IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 June 2022, accepted 29 June 2022, date of publication 6 July 2022, date of current version 12 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3188862

RESEARCH ARTICLE

6TiSCH Low Latency Autonomous Scheduling for Industrial Internet of Things

NILAM M. PRADHAN[1], (Member, IEEE), **BHARAT S. CHAUDHARI**[1], (Senior Member, IEEE), AND **MARCO ZENNARO**[2], (Senior Member, IEEE)

[1]Department of Electronics and Telecommunication Engineering, Maharashtra Institute of Technology, Pune 411038, India
[2]Abdus Salam International Centre for Theoretical Physics, Science Technology and Innovation Unit, 34151 Trieste, Italy

Corresponding author: Nilam M. Pradhan (nilam.pradhan@mitwpu.edu.in)

ABSTRACT With the advancements in the Internet of Things (IoT), machine-to-machine communication, big-data, and the associated environment, a new model of the Industrial Internet of Things (IIoT) has emerged. The IIoT brings sensors, intelligent machines and tools, instruments, and analytics together for applications like manufacturing, robotics, and many others. Most of these applications require adaptive and autonomous behavior, Quality of Service (QoS), efficient resource allocation, and reservation. One of these crucial challenges is to build and maintain a data communication schedule. Time Slotted Channel Hopping (TSCH) based network operation promises required QoS for low-power applications and enables high reliability. 6TiSCH layer is being developed by standardizing the protocol stack to achieve industrial performance requirements by using IPv6 over IEEE802.15.4e TSCH MAC. 6TiSCH aims to manage the schedule and configure it with the topology and traffic requirements in the industrial environment. This paper proposes a novel low latency autonomous scheduling scheme for the 6TiSCH networks. It generates a segmented schedule for the network where all source nodes can send application data packets to the root node in a single slotframe. The performance of the proposed technique is compared with existing scheduling techniques. Our scheme outperforms the other techniques. The result shows that the latency is reduced up to 41% in comparison with the other scheduling schemes. The proposed scheme has a lower radio duty cycle as the node's ON time is reduced, making it more energy efficient and reliable.

INDEX TERMS 6TiSCH, autonomous scheduling, industrial IoT, latency, RPL, TSCH.

I. INTRODUCTION

A new generation of communication protocols integrates web-based control with industrial operations to meet the industrial performance requirements. Time Slotted Channel Hopping (TSCH) technology enabled industrial performance parameters such as reliability, latency, duty cycle, and effect of multipath-fading by means of different standards like WirelessHART, ISA100.11a, and others. These are widely used in industrial manufacturing processes to provide wireless connectivity between sensors and actuators. The Industrial Internet of Things (IIoT) includes massive data collection and processing using the wireless infrastructure. The device connectivity is a crucial component of IIoT needing deterministic IP interoperability. WirelessHART is the first standard used for industrial wireless communication and is developed based on the IEEE 802.15.4 standard [1]. Many critical challenges in the industrial environment were not addressed by IEEE 802.15.4. WirelessHART has defined these requirements in its data link layer including Time Division Multiple Access (TDMA) model. IEEE 802.15.4e was released in 2012 with an objective to enable time synchronization by using the TSCH technique, which existed in WirelessHART. Many extensions like Low Latency Deterministic Network (LLDN), Deterministic and Synchronous Multichannel Extension (DSME), Radio frequency identification blink (RFID), Asynchronous multi-channel adaptation (AMCA), and Low Energy (LE) were added in IEEE 802.15.4e. Eventually, TSCH became an evolving standard for industrial automation and process control LLNs.

The associate editor coordinating the review of this manuscript and approving it for publication was Vyasa Sai.

WirelessHART builds a schedule in a centralized fashion, however, in IEEE 802.15.4e TSCH it is feasible in a decentralized or autonomously manner that allows improved adaptability and fault tolerance.

However, in today's Internet of Things (IoT) era which involves cloud-based solutions, these technologies are not suitable to connect industrial devices to the Internet Protocol (IP) network. IEEE 802.15.4e TSCH was designed to fits under an IPv6-enabled protocol stack for LLNs enabling stringent reliability and security in the harsh industrial environment. Sensors deployed LLNs operated for years without requiring battery replacement. However, it does not define the strategies to build and manage the communication schedule, allocate resources, impose RPL signaling to maintain the device connectivity, or handle topological changes [2]. It is essential to link the performance of industrial applications with IP-enabled infrastructure [3], [4]. The Internet Engineering Task Force (IETF) 6TiSCH Working Group (WG) is working to bridge this gap with the development of a number of communication protocols [5]. 6TiSCH defines the upper layers over the TDMA mode of the IEEE 802.15.4e standard. 6TiSCH has enabled the adoption of IPv6 in an industrial environment. The WG is lined up with other IFTF groups for operations, management, routing, and security.

6TiSCH schedule consists of cells indicated by [slotOffset, channelOffset] which is used by each device for transmission, reception, or both. These schedules can be static or dynamic, and it is created and managed by means of centralized, distributed, or autonomous techniques. Centralized scheduling techniques are most suited if latency is the critical performance parameter. In these techniques, the complete topology information is need to be maintained by using a central unit called Path Computation Element (PCE) for each network. The single unit is managing the joining and connectivity information of all nodes in the network. Control message traffic in such a dense network is significantly high. Further, any topology or traffic change requires a higher number of packets exchange with the PCE element. Also, centralized networks suffer from a single point of failure problem. If the centralized control unit fails, the whole network goes down. Centrally managed networks are more suitable for static networks however, difficult to scale in large configurations. On the contrary, in distributed scheduling techniques, the network schedule is managed by a distributed means. Every node after joining and synchronizing with the network continuously negotiates with identified preferred parent node to add, delete or relocate scheduling cells. Recently, autonomous scheduling techniques are proposed where nodes in the network build and maintain their TSCH schedule based on the knowledge of a neighbour node without any signaling overhead. There is no central unit to manage the schedule. Therefore, autonomous scheduling techniques reduce control and negotiation overhead.

As a special case of IoT, many challenging issues are needed to be addressed in the Industrial IoT (IIoT) [6]–[8], like providing interoperability between interconnected devices, enabling adaptive and autonomous behaviour, providing privacy and security, achieving timely and reliable transmission, resources reservation, reducing latency, and jitter, etc. A very challenging objective is to build and manage the schedule to satisfy the Quality of Service (QoS) requirements in the IIoT. This paper presents a novel scheduling technique for IIoT networks. It uses a slotframe segmentation approach to send data packets from any source node to the root in a minimum number of timeslots. Slotframe is designed, where, consecutive segments along the path are selected for the transmission. Also, a different number of segments are considered for analysis. The proposed technique gives a significant improvement in latency after selecting the number of segments equal to the depth of the network. The rest of the paper is arranged as follows. Section II includes the background of the 6TiSCH evolution. In section III, work related to different scheduling techniques is discussed. Section IV introduces the proposed autonomous scheduling scheme. In section V, the performance evaluation of the proposed scheduling technique is discussed. Finally, the conclusion of the paper is discussed in section VI.

II. BACKGROUND

6TiSCH has been evolved with TSCH to use IP network for industrial applications as follows:

A. TIME SLOTTED CHANNEL HOPPING (TSCH)

The IEEE802.15.4e standard has been defined as an amendment to the IEEE802.15.4 Medium Access Control (MAC) protocol. The standard is enhanced in terms of several MAC behaviours supporting deterministic communication. It includes TSCH, DSME, and LLDN [9]. DSME boosts the basic IEEE 802.15.4 to enhance the QoS that meets stringent latency, reliability, and scalability requirements for applications with very low data rates. It enables multisuperframe, Group acknowledgment, distributed beacon scheduling, and channel diversity modes for the network. LLDN is developed to make robust communication in the applications that handles critical data. It is a star topology networking technique providing more determinism and making it suitable for centrally controlled networks. TSCH was proposed for industrial automation to connect a variety of low-power devices in IoT applications with stringent requirements. Single channel solutions cannot achieve high reliability as these environments include massive deployment of nodes that causes multi-path fading and interference. TSCH MAC uses time synchronization for industrial operation determinism and channel hopping to reduce the multipath fading effect. There is always a possibility of packet drop due to channel interference and the noisy industrial environment. IEEE 802.15.4e has enabled operation solutions for Low Power Lossy Networks (LLNs) to work in an industrial environment with stringent operating conditions as well as reliability, availability, and security requirements. The LLN protocol stack has allowed new application fields to operate in industrial environments, like home automation and smart

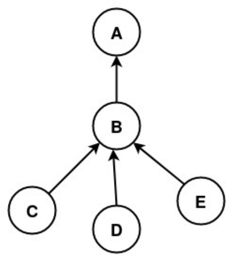

FIGURE 1. Destination oriented directional acyclic graph.

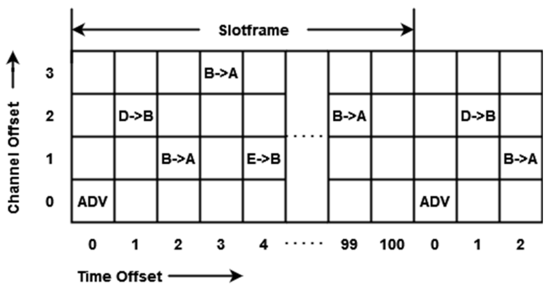

FIGURE 2. TSCH schedule.

cities [10]. In resource constrained IIoT environment, the available bandwidth for communication is low and challenged. The requirement for TSCH MAC is to reduce its bandwidth occupancy to have minimal overhead. It utilizes as fewer channels as possible while improving interoperability with existing technologies. Hence, TSCH uses 4 active channels that deliver a QoS service requirement with less wireless interference.

B. ROUTING PROTOCOL FOR LOW-POWER AND LOSSY NETWORKS (RPL)

RPL is a routing protocol for Wireless Sensor Network (WSN). It was developed to support WSN with IPv6 connectivity and to enable multi-hop routing within resource-constrained devices with restricted computational power, low data rate, and limited memory [11], [12]. To address these challenges in dense LLNs, RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) consisting of one root node and uses a packet forwarding path from every leaf node to the root node as shown in Fig. 1. To satisfy resource constraints it handles packet processing and routing optimization objectives separately. RPL Objective Function (OF) controls and maintains routing by using different messages such as DODAG Information Object (DIO), Destination Advertisement Object (DAO), and Destination Advertisement Object Acknowledgement (DAO-ACK). Every node sends a DIO advertising to all its neighbour with essential parameters that specify the status of the DODAG. It carries information about RPL rank, metric, DODAG preference, version number, and so on.

The DAO message carries destination information from the source to the root creating a downward route. For downward point-to-point or point-to-multipoint communication, RPL supports two operation modes; storing mode and non-storing mode. In the storing mode, packets are transmitted from the source in an upward direction to a common ancestor and then routed down to the destination. Whereas, in non-storing mode, packets are forwarded from the source to root and then routed down to the destination. After receiving the DAO message receiver responds with a DAO-ACK message. In an adaptive scenario, new nodes join the network by listening to the enhanced beacon frame from the other nodes and selecting one of them as a parent node. After joining, the node acquires keying material and other network configuration settings from the parent using the RPL control frame.

TSCH divides communication time into timeslots with a fixed size of 10 ms, sufficiently long to send a MAC and to receive an acknowledgement as shown in Fig. 2. These timeslots are grouped in a slotframe and repeated over time. TSCH devices use a common communication schedule matrix with rows equal to the number of available frequencies and are indexed as ChannelOffsets, and columns indicate timeslots equivalent to communication duration, indexed as TimeOffsets. The ChannelOffset is converted into a communication frequency using a specific conversion function. Pairs of neighbours hop between the different available frequencies when communicating. Time synchronization achieves high reliability whereas channel hopping reduces interference and multipath fading.

C. 6TiSCH

In smart factory applications with large-scale deployment, sensors are networked together using IEEE802.15.4e TSCH MAC because of its attractive features of supporting reliable and ultra-low power LLNs. With the evolution of IIoT, there was a need to bring the IPv6-enabled upper stack to industrial low-power TSCH MAC. Hence, 6TiSCH WG was created by the IETF defining the protocol suite over the IEEE802.15.4e TSCH MAC. The 6TiSCH stack is rooted in the TSCH MAC layer and IEEE802.15.4 physical layer. It uses lightweight Constrained Application Protocol (CoAP), RPL routing layer, and IPv6 based Low Power Wireless Personal Area Networks (6LoWPAN) with some amendments for reliable, scalable, and deterministic mesh network across the backbone. CoAP provides the connectivity of low-power constrained devices to the internet. Adaptation and header compression mechanism in 6LoWPAN allows transmission of IPv6 datagrams into IEEE 802.15.4e MAC frame of 127 bytes. 6TiSCH manages the schedule and tries to match it with the topology and traffic needs of the network. It standardizes the missing component to achieve the industrial performance requirements by using IPv6 over IEEE802.15.4e TSCH MAC. This development has addressed many challenges while using IPv6 for LLNs by supporting security management and efficiency. 6TiSCH WG has defined the 6TiSCH Operation (6top) sublayer that allows management entities to manage the schedule for seamless network operation. 6top maintains the reachability information in a neighbour table. It includes the information about packet transmission and reception time, the number of packets exchanged with its neighbour, cell performance, and link quality. This sublayer sends commands to upper layers to implement QoS parameters at the MAC layer while hiding the scheduling complexity from the upper layers. It monitors underperforming cells and reschedules them so that deterministic behaviour can be maintained. The 6TiSCH architecture defines four ways of scheduling: static scheduling, neighbour-to-neighbour scheduling, remote monitoring and schedule management, and hop-by-hop scheduling. The performance parameters of the LLNs such as latency, average packet delivery, energy consumption, and throughput depending on the approach of designing a transmission/reception schedule.

III. RELATED WORK

Different 6TiSCH scheduling techniques have been developed for deterministic LLN networks based on TSCH that build and manage the schedules [13]–[26]. 6TiSCH Minimal Scheduling Function (MSF) [27] is a simple scheduling function under standardization by IETF 6TiSCH WG. It manages the call to 6top Protocol (6P) and computes the number of cells that should be added, removed, or reallocated to a particular neighbour by means of a bandwidth estimation algorithm. These techniques experience information collection and negotiations overhead to maintain the schedule. The autonomous scheduling techniques are developed that do not involve the PCE element and there is no information negotiation overhead to build the schedule. It maintains consistent cell allocation with all neighbour nodes by using a hash of nodes MAC addresses.

Orchestra, an autonomous scheduling technique [28] is introduced where nodes autonomously determine communication schedules based on their MAC addresses. It defines multiple schedules as per traffic plane, viz. the application plane, routing plane, and MAC plane. The schedule is updated automatically as per changes in traffic and topology in the network. There are four types of slots in Orchestra: Common Shared Orchestra (CSO) slots, Receiver Based Shared Orchestra (RBSO) slots, Sender Based Shared Orchestra (SBSO) slots, and Sender Based Dedicated Orchestra (SBDO) slots. In RBSO, every node consists of one receive (Rx) cell whose location is determined by using the MAC address of the node, and one transmits (Tx) cell per neighbour whose location is determined by the MAC address of the neighbour node. In this method, several nodes allocate cells towards the same receiver and hence contention problems may arise in such slots. In the case of SBSO, the slot location is determined from the MAC address of the sender node. Every node allocates one Tx cell whose location is determined by the MAC address of the node and one Rx cell per neighbour whose location is determined by the MAC address of the neighbour node. The energy consumption in SBSO is higher than in RBSO as it has a greater number of Rx cells that always require a wakeup compared to the Tx cells. But it experiences fewer contention problems as it has one Rx cell per neighbour node.

e-TSCH-Orch [29] an enhancement over Orchestra scheduling has been proposed for the network with heterogeneous traffic. This technique keeps track of the traffic at every node by monitoring the pending packets in its queue. It negotiates with subtree to gather traffic information. This technique deals with the convergecast network and is not applicable for downward traffic. Escalator [30] is proposed where a schedule is created based on the RPL topology of the network. This schedule is built for convergecast network where every node except the root acts as the source and sends data to the root. The scheduling cell is determined by using certain rules based on the unique node ID and topology of the network. TESLA: Traffic-Aware Elastic Slotframe Adjustment [31] technique has been developed that analyses the effect of RPL control slotframe length. A smaller RPL slotframe size enables a more stable network but results in higher energy consumption. However, a larger slotframe size results in congestion collisions if the network involves a frequent change in a preferred parent. ALICE, a link-based autonomous cell scheduling technique [32], [33] is proposed in which a unique cell is scheduled for every link in the network. This technique does not involve any negotiation overhead to build a schedule instead it uses a hash of the MAC addresses of the sender and receiver. Each node uses a unique cell for every transmit and receive link with its neighbour. Also, this technique uses different schedules in different slotframe to reduce the collision of a particular cell. It uses slotframe number to change the schedule in every slotframe. ALICE uses multiple channel offsets whereas Escalator uses a single channel offset for every hop.

In the Orchestra scheme, a contention problem occurs if there is a high density of nodes and heavy traffic load in the network. In the scenario where all nodes except the root generate data packets and forward them to the root node, there is heavy traffic at the nodes near the root node and that results in congestion in the network. Also, latency increases with the traffic as every node needs additional timeslots to forward the packets. In ALICE, cell allocation is done randomly in a slotframe by using the nodes MAC addresses and hence, every source node needs multiple slotframes to deliver a packet to the root node. The performance of the network is affected with increase in slotframe length and traffic in the network. RBSO scheme suffers from high congestion which results in packet drops under heavy traffic conditions. In the SBSO scheme, only one cell is allocated to transmit to its parent, and hence there is a significant increase in the delay. Hence, there is further scope for the development of a new scheduling scheme for the industrial deterministic network to improve the overall communication between devices that handle critical data.

IV. PROPOSED SCHEDULING TECHNIQUE

This paper proposes a new Low Latency Autonomous (LLA) scheduling technique for the 6TiSCH convergecast network.

TABLE 1. Parameters and its values.

Parameters Definition	Value
Number of nodes N	50
Enhanced beacon slotframe size (SF_{eb})	397 slots
RPL control message slotframe size	31 slots
Unicast data slotframe size (SF_{uc})	29, 43, 61, 73, and 101 slots
Maximum hop count (H)	6 hops
Packet transmission interval	15 sec
Channel offset for enhanced beacon frame (Co_{eb})	0
Channel offset for RPL control message (Co_{rc})	1
Channel offsets for unicast data transmission (Co_{uc})	1, 2, and 3

In a convergecast network, all outlying nodes can send data to a root node using wireless links over multiple hops. The proposed technique schedules communication cells in consecutive segments of slotframe autonomously for each directional link from the source of data packets to the root sink node in the network. It creates a schedule where all the nodes in the network can deliver the data packet to the root node in a single slotframe. Therefore, it minimizes the latency by allocating a timeslot in consecutive segments along the packet forwarding route from a source to the sink node. It does not require any centralized mechanism, instead, it relies on the existing network topology and node MAC address for maintaining the schedules. Hence, it is a purely autonomous scheduling mechanism.

This technique consists of three slotframes, viz. slotframe for TSCH beacon, slotframe for RPL control message, and slotframe for unicast data. It also uses channel hopping for unicast data transmission to mitigate the multipath fading effect. Channel offset 0 is used for the TSCH beacon frame labeled as co_{eb}, channel offset 1 is used for the RPL control message labeled as co_{rc} and channel offsets $1, 2,$ and 3 are used for unicast data transmission from node w to s and labeled as $co_{uc}(w)$. Table. 1 show the parameters and their values used in the simulation.

TSCH Beacon Slotframe: Every node allocates two cells in an enhanced beacon slotframe, one to transmit a beacon frame and one to receive a frame. Time offset for TSCH beacon frame is calculated as a hash of sender MAC address,

$$to_{eb}(w) = hash\left[mac\_addr(w)\right] \% SF_{eb} \tag{1}$$

where $mac\_addr(w)$ is the MAC address of node w and SF_{eb} is enhanced beacon frame size. With $co_{eb} = 0$, one fixed cell is allocated by every node to transmit enhanced beacon frame, [slotOffset, channelOffset] $= [to_{eb}(w), 0]$.

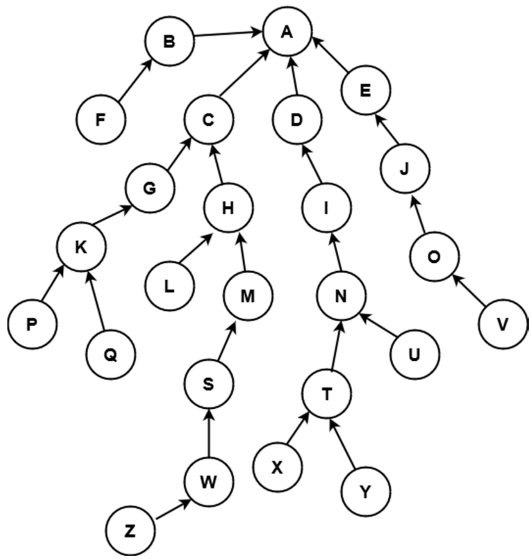

FIGURE 3. DODAG network.

RPL Control Message Slotframe: This slotframe is used for broadcasting RPL control messages such as DAO, DIO to synchronize nodes within the network. Time offset to_{rc} is selected as 0 and channel offset co_{rc} is selected as 1. Hence, one fixed cell is used for a control message at coordinate $[0, 1]$. This cell is shared by all nodes in the network for broadcasting control messages. The slotframe size should be sufficiently small to repeat the cell cycle more frequently. The control link is always-on for seamless communication with all neighbours.

Unicast Data Slotframe: In the directed acyclic graph shown in Fig. 3, each node except the root is a source of data and sends unicast data to the root node. In the proposed autonomous scheduling technique for such a scenario, nodes that are far away from the root schedules cells at the beginning of the slotframe, and nodes that are close to the root schedules cells at the end of the slotframe.

The slotframe is divided into a number of segments, and one unique cell is allocated to every link in the assigned segment. We divide the slotframe into H segments, which is equal to the number of maximum hops in the network. A slotframe with 73 timeslots is shown in Fig. 4 and it is divided into 6 segments. The segment length is written as

$$L_{seg} = \frac{SF_{uc}}{H} \tag{2}$$

where H is the number of maximum hops in the network and SF_{uc} is a unicast slotframe length. Time offset and channel offset for unicast data transmission are computed using MAC addresses of sender and receiver node as a hash of directional link.

This scheme computes one unique cell in a specific segment for every link in the network as per the hop distance of the node. If a sender node w is at k hop distance from the root node and a receiver node s is a parent node of w, the time

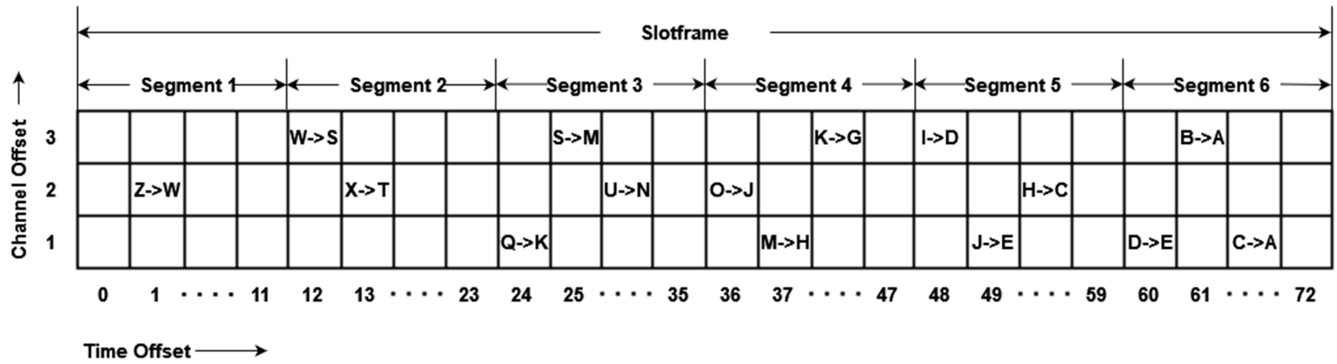

FIGURE 4. Slotframe segmentation.

Maximum Number of hops in network = H
Define Unicast slotframe length → SF_{uc}
Divide unicast slotframe into H segments
Length of each segment → $L_{seg} = SF_{uc} / H$
Derive Unicast timeOffset →
  $to_{ut}(ws)$ for link between node w of hop distance k and its parent node →
    for (hop distance k from H to 1)
      $to_{ut}(ws)$ → in segment $(H - k + 1)$
  $to_{ur}(zw)$ for link between node w of hop distance k and its child node →
    for (hop distance k from $(H - 1)$ to 0)
      $to_{ur}(zw)$ → in segment $(H - k)$
Derive channelOffset for sender w →
    $co_{uc}(w) \rightarrow hash[mac\_addr(w)]\ mod\ c_{nu} + 1$
Allocate cell for link between node w and its parent node s →
    $[to_{ut}(ws), co_{uc}(w)]$
Allocate cell for link between node w and its child node z →
    $[to_{ur}(zw), co_{uc}(z)]$

FIGURE 5. Pseudocode of the proposed scheduling algorithm.

offset for the link between node w and s can be written as

$$to_{ut}\,(ws) = hash\,[d \cdot mac\_addr(w) + mac\_addr\,(s)]\,\%L_{seg} + (H-k)\cdot L_{seg} \quad (3)$$

where d is the direction of data transmission and it distinguishes between link $(w \rightarrow s)$ and link $(s \rightarrow w)$. The cell location for a link $(w \rightarrow s)$ is determined from their MAC addresses and using the hash function. Hence, the Tx time offset for sender node w to its parent node s is computed in slotframe segment number $(H - k + 1)$.

Similarly, if the receiver node w is at k hop distance from the root node and the sender node z is a child node of w, the Rx time offset for the link between node z and w is written as

$$to_{ur}\,(zw) = hash\,[d \cdot mac\_addr(z) + mac\_addr\,(w)]\,\%L_{seg} + (H-k-1)\cdot L_{seg} \quad (4)$$

Time offset for a link $(z \rightarrow w)$ is computed in a slotframe segment number $(H - k)$. With this allocation, node w

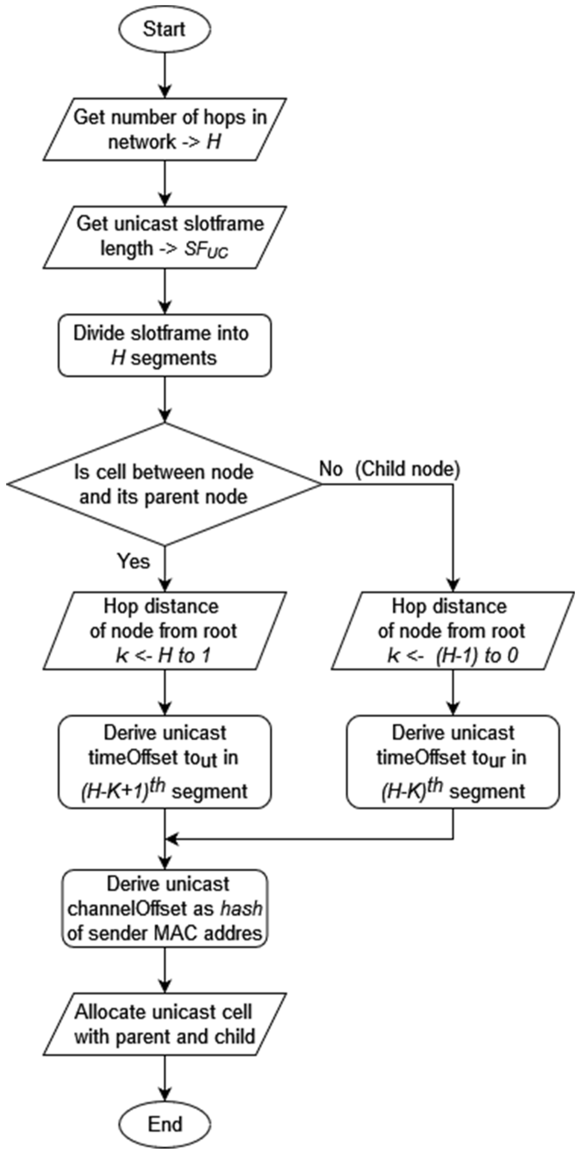

FIGURE 6. Flowchart of the proposed scheduling algorithm.

receives a data packet from node z in a segment $(H - k)$ and can forward it to node s immediately in the next segment.

A leaf node that is H hop away, viz. at maximum hop distance, from the root node, allocates the Tx cell in the first segment of slotframe for communication with its parent node. A node that is one hop away from the root node allocates a Tx cell in H^{th} segment and Rx cell in $(H-1)^{th}$ segment of the slotframe.

For the network in Fig. 3, a unique Tx cell in segment 1 is allocated to every link from nodes that are H hop away to $(H-1)$ hop away nodes. One Tx cell in segment 2 is allocated to every link between nodes that are $(H-1)$ and $(H-2)$ hop away and so on. For a link between the root node and its neighbors, Rx cells for the root are allocated in the last segment. To allocate a unique cell for every directional link slotframe length should be higher than the $(2N-2)/3$ for N nodes. Small slotframe size enables more transmission opportunities as timeslots repeat more often. On the other hand, it results in more energy consumption as the node wakes up frequently. The large slotframe size is resulting in higher latency and lower PDR.

Channel offset for unicast data transmission is derived as a hash of the sender's MAC address. If node w is the sender and node s is the receiver, then the channel offset is written as

$$co_{uc}(w) = hash\,[mac\_addr(w)]\,\%C_{nu} + 1 \quad (5)$$

where C_{nu} is the number of channel offsets used for unicast traffic and it is equal to 3. The sender node uses its MAC address to determine the channel offset and it is the same all the time. However, communication channel frequency is determined as

$$Channel\_freq = F\,\{(ASN + channelOffset)\,\%C_{nu}\} \quad (6)$$

where F is the look-up table of channel frequencies and ASN is Absolute Slot Number. ASN represents the number of time slots elapsed since the network is started. It changes in every time slot resulting in different channel frequencies in a new slotframe. Channel offset 0 is used for the TSCH beacon frame which is the highest priority frame. Hence, to avoid the overlap on a particular cell in a slotframe with unicast data and dropping of a packet on that cell channel offset 0 is not used for data frames.

Fig. 5 and Fig. 6 show the pseudocode and flowchart of the proposed scheduling algorithm, respectively.

V. PERFORMANCE EVALUATION

The proposed scheduling scheme is implemented on the open-source Contiki-NG IoT platform [34] and for the performance evaluation, we have used the Contiki Cooja simulator that supports hardware-level emulation. The performance of a proposed scheme is compared with the existing autonomous scheduling techniques such as SBSO and ALICE. The different number of slotframes and segments were considered for the performance analysis and comparison. As Orchestra, the proposed scheme uses three slotframes as per the type of traffic as; the enhanced beacon slotframe with 397 timeslots for transmission of the beacon data, the broadcast slotframe

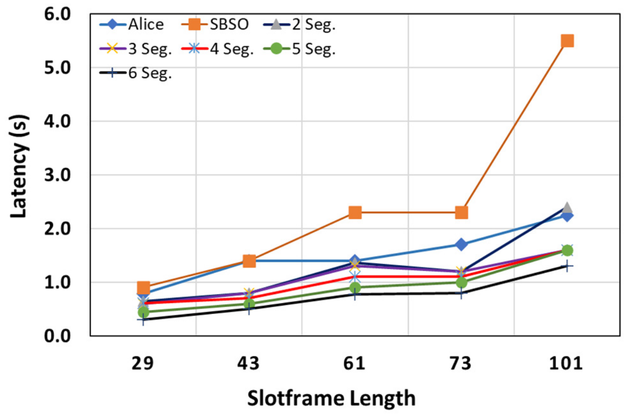

FIGURE 7. Latency versus slotframe length.

with a length of 31 timeslots for the RPL control messages exchanges, and the unicast slotframe of variable length for the transmission of application data. The size of a beacon slotframe should be larger than the number of nodes in the network for contention free transmission. Every node has one Tx cell and one Rx cell to exchange beacon data from its selected time source. The probability of contention is a function of traffic rate and the slotframe length. To avoid the collision of the RPL control message with the highest priority enhanced beacon cell, slotframe length of 31 is selected for RPL control messages.

The simulation setup consists of 50 nodes that are placed in the 7×4 grid (700 m $\times$ 400 m) as a usual scenario in the IIoT network. Data from different machines in the industry are collected at the root node at 6 hops away and sent over the internet for processing and analysis purposes. The network topology consists of the root as a sink node and 49 source nodes. The transmission and interference range of nodes is set to 100 m and the packet transmission interval is set to 15 sec for each scheduling scheme. The parameters such as average end-to-end latency, average duty cycle, and Packet Delivery Ratio (PDR) of a node for different slotframe sizes are considered for performance analysis. Slotframe length is varied from 29 slots to 101 slots. Size is selected based on the number of nodes in the network, traffic conditions, and performance parameters like energy consumption and radio duty cycle. Smaller slotframe results in more energy consumption, whereas, a higher slotframe size leads to more latency. Also, PDR is affected with increasing slotframe size as a lesser number of cells are available for transmission.

As demonstrated in Fig. 7, the results show that the proposed LLA scheme for a slotframe with 29 slots and 6 segments has a latency of only 0.3 sec, which is 66.6% and 62.5% lesser than that of SBSO and ALICE, respectively. For a slotframe with 101 slots, our scheme has maximum latency of 1.3 sec, which is 76.4% and 41% lesser than that of SBSO and ALICE, respectively. In the SBSO scheme, latency increases with an increase in slotframe length as for every node, multiple timeslots are required to receive packets from all neighbours while only one timeslot to send them to the parent. Hence in multihop networks, the latency increases proportionally to the slotframe length. In the proposed scheme, cells are allocated in consecutive segments for the transmission path from the source of the packet to the root node and can transmit a packet in one slotframe. This is because of dividing a slotframe into segments equal to the maximum number of hops in the network. We have also analysed the performance by dividing the slotframes into 2, 3, 4, and 5 segments. In all cases, the latency performance of the proposed scheme is better than the scheduling schemes reported in the literature. However, if we continue segmenting the slotframe, it may lower the performance due to the non-availability of sufficient cells for communication.

End-to-end latency in the proposed scheme is lesser as cells are allocated in consecutive segments of the slotframe. In the best-case scenario, a packet is transmitted from source to destination in less than 1 sec if slotframe size is 101 slots. Whereas, if link layer packet is need to be retransmitted then the end-to-end transmission may take more than one slotframe which increases the end-to-end latency. Latency is minimum if the number of segments is equal to the maximum number of hops in the network. SBSO is a node-based scheduling technique in which one transmit cell is scheduled and one receive cell is scheduled per neighbour. If there are more packets in the queue, this scheme suffers from a latency problem as the node can send only one packet in every slotframe. The latency in ALICE is higher if all nodes except the root are a source of data and also forwarding the packets from other sources to the root. Cell allocation is done randomly within the slotframe using the MAC address of the sender and receiver node. Also, the heterogeneous distribution of traffic within the industrial environment results in congestion at the nodes close to the sink. The latency increases with the increase in slotframe length, a number of nodes, and traffic in the network. In a lossy network, if any intermediate node fails to forward the packet in a specified segment, then it has to wait for the next slotframe. In this case, the latency of the proposed algorithm will be the same as existing techniques.

The radio duty cycle is the percentage of radio ON time to the total timeslot period of the node. The higher duty cycle of the node consumes more energy and reduces its lifetime. With the increase in slotframe size, the duty cycle of a node is reduced as a greater number of cells are available for data communication. The results of a node duty cycle for different sizes of unicast slotframe are shown in Fig. 8. At receive cell for receiving any packet from neighbour nodes, each node wakes up and listens to the medium. As illustrated in Fig. 8, the duty cycle of the proposed scheme is 4% and 2.7% for the 6 segment slotframe with 29 slots and 101 slots, respectively, which is the lowest amongst the other scheduling schemes. For SBSO, it is 5% (for 29 slots) and 4.5% (for 101 slots) as each node wakes up a number of times in every slotframe. The reduction in duty cycle in SBSO is at the cost of lower PDR. In the case of ALICE, the duty cycle is 4.2% for slotframe of 29 slots, and it is reduced to 2.8% if slotframe size is increased to 101 slots but for this large slotframe size, the average end-to-end latency is higher.

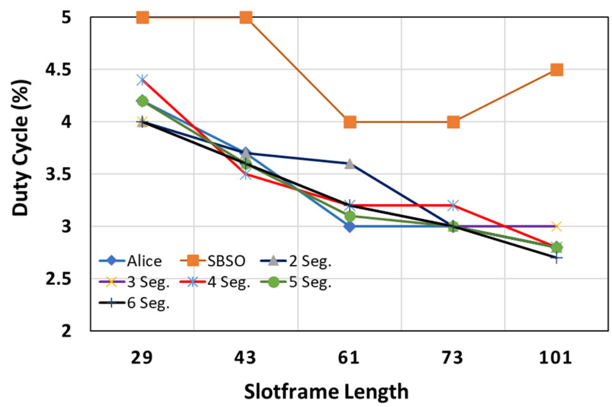

FIGURE 8. Duty cycle versus slotframe length.

TABLE 2. PDR for different slotframe length.

Slotframe length	Alice	SBSO	Proposed technique with 6 Segments
29	99.98	99.96	100
43	99.79	99.96	100
61	99.92	99	100
73	99.96	99.5	100
101	99.9	69.64	99.4

The PDR in a network is the percentage of the total number of packets delivered to the destination to the total number of packets sent by the source node. Table 2 shows the PDR performance for different slotframe sizes. As in ALICE, the PDR of the proposed scheme is higher than 99%. In the case of SBSO, PDR is restricted to 69.5% if slotframe size is increased to 101 slots due to lesser transmission opportunities. PDR is observed to validate the performance of the proposed schedule by increasing the slotframe size. Slotframe is divided into a number of segments that may decrease the number of cells available for transmission for the larger slotframe size. This method selects a specific segment for the transmission based on the hop distance of the node. It uses MAC addresses of sender and receiver along with the direction of transmission to find the cell location. Although the proposed technique is based on slotframe segmentation computation, it shows a significant reduction in end-to-end latency in the transmission.

VI. CONCLUSION

A new Low Latency Autonomous (LLA) scheduling technique for the 6TiSCH deterministic industrial applications is proposed. We introduce the concept of segmentation in the slotframe for cell scheduling. The schedule uses the MAC addresses of the sender and receiver node and does not create any traffic overhead for cell negotiation. The performance of the proposed scheme is observed by dividing slotframe into a different number of segments. It is observed that performance parameter latency is greatly improved if the number of segments is equal to the maximum number of hops. This approach is more efficient as all nodes at any specific level in the network will acquire unique cells in the same segment for communication. The results prove that the proposed scheme outperforms the existing techniques by reducing the latency and radio duty cycle. The proposed scheme reduces the end-to-end latency up to 41% and the duty cycle up to 3.5% as compared to existing schemes. The proposed scheduling scheme can be improved by considering the congestion near the root node.

REFERENCES

[1] D. Chen, M. Nixon, S. Han, A. K. Mok, and X. Zhu, "WirelessHART and IEEE 802.15.4e," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Busan, South Korea, Feb. 2014, pp. 760–765, doi: 10.1109/ICIT.2014.6895027.

[2] R. Severino, S. Ullah, and E. Tovar, "A cross-layer QoS management framework for ZigBee cluster-tree networks," *Telecommun. Syst.*, vol. 63, no. 3, pp. 381–397, Nov. 2016, doi: 10.1007/s11235-015-0128-0.

[3] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert, "IETF 6TiSCH: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 595–615, 1st Quart., 2020, doi: 10.1109/COMST.2019.2939407

[4] B. S. Chaudhari, M. Zennaro, and S. Borkar, "LPWAN technologies: Emerging application characteristics, requirements, and design considerations," *Future Internet*, vol. 12, no. 3, p. 46, Mar. 2020, doi: 10.3390/fi12030046.

[5] P. Thubert, *An Architecture for IPv6 Over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)*, document RFC 9030, May 2021. [Online]. Available: https://datatracker.ietf.org/doc/rfc9030/

[6] M. R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, and T. Engel, "6TiSCH wireless industrial networks: Determinism meets IPv6," in *Internet of Things* (Smart Sensors, Measurement and Instrumentation), vol. 9, S. Mukhopadhyay, Ed. Cham, Switzerland: Springer, Jan. 2014, pp. 111–141, doi: 10.1007/978-3-319-04223-7_5.

[7] B. S. Chaudhari, S. Ghorpade, and M. Zennaro, "Towards green computing: Intelligent bio-inspired agent for IoT-enabled wireless sensor networks," *Int. J. Sensor Netw.*, vol. 35, no. 2, p. 121, 2021, doi: 10.1504/IJSNET.2021.10036232.

[8] X. Vilajosana, T. Watteyne, M. Vucinic, T. Chang, and K. Pister, "6TiSCH: Industrial performance for IPv6 Internet-of-Things networks," *Proc. IEEE*, vol. 107, no. 6, pp. 1153–1165, Jun. 2019, doi: 10.1109/JPROC.2019.2906404.

[9] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "IEEE 802.15.4e in a nutshell: Survey and performance evaluation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1989–2010, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2800898.

[10] M. B. Hassan, E. Sayed Ali, R. A. Mokhtar, R. A. Saeed, and B. S. Chaudhari, "NB-IoT: Concepts, applications, and deployment challenges," in *LPWAN Technologies for IoT and M2M Applications*, B. Chaudhari and M. Zennaro, Eds. New York, NY, USA: Academic, 2020, ch. 6, pp. 119–144, doi: 10.1016/B978-0-12-818880-4.00006-5.

[11] M. Vucinic, T. Chang, B. Skrbic, E. Kocan, M. Pejanovic-Djurisic, and T. Watteyne, "Key performance indicators of the reference 6TiSCH implementation in Internet-of-Things scenarios," *IEEE Access*, vol. 8, pp. 79147–79157, 2020, doi: 10.1109/ACCESS.2020.2990278.

[12] S. N. Ghorpade, M. Zennaro, and B. Chaudhari, "Binary grey wolf optimisation-based topology control for WSNs," *IET Wireless Sensor Syst.*, vol. 9, no. 6, pp. 333–339, 2019.

[13] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, "Decentralized traffic aware scheduling in 6TiSCH networks: Design and experimental evaluation," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 455–470, Dec. 2015, doi: 10.1109/JIOT.2015.2476915.

[14] M. Domingo-Prieto, T. Chang, X. Vilajosana, and T. Watteyne, "Distributed PID-based scheduling for 6TiSCH networks," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1006–1009, May 2016, doi: 10.1109/LCOMM.2016.2546880.

[15] M. R. Palattella, T. Watteyne, Q. Wang, and K. Muraoka, "On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks," *IEEE Sensors J.*, vol. 16, no. 2, pp. 550–560, Jan. 2016, doi: 10.1109/JSEN.2015.2480886.

[16] T. Chang, T. Watteyne, Q. Wang, and X. Vilajosana, "LLSF: Low latency scheduling function for 6TiSCH networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Washington, DC, USA, May 2016, pp. 93–95, doi: 10.1109/DCOSS.2016.10.

[17] K. Muraoka, T. Watteyne, N. Accettura, X. Vilajosana, and K. S. J. Pister, "Simple distributed scheduling with collision detection in TSCH networks," *IEEE Sensors J.*, vol. 16, no. 15, pp. 5848–5849, Aug. 2016, doi: 10.1109/JSEN.2016.2572961.

[18] T. P. Duy, T. Dinh, and Y. Kim, "Distributed cell selection for scheduling function in 6TiSCH networks," *Comput. Standards Interfaces*, vol. 53, pp. 80–88, Aug. 2017, doi: 10.1016/j.csi.2017.03.008.

[19] G. Daneels, B. Spinnewyn, S. Latré, and J. Famaey, "ReSF: Recurrent low-latency scheduling in IEEE 802.15.4e TSCH networks," *Ad Hoc Netw.*, vol. 69, no. 1, pp. 100–114, Feb. 2018, doi: 10.1016/j.adhoc.2017.11.002.

[20] I. Amezcua Valdovinos, P. E. Figueroa Millán, J. A. Pérez-Díaz, and C. Vargas-Rosales, "Distributed channel ranking scheduling function for dense industrial 6TiSCH networks," *Sensors*, vol. 21, no. 5, p. 1593, Feb. 2021, doi: 10.3390/s21051593.

[21] A. Karaagac, I. Moerman, and J. Hoebeke, "Hybrid schedule management in 6TiSCH networks: The coexistence of determinism and flexibility," *IEEE Access*, vol. 6, pp. 33941–33952, 2018, doi: 10.1109/ACCESS.2018.2849090.

[22] O. Tavallaie, J. Taheri, and A. Y. Zomaya, "Towards optimizing time-slotted channel hopping scheduling on 6TiSCH networks: Poster abstract," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, Nov. 2020, pp. 737–738, doi: 10.1145/3384419.3430454.

[23] F. Righetti, C. Vallati, S. K. Das, and G. Anastasi, "Analysis of distributed and autonomous scheduling functions for 6TiSCH networks," *IEEE Access*, vol. 8, pp. 158243–158262, 2020, doi: 10.1109/ACCESS.2020.3020153.

[24] D. Vasiljevic and G. Gardasevic, "Packet aggregation-based scheduling in 6TiSCH networks," in *Proc. IEEE 18th Int. Conf. Smart Technol. (EUROCON)*, Novi Sad, Serbia, Jul. 2019, pp. 1–5, doi: 10.1109/EUROCON.2019.8861910.

[25] S. Kim, H.-S. Kim, and C.-K. Kim, "A3: Adaptive autonomous allocation of TSCH slots," in *Proc. 20th Int. Conf. Inf. Process. Sensor Netw. (Co-Located CPS-IoT Week)*, Nashville, TN, USA, May 2021, pp. 299–314, doi: 10.1145/3412382.3458273.

[26] M. Osman and F. Nabki, "OSCAR: An optimized scheduling cell allocation algorithm for convergecast in IEEE 802.15.4e TSCH networks," *Sensors*, vol. 21, no. 7, p. 2493, Apr. 2021, doi: 10.3390/s21072493.

[27] T. Chang, M. Vučinić, X. V. Guillén, D. Dujovne, and T. Watteyne, "6TiSCH minimal scheduling function: Performance evaluation," *Internet Technol. Lett.*, vol. 3, no. 4, Jul. 2020, doi: 10.1002/itl2.170.

[28] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, Seoul, South Korea, Nov. 2015, pp. 337–350, doi: 10.1145/2809695.2809714.

[29] S. Rekik, N. Baccour, M. Jmaiel, K. Drira, and L. A. Grieco, "Autonomous and traffic-aware scheduling for TSCH networks," *Comput. Netw.*, vol. 135, pp. 201–212, Apr. 2018, doi: 10.1016/j.comnet.2018.02.023.

[30] S. Oh, D. Hwang, K.-H. Kim, and K. Kim, "Escalator: An autonomous scheduling scheme for convergecast in TSCH," *Sensors*, vol. 18, no. 4, p. 1209, 2018, doi: 10.3390/s18041209.

[31] S. Jeong, J. Paek, H.-S. Kim, and S. Bahk, "TESLA: Traffic-aware elastic slotframe adjustment in TSCH networks," *IEEE Access*, vol. 7, pp. 130468–130483, 2019, doi: 10.1109/ACCESS.2019.2940457.

[32] S. Kim, H.-S. Kim, and C. Kim, "ALICE: Autonomous link-based cell scheduling for TSCH," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Montreal, QC, Canada, Apr. 2019, pp. 121–132, doi: 10.1145/3302506.3310394.

[33] A. Elsts, S. Kim, H.-S. Kim, and C. Kim, "An empirical survey of autonomous scheduling methods for TSCH," *IEEE Access*, vol. 8, pp. 67147–67165, 2020, doi: 10.1109/ACCESS.2020.2980119.

[34] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki–A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Tampa, FL, USA, Nov. 2004, pp. 455–462, doi: 10.1109/LCN.2004.38.

NILAM M. PRADHAN (Member, IEEE) received the M.E. degree in electronics and telecommunication engineering, in 2012. She is currently pursuing the Ph.D. degree with the Maharashtra Institute of Technology, Pune, India, under Savitribai Phule Pune University. She is also working as an Assistant Professor with the School of Electronics and Communication Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune. Her research interests include the Internet of Things and wireless sensor networks. She is a member of SAE and a Life Time Member of ISTE.

BHARAT S. CHAUDHARI (Senior Member, IEEE) received the M.E. degree in telecom engineering and the Ph.D. degree from Jadavpur University, Kolkata, India, in 1993 and 2000, respectively. He was a Full Professor of electronics and telecommunication engineering at the Pune Institute of Computer Technology and the Dean with the International Institute of Information Technology, Pune, India. He joined as a Professor with MIT World Peace University, (then MIT Pune) Pune, in 2014. He has been a Simon's Associate with the International Centre for Theoretical Physics (ICTP), Italy, since 2015. He has authored more than 82 research papers in the field of wireless, telecom, and optical networks. His research interests include low power wide area networks, the Internet of Things, wireless sensors networks, and optical networks. He received the Young Scientist Research Grant from the Department of Science and Technology, Government of India. He is the Founder Chair of the IEEE Pune Section and a fellow of IETE and IE(I).

MARCO ZENNARO (Senior Member, IEEE) received the M.Sc. degree in electronic engineering from the University of Trieste, Italy, and the Ph.D. degree from the KTH-Royal Institute of Technology, Stockholm, Sweden. He is currently a Research Scientist with the Abdus Salam International Centre for Theoretical Physics, Trieste, Italy, where he is an Unit Co-Ordinator with the Science, Technology and Innovation Department. He is a Visiting Professor with the KIC-Kobe Institute of Computing, Japan. His research interests include ICT4D, the use of ICT for development, and in particular he investigates the use of the IoT in developing countries. He has given lectures on wireless technologies in more than 30 countries.