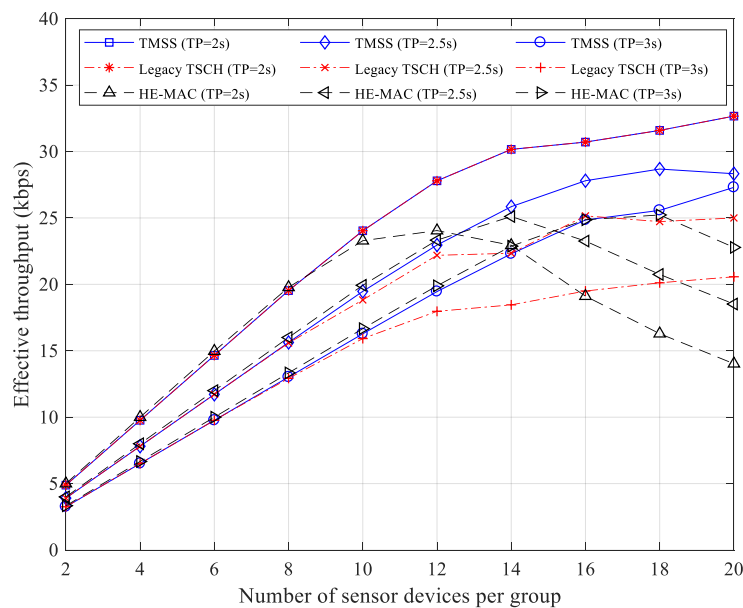
Figure 7a,b show the effective throughput of successful devices for short and long TP scenarios, respectively. TMSS exhibits higher effective throughput compared to legacy TSCH and HE-MAC. In TMSS, the sensor device sets its slotframe length equal to the TP length and thus it can maintain a larger number of successful devices compared to others. Overall, the shorter the TP length, the higher the throughput because the sensor devices transmit data packets more frequently. However, in legacy TSCH of the short TP scenario, the sensor devices are highly likely to select data cells after the TP within the slotframe because the TP length is shorter than the slotframe length (i.e., 2 s). Therefore, as shown in Figure 7a, in legacy TSCH, as the TP length decreases, the number of successful devices decreases and as a result, the effective throughput decreases. For each TP, as the number of sensor devices per group increases, the throughput of TMSS and legacy TSCH increases and then remains constant from a certain point. On the other hand, the throughput of HE-MAC increases and then decreases from a certain point in time. This is because the average number of successful devices of HE-MAC decreases due to high



contention, as shown in Table 2. Quantitatively, the effective throughput of TMSS is 61.66% and 75.30% higher than that of legacy TSCH and HE-MAC, respectively.



(a)

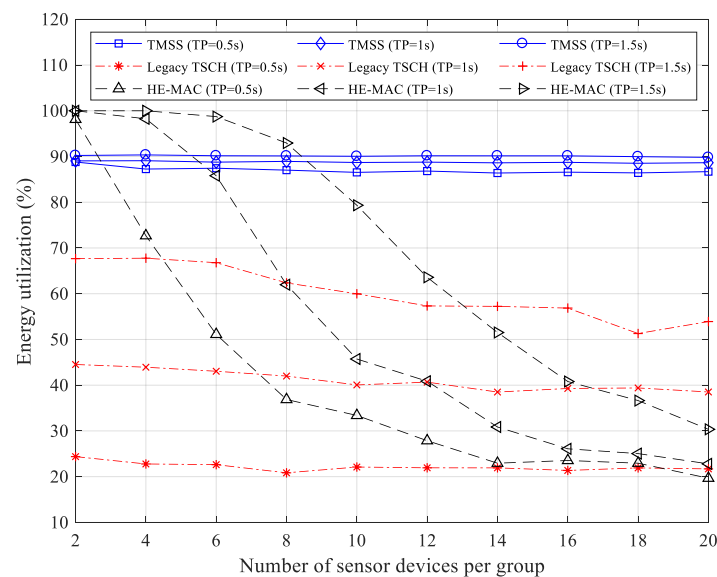


(b)

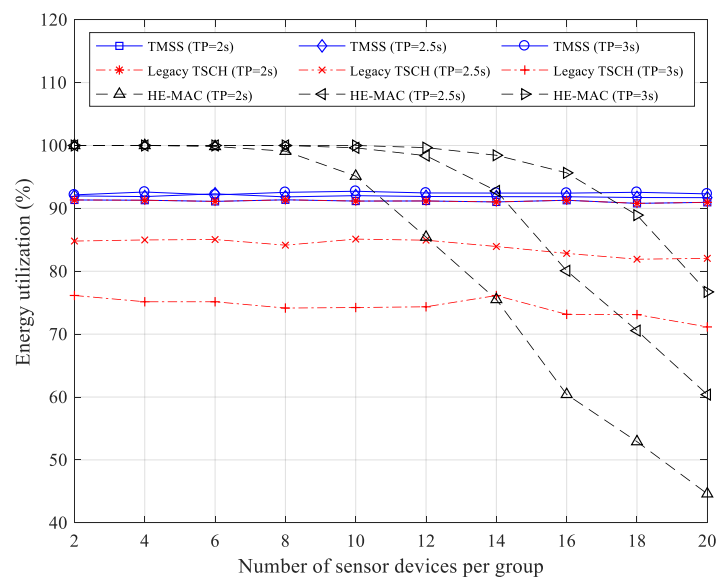
Figure 7. Effective throughput: (a) short TP scenario; (b) long TP scenario.

Figure 8a,b show the energy utilization for short and long TP scenarios, respectively. Energy utilization indicates the ratio of the amount of energy consumed by successful devices to the amount of energy harvested during the TP. TMSS exhibits approximately 90% energy utilization, regardless of the TPs and the number of sensor devices per group. In TMSS, all power and data cells allocated in a slotframe are used for energy harvesting and data transmission, respectively. However, the amount of energy harvested is slightly greater than the amount of energy consumed because it does not exactly match the amount of energy required for data transmission. In the figures, the energy utilization of legacy TSCH decreases as the difference between the slotframe length (i.e., 2 s) and the TP

length increases. In legacy TSCH, the power and data cells are independently allocated in a slotframe and when the TP is shorter than 2 s, even if the energy is harvested in the power cells within the TP, the data cells after the TP may not be used by successful devices. Conversely, when the TP is longer than 2 s, the energy harvested in the power cells within the TP may be wasted by data transmission in the data cells allocated to the second slotframe in the TP. In HE-MAC, energy harvesting and data transmission are performed consecutively; thus, it exhibits high energy utilization when the number of sensor devices per group is small. However, as the number of sensor devices per group increases, the number of successful devices decreases due to the data transfer delay in high contention situations and accordingly, its energy utilization sharply decreases. On average, the TMSS exhibits 44.28% and 25.48% higher energy utilization compared to legacy TSCH and HE-MAC, respectively.



(a)



(b)

Figure 8. Energy utilization: (a) short TP scenario; (b) long TP scenario.

Figure 9 shows the accumulated overhead time when the TP is 1 s. The accumulated overhead time refers to the total time required for network configuration from the time

when the TS-SWIPT network is deployed. As mentioned earlier, TMSS uses a TSCH MAC as a baseline MAC scheme, which is a typical TDMA-based MAC protocol. Thus, like legacy TSCH, it involves the initial overhead required to schedule power and data cells within a slotframe, which can be a weak point of TMSS. However, the network configuration of TMSS is performed only once at the time of deployment and thus, its overhead can be neglected when considering a long network operation time. In the figure, TMSS maintains a similar level of overhead time as legacy TSCH over time because it uses a modified three-step 6P transaction for cell allocation procedure without changing the number and size of exchanged messages. On the other hand, in HE-MAC, the overhead time continuously increases over time because the individual sensor devices compete to access the channel without any schedule information whenever necessary. Moreover, as the number of sensor devices increases, the high overhead is accrued for resolving collisions. The accumulated overhead time of TMSS becomes shorter than that of HE-MAC from 45 s and 56 s in simulation time when the total number of sensor devices is 60 and 30, respectively.

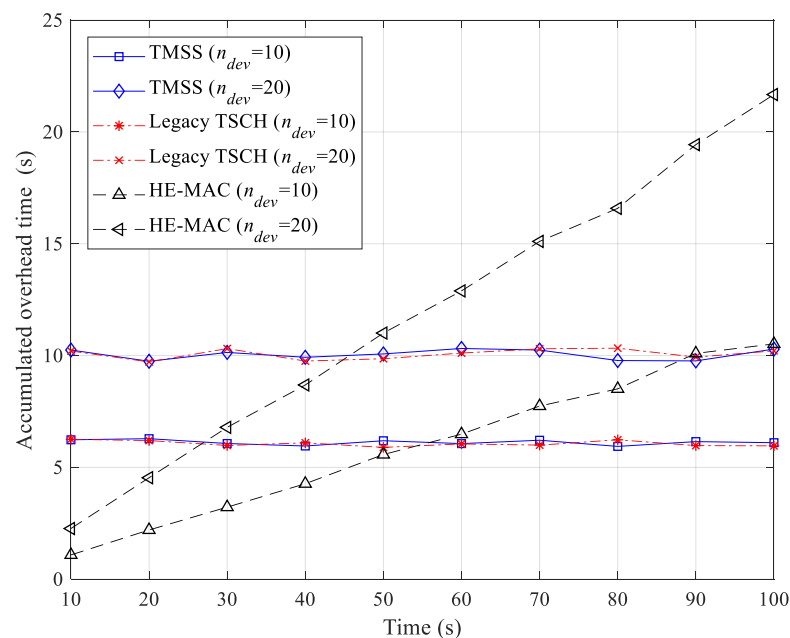
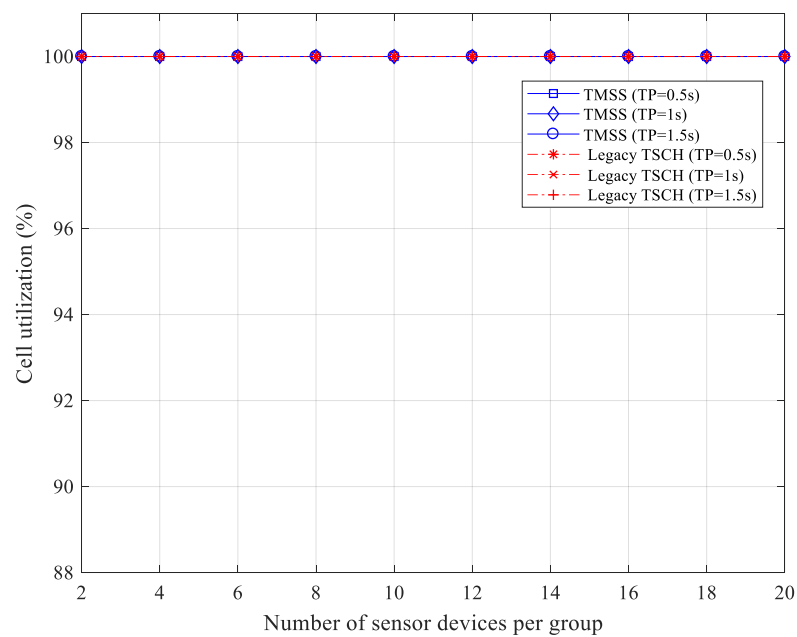
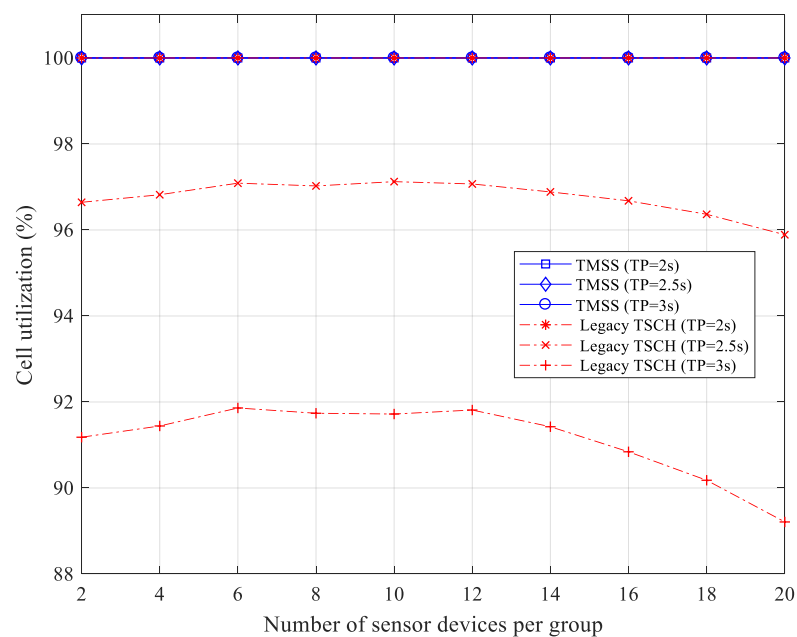


Figure 9. Accumulated overhead time.

Figure 10a,b show the cell utilization of TMSS and legacy TSCH for short and long TP scenarios, respectively. The cell utilization indicates the ratio of the number of cells used for energy harvesting and data transmission to the total number of cells allocated to the sensor devices. The higher the cell utilization, the less cells are wasted within the slotframe, thus the cell utilization is an important indicator of cell scheduling efficiency for TSCH MAC. The cell utilization of TMSS is 100% in all scenarios because the cells are allocated to the slotframe considering the TP of the sensor devices. That is, all cells allocated to sensor devices are used for energy harvesting or data transmission without exception. However, the cell utilization of legacy TSCH decreases when the TP is longer than the slotframe length (i.e., the EB slotframe length). Specifically, in the long TP scenario, the sensor devices using the legacy TSCH should add at least two cells to the slotframe for energy harvesting and data transmission. In this case, several data cells may not be used for data transmission because the allocated data cells are repeated in a slotframe length shorter than the TP. Moreover, even if the sensor device does not transmit data packets in the slotframe, energy harvesting is performed periodically, thus unnecessary energy harvesting may occur. In the long TP scenario, the cell utilization of TMSS is 6.44% higher than that of legacy TSCH.



(a)



(b)

**Figure 10.** Cell utilization: (a) short TP scenario; (b) long TP scenario.

## 6. Conclusions

In this paper, we presented a TMSS protocol that configures a TSCH slotframe and allocates power and data cells to support the timeliness of energy harvesting and data transmission operations of sensor devices with different TPs. To this end, we used the cell allocation procedure, applying a modified three-step 6P transaction. In TMSS, the sensor device sets a slotframe length equal to its TP and estimates the numbers of power and data cells to be allocated in the configured slotframe and then requests cell allocation to the HAP. Upon receiving a request from a sensor device, the HAP executes a COP algorithm to resolve the cell-overlapping problem and responds to the sensor device with a candidate cell list. Upon receiving the response from the HAP, the sensor device determines its

power and data cells by referring to the received cell list. To verify the superiority of TMSS, the experimental simulation was conducted in short and long TP scenarios and the performance of TMSS was compared with that of legacy TSCH and HE-MAC. The results revealed that TMSS ensures the timeliness for energy harvesting and data transmission and improves the network performance in terms of delay, effective throughput, energy utilization and cell utilization. On average, TMSS achieved 40.41% and 20.20% lower delay, 61.66% and 15.30% higher effective throughput and 44.31% and 25.48% higher energy utilization compared to legacy TSCH MAC and HE-MAC, respectively. Furthermore, in the long TP scenario, TMSS exhibited 6.44% higher cell utilization compared to legacy TSCH.

**Author Contributions:** Conceptualization, J.-H.K. and E.-J.K.; methodology, D.K., J.-H.K. and E.-J.K.; software, D.K., J.-H.K. and E.-J.K.; validation, J.-H.K. and E.-J.K.; formal analysis, J.-H.K. and E.-J.K.; investigation, J.-H.K. and E.-J.K.; resources, E.-J.K.; data curation, J.-H.K.; writing—original draft preparation, D.K., J.-H.K. and E.-J.K.; writing—review and editing, D.K., J.-H.K. and E.-J.K.; visualization, J.-H.K.; supervision, D.K. and E.-J.K.; project administration, D.K.; funding acquisition, E.-J.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea government (MOTIE) (20001056, Development of Low-Power Massive MISO SWIPT for IoT Wearable).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tran-Dang, H.; Krommenacker, N.; Charpentier, P.; Kim, D.-S. Toward the Internet of Things for Physical Internet: Perspectives and Challenges. *IEEE Internet Things J.* **2020**, *7*, 4711–4736. [[CrossRef](#)]
2. González-Zamar, M.-D.; Abad-Segura, E.; Vázquez-Cano, E.; López-Meneses, E. IoT Technology Applications-Based Smart Cities: Research Analysis. *Electronics* **2020**, *9*, 1246. [[CrossRef](#)]
3. Mehmood, F.; Ullah, I.; Ahmad, S.; Kim, D.-H. A Novel Approach towards the Design and Implementation of Virtual Network Based on Controller in Future IoT Applications. *Electronics* **2020**, *9*, 604. [[CrossRef](#)]
4. Ang, K.L.M.; Seng, J.K.P. Application specific Internet of Things (ASIoTs): Taxonomy, applications, use case and future directions. *IEEE Access* **2019**, *7*, 56577–56590. [[CrossRef](#)]
5. Lee, S.K.; Bae, M.; Kim, H. Future of IoT networks: A survey. *Appl. Sci.* **2017**, *7*, 1072. [[CrossRef](#)]
6. Samuel, A.; Sipes, C. Making Internet of Things Real. *IEEE Internet Things Mag.* **2019**, *2*, 10–12. [[CrossRef](#)]
7. Wen, Z.; Yang, R.; Garraghan, P.; Lin, T.; Xu, J.; Rovatsos, M. Fog orchestration for internet of things services. *IEEE Internet Comput.* **2017**, *21*, 16–24. [[CrossRef](#)]
8. Mishu, M.K.; Rokonuzzaman, M.; Pasupuleti, J.; Shakeri, M.; Rahman, K.S.; Hamid, F.A.; Tiong, S.K.; Amin, N. Prospective Efficient Ambient Energy Harvesting Sources for IoT-Equipped Sensor Applications. *Electronics* **2020**, *9*, 1345. [[CrossRef](#)]
9. Khan, P.W.; Abbas, K.; Shaiba, H.; Muthanna, A.; Abuarqoub, A.; Khayyat, M. Energy Efficient Computation Offloading Mechanism in Multi-Server Mobile Edge Computing—An Integer Linear Optimization Approach. *Electronics* **2020**, *9*, 1010. [[CrossRef](#)]
10. Abd-Elmagid, M.A.; Kishk, M.A.; Dhillon, H.S. Joint energy and SINR coverage in spatially clustered RF-powered IoT network. *IEEE Trans. Green Commun. Netw.* **2018**, *3*, 132–146. [[CrossRef](#)]
11. Sabovic, A.; Delgado, C.; Subotic, D.; Jooris, B.; De Poorter, E.; Famaey, J. Energy-Aware Sensing on Battery-Less LoRaWAN Devices with Energy Harvesting. *Electronics* **2020**, *9*, 904. [[CrossRef](#)]
12. Al-Kadhimi, H.M.; Al-Raweshidy, H.S. Energy efficient and reliable transport of data in cloud-based IoT. *IEEE Access* **2019**, *7*, 64641–64650. [[CrossRef](#)]
13. Perera, T.D.P.; Jayakody, D.N.K.; Sharma, S.K.; Chatzinotas, S.; Li, J. Simultaneous wireless information and power transfer (SWIPT): Recent advances and future challenges. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 264–302. [[CrossRef](#)]
14. Pan, N.; Rajabi, M.; Claessens, S.; Schreurs, D.; Pollin, S. Transmission Strategy for Simultaneous Wireless Information and Power Transfer with a Non-Linear Rectifier Model. *Electronics* **2020**, *9*, 1082. [[CrossRef](#)]
15. Yu, H.; Zhang, Y.; Guo, S.; Yang, Y.; Ji, L. Energy Efficiency Maximization for WSNs with Simultaneous Wireless Information and Power Transfer. *Sensors* **2017**, *17*, 1906. [[CrossRef](#)]
16. Lu, G.; Lei, C.; Ye, Y.; Shi, L.; Wang, T. Energy Efficiency Optimization for AF Relaying with TS-SWIPT. *Energies* **2019**, *12*, 993. [[CrossRef](#)]

17. Jiang, R.; Xiong, K.; Fan, P.; Zhang, Y.; Zhong, Z. Power minimization in SWIPT networks with coexisting power-splitting and time-switching users under nonlinear EH model. *IEEE Internet Things* **2019**, *6*, 8853–8869. [[CrossRef](#)]
18. Pan, G.; Lei, H.; Yuan, Y.; Ding, Z. Performance analysis and optimization for SWIPT wireless sensor networks. *IEEE Trans. Commun.* **2017**, *65*, 2291–2302. [[CrossRef](#)]
19. Na, H.; Lee, C. A full-duplex SWIPT system with self-energy recycling to minimize energy consumption. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 253. [[CrossRef](#)]
20. Sreelakshmy, K.R.; Jacob, L. Simultaneous wireless information and power transfer in heterogeneous cellular networks with underlay D2D communication. *Wirel. Netw.* **2020**, *26*, 3315–3330. [[CrossRef](#)]
21. Le, T.N.; Pegatoquet, A.; Berder, O.; Sentieys, O. Energy-Efficient Power Manager and MAC Protocol for Multi-Hop Wireless Sensor Networks Powered by Periodic Energy Harvesting Sources. *IEEE Sens. J.* **2015**, *15*, 7208–7220. [[CrossRef](#)]
22. Do, N.T.; da Costa, D.B.; Duong, T.Q.; Bao, V.N.Q.; An, B. Exploiting direct links in multiuser multirelay SWIPT cooperative networks with opportunistic scheduling. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 5410–5427. [[CrossRef](#)]
23. Tang, J.; So, D.K.; Shojaeifard, A.; Wong, K.K. Energy efficiency optimization with simultaneous wireless information and power transfer in MIMO broadcast channels. *IEEE Internet Things* **2018**, *5*, 2605–2619. [[CrossRef](#)]
24. Amirinasab Nasab, M.; Shamshirband, S.; Chronopoulos, A.T.; Mosavi, A.; Nabipour, N. Energy-Efficient Method for Wireless Sensor Networks Low-Power Radio Operation in Internet of Things. *Electronics* **2020**, *9*, 320. [[CrossRef](#)]
25. Lee, S.; Azfar Yaqub, M.; Kim, D. Neighbor Aware Protocols for IoT Devices in Smart Cities—Overview, Challenges and Solutions. *Electronics* **2020**, *9*, 902. [[CrossRef](#)]
26. Huang, J.; Xing, C.C.; Wang, C. Simultaneous wireless information and power transfer: Technologies, applications, and research challenges. *IEEE Commun. Mag.* **2017**, *55*, 26–32. [[CrossRef](#)]
27. Do, T.P.; Song, I.; Kim, Y.H. Simultaneous wireless transfer of power and information in a decode-and-forward two-way relaying network. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1579–1592. [[CrossRef](#)]
28. Naderi, M.Y.; Nintanavongsa, P.; Chowdhury, K.R. RF-MAC: A Medium Access Control Protocol for Re-Chargeable Sensor Networks Powered by Wireless Energy Harvesting. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 3926–3937. [[CrossRef](#)]
29. Kim, T.; Park, J.; Kim, J.; Noh, J.; Cho, S. REACH: An efficient MAC protocol for RF energy harvesting in wireless sensor network. *Wirel. Commun. Mob. Comput.* **2017**, 6438726. [[CrossRef](#)]
30. Ha, T.; Kim, J.; Chung, J.M. HE-MAC: Harvest-Then-Transmit Based Modified EDCF MAC Protocol for Wireless Powered Sensor Networks. *IEEE Trans. Wirel. Commun.* **2017**, *17*, 3–16. [[CrossRef](#)]
31. Kim, E.-J.; Youm, S.; Kang, C.-H. Power-Controlled Topology Optimization and Channel Assignment for Hybrid MAC in Wireless Sensor Networks. *IEICE Trans. Commun.* **2011**, *94*, 2461–2472. [[CrossRef](#)]
32. Ju, H.H.; Zhang, R. Throughput Maximization in Wireless Powered Communication Networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 418–428. [[CrossRef](#)]
33. Choi, H.-H.; Shin, W. Slotted ALOHA for Wireless Powered Communication Networks. *IEEE Access* **2018**, *6*, 53342–53355. [[CrossRef](#)]
34. Cho, S.; Lee, K.; Kang, B.; Joe, I. A hybrid MAC protocol for optimal channel allocation in large-scale wireless powered communication networks. *EURASIP J. Wirel. Comm.* **2018**, *2018*, 9. [[CrossRef](#)]
35. IEEE Std. 802.15.4-2015 (Revision of IEEE Std. 802.15.4-2011). *IEEE Standard for Low-Rate Wireless Networks*; IEEE Standards Association: Piscataway, NJ, USA, 2016.
36. Wang, Q.; Vilajosana, X.; Watteyne, T. 6top Protocol (6P). Available online: <https://tools.ietf.org/html/rfc8480> (accessed on 22 September 2020).
37. Hermeto, R.T.; Gallais, A.; Theoleyre, F. Scheduling for IEEE802. 15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey. *Comput. Commun.* **2017**, *114*, 84–105. [[CrossRef](#)]
38. Palattella, M.R.; Watteyne, T.; Wang, Q.; Muraoka, K.; Accettura, N.; Dujovne, D.; Grieco, L.A.; Engel, T. On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks. *IEEE Sens. J.* **2015**, *16*, 550–560. [[CrossRef](#)]
39. Vilajosana, X.; Pister, K.; Watteyne, T. Minimal IPv6 over the TSCH Mode of IEEE 802.15. 4e (6TiSCH) Configuration. Available online: <https://tools.ietf.org/html/rfc8180> (accessed on 22 September 2020).
40. Li, G.; Geng, E.; Ye, Z.; Xu, Y.; Lin, J.; Pang, Y. Indoor Positioning Algorithm Based on the Improved RSSI Distance Model. *Sensors* **2018**, *18*, 2820. [[CrossRef](#)]
41. Vilajosana, X.; Wang, Q.; Chraim, F.; Watteyne, T.; Chang, T.; Pister, K.S. A realistic energy consumption model for TSCH networks. *IEEE Sens. J.* **2013**, *14*, 482–489. [[CrossRef](#)]
42. He, S.M.; Xie, K.; Chen, W.W.; Zhang, D.F.; Wen, J.G. Energy-aware Routing for SWIPT in Multi-hop Energy-constrained Wireless Network. *IEEE Access* **2018**, *6*, 17996–18008. [[CrossRef](#)]