

# Traffic-Aware 6TiSCH Routing Method for IIoT Wireless Networks

Yuvin Ha<sup>1</sup>, *Member, IEEE*, and Sang-Hwa Chung<sup>1</sup>, *Member, IEEE*

**Abstract**—The 6TiSCH network is a key technology of the Industrial Internet of Things (IIoT), providing high reliability and deterministic features. However, IIoT sensors and facility applications generate heavy traffic loads. Various time-slotted channel hopping (TSCH) medium access layer (MAC) and routing protocols for low-power and lossy network (RPL) methods have been studied at each network layer to address this issue. Although TSCH MAC is standard for IIoT low-rate wireless networks, it has not been adequately considered in conventional RPL studies. Conventional RPL methods do not effectively utilize the characteristics of the TSCH MAC in heavy-traffic networks and may hinder performance. In this article, we introduce a traffic-aware (TA) RPL method that utilizes the cell allocation information of TSCH MAC. The number of allocated TSCH cells determines the bandwidth reserved by each node and includes the link quality and traffic information measured by the MAC layer. TA RPL facilitates efficient load balancing using cell allocation information and improves the bandwidth utilization of the network. Additionally, the proposed method can improve the deterministic features of the IIoT wireless network because the effect of the routing decision of each device can be predicted. Moreover, the packet delivery ratio and bandwidth utilization are improved by up to 31% and 20%, respectively, compared with the conventional methods based on simulation and testbed evaluations.

**Index Terms**—6TiSCH, Industrial Internet of Things (IIoT), load balancing, routing protocol for low-power and lossy network, time-slotted channel hopping (TSCH), wireless network, IEEE802.15.4.

## I. INTRODUCTION

INTERNET of Things (IoT) networks connect numerous smart devices over wireless channels [1] and are being rapidly adopted by global industries, such as healthcare [2], smart homes [3], and smart cities [4]. Industrial IoT (IIoT) technologies connect numerous facilities and workers. Consequently, much effort is being directed toward realizing intelligent and efficient industrial

Manuscript received 9 March 2022; revised 13 May 2022; accepted 6 June 2022. Date of publication 4 July 2022; date of current version 7 November 2022. This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, through the Grand Information Technology Research Center Support Program under Grant IITP-2022-2016-0-00318 supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) and in part by IITP Grant funded by the Korea Government (MSIT, Artificial Intelligence Convergence Research Center [Pusan National University]) under Grant 2020-0-01450. (Corresponding author: Sang-Hwa Chung.)

The authors are with the Department of Information Convergence Engineering, Pusan National University, Busan 46241, South Korea (e-mail: yuobin.ha.1013@gmail.com; shchung@pusan.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3184310

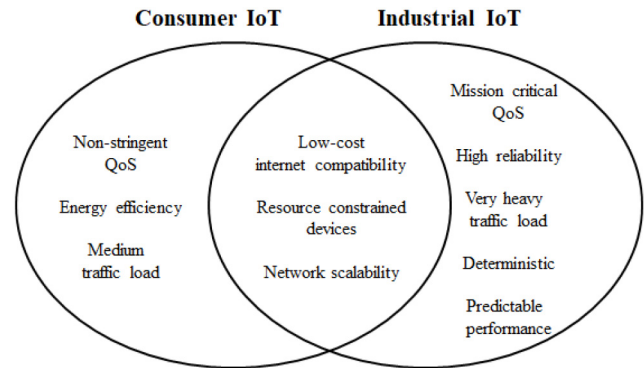


Fig. 1. Requirements and characteristics of consumer IoT and IIoT networks.

performance improvements [5]. Low-rate wireless network technologies are required for IIoT devices.

The requirements and characteristics of consumer IoT and IIoT networks [6], [7] are illustrated in Fig. 1. The latter requires a higher level of performance, and specifications are stricter because they are directly linked to productivity and safety. An IIoT network must be reliable, even in the presence of physical obstacles, external noise, and heavy traffic loads. Moreover, it must allow for predictable performance based on the addition/removal of devices and traffic-volume changes.

IEEE 802.15.4 provides a description of the time-slotted channel hopping (TSCH) medium access layer (MAC) protocol—a standard technology for low-rate wireless networks that provides reliability and deterministic characteristics via coordinated communications based on link scheduling [8]. Furthermore, it is robust against external interference owing to its channel-hopping implementation [9]. The IPv6 routing protocol for low-power and lossy networks (RPL) [10] builds network paths for IoT applications atop the TSCH MAC. RPL creates distributed routing paths with low overhead and is designed to facilitate performance suitable for specific purposes. RFC5673 [11] analyzes and describes the industrial requirements for RPL.

CISCO's CG-Mesh [12] is a commercial RPL and IEEE 802.15.4 network solution for industrial applications. CG-Mesh's single RPL network connects up to 5000 nodes, and it feasibly meets the market demand for large-scale IIoT networks. Although each IoT device transmits only a few bytes of data in a fixed period, a sink (or gateway) of a large-scale

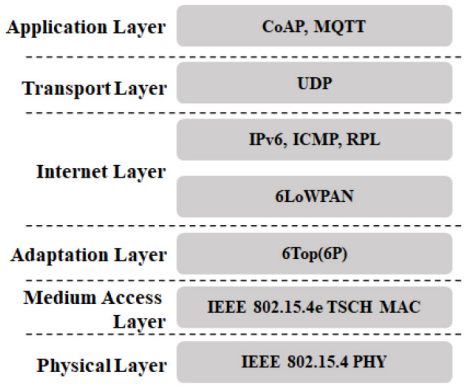


Fig. 2. 6TiSCH IoT network stack.

network must handle huge traffic loads, which presents a very important issue for IIoT networks.

The 6TiSCH working group (WG) defined the 6TiSCH operational (6top) [13] sublayer as the one to adopt the TSCH design for industrial environments within the RPL context [14]. Fig. 2 displays the 6TiSCH network stack, which provides a key IIoT technology that assures high reliability and deterministic features [8].

TSCH is designed for high reliability and deterministic networks based on the assumption of an industrial environment, and various link-scheduling methods for mission-critical Quality of Service (QoS) [15]–[17] have been studied. Recent RPL studies [18]–[20] have introduced various techniques to facilitate the provision of QoS, which is required in industrial settings. However, they have not sufficiently considered the 6TiSCH network.

In most wireless multihop networks, devices are bottlenecked based on their proximity to the root node. Conventional reactive load-balancing methods [21] generally change routing paths after detecting contention or congestion. However, when a routing path is changed in the 6TiSCH network, all previously allocated link schedules are discarded, and a new link schedule is established. Hence, network performance degradation is unavoidable until the link schedule is sufficiently saturated. This is not a small overhead, and a proactive path selection policy in the 6TiSCH network for IIoT can provide a better solution.

It is difficult to predict the bandwidth required to handle a given amount of traffic in the contention-based MAC (e.g., ContikiMAC [22]). By contrast, TSCH MAC calculates the appropriate amount of bandwidth required for traffic through a link-scheduling method. The link schedule includes link quality and traffic information, and each device can accurately estimate the available bandwidth of each routing path using this information.

In this article, we propose a traffic-aware (TA)-RPL method that efficiently handles heavy traffic using a cross-layer approach for TSCH MAC. The proposed method enables accurate and intuitive load balancing by using the link schedule information of the TSCH MAC. Thus, the approach can improve the deterministic feature of the IIoT wireless network. The main contributions of this study are as follows.

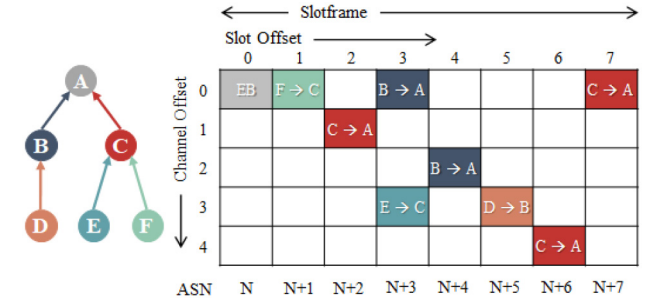


Fig. 3. Example of TSCH slotframe structure and link schedule.

- 1) We analyze the effect of heavy traffic on the 6TiSCH network and illustrate the bandwidth problems associated with the conventional RPL.
- 2) To address these problems, we introduce a TAR method based on the TSCH MAC protocol.
- 3) We evaluate the performance of the proposed method based on an extensive analysis of simulation and experimental results on a testbed.
- 4) We describe the load balancing characteristics of each metric in the 6TiSCH network through comparison with various methods.

The remainder of this article is structured as follows. Section II describes the background and related research to elucidate the proposed method. In Section III, we analyze the limitations of the conventional RPL and the problems caused by heavy traffic. In Section IV, mechanisms and methodologies that are used to solve these problems are introduced, and in Section V, the performance of the proposed method is evaluated through extensive experimental testing. Finally, the main conclusions are summarized in Section VI.

## II. BACKGROUND AND RELATED WORK

In this section, we briefly introduce the core technology of the 6TiSCH network stack, as shown in Fig. 2, and we describe the related literature on handling heavy traffic loads in RPL networks.

### A. IEEE 802.15.4 TSCH MAC

IEEE 802.15.4 TSCH [23] operates in units of slotframes of a matrix structure divided into time and channel. A slotframe consists of a fixed number of timeslots, and each device calculates the channel used for communication using (1) for each

$$CH = \text{macHopSeq}[(ASN + Ch_{\text{offset}}) \% \text{macHopSeqLen}] \quad (1)$$

where  $\text{macHopSeq}$  is an array that specifies the hopping order and the channel offset  $Ch_{\text{offset}}$  is a logical channel allocated by the link schedule. The absolute slot number (ASN) represents the sequence of slots counted from the beginning of the network. Therefore, its value continuously changes. Thus, the same channel-offset value is calculated differently in each timeslot. This feature provides robustness against external interference and the multipath fading problem [9].

An example of the TSCH slotframe structure and the link schedule is displayed in Fig. 3. A time-and-channel element

pair in the slot frame matrix is a *link* or *cell* in 6TiSCH. Link scheduling involves the allocation of a communication time and a frequency to be used for each device pair. It is calculated in units of slotframes and is continuously repeated. Because all devices communicate based on allocated schedules, interference by other devices in the network does not occur if ideal link scheduling is assumed.

Link scheduling must consider various requirements. First, because a resource-constrained device has a single transceiver, it can communicate with one device. Therefore, devices with an interference relationship must allocate links to different times or frequencies. This is the most basic purpose of link scheduling to provide reliability by preventing collisions inside the network.

In the example presented in Fig. 3, device D creates a data packet at slot offset 0. Device D transmits the packet to device B on the link of the assigned slot offset 5, and device B may transmit the packet to device A at slot offset 3 of the next slotframe. In this case, 11 timeslots are required to deliver the packet to device A, but the packet may be delivered in at least three timeslots according to the link-scheduling scheme. TSCH link scheduling is associated with various issues in terms of QoS, and a method for addressing these problems is introduced in [17].

Many TSCH link-scheduling studies presuppose that devices generate data traffic at a fixed period [15]–[17], and that the link-scheduling method must be able to allocate enough links to address the resultant traffic. Device C in Fig. 3 must deliver data packets of devices E and F as well as its packet to device A. Therefore, three CA schedules are allocated. Additionally, numerous TSCH link-scheduling studies have been conducted based on various methodologies and QoS [24].

### B. Internet Engineering Task Force RPL

IoT networks must be optimized to meet the requirements of a variety of services based on an objective function (OF). RPL devices periodically transmit a destination-oriented directed acyclic graph (DODAG) information object (DIO) to form a DODAG directed toward the root. DIO messages include network information and the metrics of each device. The device that receives the DIO calculates the *rank*, which is the logical distance to the root, by substituting various metric information [25] into the OF. The device with the lowest rank is selected as the preferred parent and is regarded as the major routing path

$$\text{rank}(N) = \text{rank}(P) + \text{rankIncrease} \quad (2)$$

$$\text{rankIncrease} = (\text{Rf} * \text{Sp} + \text{Sr}) * \text{MinHopRankIncrease}. \quad (3)$$

The Internet Engineering Task Force (IETF) ROLL WG defined a mechanism for calculating the rank of each device using (2) and (3), as defined in OF0 [26], but it does not offer metrics. Table I shows the OF0 parameter setting proposed by RFC8180 [27] to supplement OF0. RPL devices using this setting select the device with the lowest expected transmission count (ETX) as the preferred parent.

TABLE I  
OF0 PARAMETERS

OF0 Parameters	Value
Rf	1
Sp	(3*ETX)-2
Sr	0
MinHopRankIncrease	256
MINIMUM_STEP_OF_RANK	1
MAXIMUM_STEP_OF_RANK	9
ETX limit to select a parent	3

RPL can optimize the network in various ways according to the OF design and is associated with a variety of issues, such as uplink/downlink traffic, load balancing, multicast, multi-sync, mobility, and delay time [21]. Recent RPL studies have considered the industrial environment (this is discussed in Section II-D).

### C. 6TiSCH

IEEE 802.15.4 [23] only defines the TSCH operation mechanism and does not specify the link-scheduling creation, exchange, and operation policy. IETF 6TiSCH WG was established to address this technology gap and improve the usability of TSCH [13]. The 6TiSCH WG defines a 6top sublayer as an adaptation layer between TSCH and RPL. RFC8180 proposed a 6TiSCH minimal configuration that complements elements that are not defined in the TSCH and RPL standards for the basic compatibility of IoT devices [27]. Additionally, the 6TiSCH WG proposed the minimal scheduling function (MSF) [28] as a simple and lightweight link-scheduling method and defines 6top Protocol (6P) [29], including the protocols and methodologies, for schedule allocation.

### D. Related Work

Over the last decade, various methods have been proposed to improve the performance of RPL networks [21], [30]. ELITE [31] introduced a cross-layer method that improves the efficiency of RPL by utilizing MAC layer information. ELITE uses the number of *strokes* used in ContikiMAC [22] as a metric. Stroke is a type of preamble, and a device transmits a certain number of strokes to indicate the start of communication. Because ELITE uses the number of strokes as an OF metric, each device selects a routing path that uses the minimum number of strokes. It has been demonstrated that a routing path that utilizes cross-layer RPL can improve the efficiency of a specific MAC technique.

Load balancing is a major issue in routing protocols, and the most widely used metrics are ETX, energy, and hop count; techniques are classified according to how the metrics are used. Among recently proposed load-balancing methods, we introduce the ones that are suitable for 6TiSCH networks. With the performance evaluation described in Section V-A, we analyze the functionality of various load-balancing metrics in the 6TiSCH network.

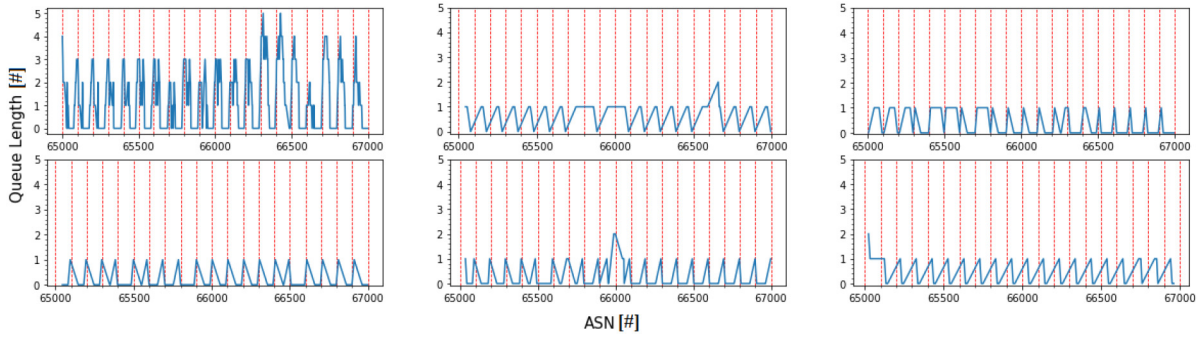


Fig. 4. Packet QU of devices in the 6TiSCH network.

ALABAMO [32] is a load-balancing method based on MRHOF [33] that combines ETX and workload metrics. ALABAMO selects a parent by path calculation using ETX. ETX is an intuitive metric for link evaluation. However, true ETX cannot be measured until communications take place. Therefore, THE ETX-based parent selection is used to avoid bad paths, and ETX is classified as a reactive metric.

Queue-utilization (QU)-RPL [18] uses OF to efficiently handle heavy traffic loads. The QU of the parent nodes serves as a metric—the queue level increases rapidly in congested devices. The authors argued that efficiency of changing the device parent with a high QU would not be enough for load balancing, and that it is necessary to propagate the high QU information to induce the child devices to migrate to another parent. Similarly, a congestion detection and control method that uses the QU of each device and its parent as a metric has been proposed [34]. The method facilitates load balancing by inducing parent changes according to the degree of congestion in various traffic environments. These techniques yield performance improvements in heavy-traffic networks. However, they do not consider the characteristics of TSCH, which is very efficient for IIoT networks.

The QU of six randomly selected devices in a 6TiSCH network simulation comprising 20 nodes is shown in Fig. 4. In the simulation, the size of the TSCH slotframe is 101, separated by a red vertical line in the graph. The devices use MSF link scheduling. In the example presented in Fig. 4, the QU mostly returns to zero at each slotframe. In the case of a stable 6TiSCH network, because traffic is completely resolved in every slotframe, it is difficult to assert that this is a problem, even if the QU is high. In the 6TiSCH network, the QU metric can be used to detect whether a problem occurs through the statistics of the overall state of slotframe. Additionally, the QU has limitations of being a reactive metric that detects and responds to problems.

The network interface average power (NIAP) metric [35] supports a load-balancing method that provides a software solution for estimating energy consumption and calculating the energy costs for each path. This energy metric has properties similar to QU, and it represents various other metrics related to traffic load, including link quality, workload, and congestion. However, because energy consumption is an indirect metric, it takes time to be reflected, and its effect is diluted as a result.

Yang *et al.* [19] asserted that IIoT network devices have sociality. Consequently, they introduced a method that utilizes this feature called social-cognitive (SC)-RPL. SC-RPL creates a routing path that reduces network delay and balances the load. The nodes in the SC-RPL network broadcast packets, and all neighboring nodes in the transmission range of the sender can receive them. Each node decides whether or not to transmit based on its social conditions. However, the authors did not consider the characteristics of TSCH, and their method based on broadcasting is not suitable for IIoT networks that handle heavy traffic loads.

Many conventional RPL studies presuppose an asynchronous MAC protocol [21], [30], and they do not consider TSCH MAC in heavy traffic-load scenarios. In particular, the overhead of the 6TiSCH network when changing the routing path has not been considered. The conventional reactive methods have limitations in the 6TiSCH network; therefore, a new proactive load-balancing method for IIoT is required.

### III. PROBLEM STATEMENT

This section describes the characteristics of the load-balancing method required for the 6TiSCH network and the limitations of the conventional metric. We also analyze the bandwidth problem that can occur with TSCH MAC in heavy-traffic networks and describe how RPL should support TSCH MAC to solve it.

Although RPL supports various traffic patterns, this study considers periodic converge-cast traffic for sensor data transmission because most IIoT traffic includes sensor data from devices converging toward a sink, which may be a gateway or an application manager [11].

#### A. Load-Balancing in 6TiSCH Network

6TiSCH devices exchange control packets (i.e., 6P transactions) for TSCH cell allocations after path selection. Because this method of cell allocation is cascaded to the upper path to handle new traffic, path selection, and changes in a single 6TiSCH device cause management overhead across the entire path. When the RPL path changes, the cell allocation to the previous parent is discarded, and allocation to the new parent is re-established. Therefore, in a heavy-traffic network with numerous cell allocations, a 6TiSCH network requires significant overhead for parent changes. Assuming the worst case,

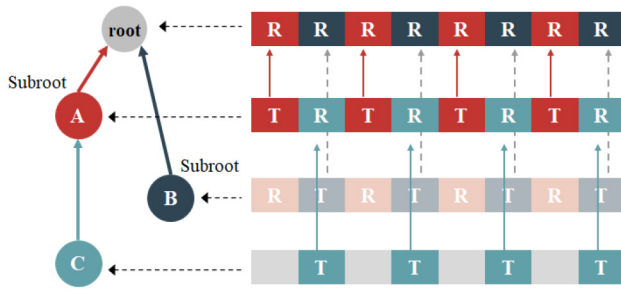


Fig. 5. Example of maximum throughput of the 6TiSCH network.

a parent change can lead to congestion of the new parent, which can lead to a cascade of parent changes throughout the network. This can lead to burst-6P transactions that further increase network congestion [36]. Until the 6P storm is stabilized, it is difficult for the network to operate normally, which is not suitable for IIoT networks that require reliability.

The methodology of the conventional load-balancing method involves the detection of congestion after parent selection and induces parent changes that can resolve or alleviate congestion. This reactive methodology is more critical to the IIoT network because it is difficult to predict how the network performance will be affected by the addition of a new device and an increase in traffic.

In a TSCH network, a link-scheduling method that allows contention, such as Orchestra [37], is used. However, link scheduling eliminates all contention within a network (e.g., TASA [15]), which can provide consistent throughput for each device. Therefore, the following problems are associated with heavy traffic in the contention-free TSCH network.

- 1) During link scheduling (i.e., the scheduling function), the method fails to predict incoming traffic.
- 2) During link scheduling, the method fails to allocate transmission bandwidth properly.
- 3) The bottleneck node runs out of bandwidth because all resources (i.e., cells) are used.

Case 1) can be solved by improving link scheduling and schedule allocation methods, and various studies [15], [16], [36] have been conducted in this regard. However, cases 2 and 3 in 6TiSCH networks have not been actively investigated. Case 2) includes the parent-change problem described above. When sufficient transmission bandwidth is not secured because the existing schedule is discarded after the parent change, the packet queue becomes full, and incoming packets are dropped. Case 3) causes resource depletion of the 6TiSCH network due to traffic-load concentrations, which is described in Section III-B. Link schedule and allocation methods for solving case 1) are out of the scope of this method; hence, we focus on cases 2) and 3).

### B. Bandwidth Problem in Heavy-Traffic 6TiSCH Network

In an IIoT network, most traffic must be delivered to the server through the root node. Therefore, the total network bandwidth and throughput are determined by the root node. Fig. 5 shows an example of a 6TiSCH network in which all bandwidths are allocated. The examples in this section

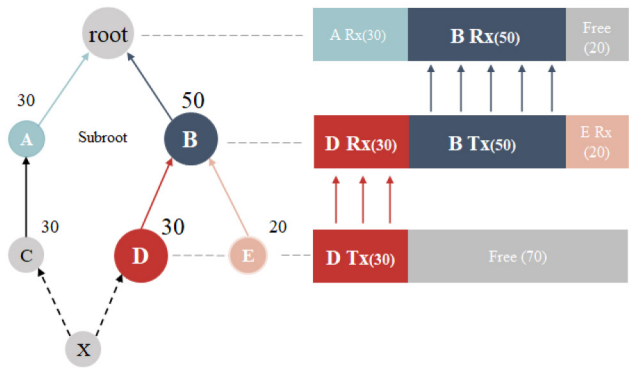


Fig. 6. Path selection problem in a heavy-traffic network.

exclude shared cells for convenience of description; however, the proposed method in the next section considers them. In this study, we presume a general gateway with a single radio interface, and the direct child nodes of the root, such as nodes A and B in Fig. 5, are called *subroots*. Assuming that there is no downlink traffic, the root node can use all available cells for Rx. However, subroot nodes A and B are allocated half the bandwidth of the root. Unlike the root node, the subroot node requires Tx cells to transfer data from its child nodes to the root. Therefore, the maximum subroot bandwidth is limited to half that of the root. However, the bandwidth imbalance of subroot nodes causes path selection problems in heavy traffic networks.

The path selection problem in a heavy-traffic network is illustrated in Fig. 6. The number next to each node represents the traffic percentage that it handles and, in the example, it is assumed that the root and subroot nodes do not generate traffic. The bar graph connected to the node shows its bandwidth status. Because the network-wide generated traffic load is 80% (i.e., traffic from nodes C–E), the root has an available bandwidth of 20%. Node B receives traffic loads corresponding to 30% and 20% from child nodes, and it uses all the bandwidth by allocating 50% for its delivery to the root node.

When node X, which generates 10% traffic load, participates in the network, it may select node C or D as the RPL parent. This is because from a local point of view, nodes C and D have sufficient available bandwidth at 70%. When node X selects node D as the RPL parent, node D cannot allocate additional bandwidth from parent-node B, and the link between nodes B and D will continuously drop 10% of the traffic load. In this case, conventional reactive RPL techniques will cause the migration of device D, which has a large rescheduling overhead. Moreover, owing to the parent change in node D, node A will have 70% traffic loads, and 20% will be dropped. Notably, the maximum bandwidth of the subroots is 50%, which is half that of the root node. The strategy of avoiding bad parents in conventional reactive RPL techniques can trigger cascading parent changes.

Accettura *et al.* [16] proposed a cell allocation method (i.e., *even/odd scheduling*) that considers the bandwidth imbalance problem. However, when traffic exceeding the maximum bandwidth is concentrated on a single path, the link-scheduling method cannot handle it. To resolve this, a metric that includes

information about the entire path leading to the root node is required. In this study, we propose the TA-RPL method that uses the number of available TSCH cells as a metric. This proactive approach prevents the concentration of traffic that exceeds the maximum bandwidth of the path and improves the efficiency of TSCH MAC by minimizing unnecessary path changes. Furthermore, because the cost of the entire path to the root can be determined, it is possible to estimate the impact of changes on the network.

#### IV. TA-RPL

##### A. Available Bandwidth Metric

Table II details the variables used to describe and define TA-RPL. The slotframe of the TSCH MAC consists of a predefined number of timeslots, and one communication is performed in each. Therefore, the throughput per second of the TSCH network is determined by the size of the timeslot, as follows:

$$nTps = 1000/IT. \quad (4)$$

The number of timeslots per second  $nTps$  can be calculated using (4). The length of the timeslot  $IT$  is generally set to 10 ms in the 2.4-GHz TSCH, and the  $nTps$  of the 6TiSCH network will have 100 timeslots. Thus, even if the root node uses all timeslots for reception, packet drop is inevitable when the total network traffic load exceeds 100 pps

$$mCpF = IF - (nC_{Adv} + nC_{Auto} + nC_{Sp}). \quad (5)$$

The 6TiSCH network uses control messages for management (i.e., EBs and DIOs), and a bandwidth portion is reserved for them. Excluding the bandwidth used for the control message, the maximum number of available cells per single slotframe,  $mCpF$ , can be calculated using (5). By subtracting the number of cells for EB and DIO ( $nC_{Adv}$ ), the number of autonomous cells for 6P ( $nC_{Auto}$ ) and the number of special-purpose cells ( $nC_{Sp}$ ) from the length of the slotframe ( $IF$ ),  $mCpF$  can be calculated. Special-purpose cells may be required, depending on the implemented network and tasks that may affect the TSCH timing (e.g., peripheral device control)

$$mCps = mCpF * nTps/IF \quad (6)$$

$$tpps = pps * nN. \quad (7)$$

The maximum number of available cells per second,  $mCps$ , can be calculated using (6), and the  $mCps$  of the root node is equal to the actual maximum throughput per second of the 6TiSCH network. The number of packets each node generates per second is denoted as  $pps$ , and  $nN$  is the number of nodes in the network. Using these parameters, the total traffic load of the network,  $tpps$ , can be calculated using (7)

$$tpps * aETX \leq mCps. \quad (8)$$

The total traffic load generated per second must be less than or equal to the  $mCps$  of the root node. However, because the IIoT wireless network has an unstable link, the average ETX of the network should be considered. Therefore, if (8) is not

TABLE II  
VARIABLES USED TO DESCRIBE AND DEFINE TA-RPL

Parameters	Description
<i>Network Parameters</i>	
$r$	Root node
$sr_i$	Subroot node $i$
$n_i$	Node $i$ (for non-root/non-subroot nodes)
$nN$	Number of non-root nodes in the network
$Child_x$	Set of child nodes of node $x$
$P_x$	Set of parent candidate nodes of node $x$
$P(x)$	Preferred parent node of node $x$
$N_x$	Set of neighbors of node $x$
<i>TSCH Parameters</i>	
$IT$	Timeslot length (ms)
$IF$	Slotframe length (timeslots)
$nC_t$	Number of cells that match type $t$ <i>type : Adv (Advertisement), Auto (Autonomous), Sp (Special-purpose)</i>
$nTx(x)$	Number of Tx cells of node $x$
$nRx(x)$	Number of Rx cells of node $x$
$CellReq(x)$	Number of cells required for node $x$ to send all traffic to its parent node
<i>Traffic-Aware RPL Parameters</i>	
$nTps$	Number of timeslots per second
$D$	Ratio of downlink traffic to total traffic load
$mCpF$	Maximum number of available cells per slotframe
$mCps$	Maximum number of available cells per second
$mB_x$	Maximum available bandwidth of node $x$ for the uplink traffic
$B_x$	Available bandwidth of node $x$ for the uplink traffic
$M(x)$	Metric calculation function for node $x$ in traffic-aware RPL
$R(x)$	Evaluation function for node $x$ in traffic-aware RPL
$\sigma(x)$	Probability threshold of node $x$ for determining parent change
<i>General Parameters</i>	
$pps$	Number of packets each node generates per second
$tpps$	Total traffic load of the network
$t(x)$	Amount of traffic generated by node $x$
$T'(x)$	Estimated total traffic load of node $x$
$aETX$	Average ETX of the network
$ETX_{x,y}$	ETX of the link between node $x$ and node $y$
$Rank(x)$	RPL rank of node $x$
$h(x)$	Hop count from node $x$ to the root node

satisfied, the 6TiSCH network is not perfectly reliable. Using this approach, it is possible to obtain the deterministic throughput limit of the 6TiSCH network and to design a delicate IIoT wireless network that meets service requirements.

In this study, we focus on uplink traffic, which constitutes most IIoT traffic patterns. However, because downlink traffic for facility control is also important, we include a basic consideration for downlink traffic. Considering the deterministic characteristics of the 6TiSCH network, a scenario for management of the network QoS using a predefined uplink/downlink

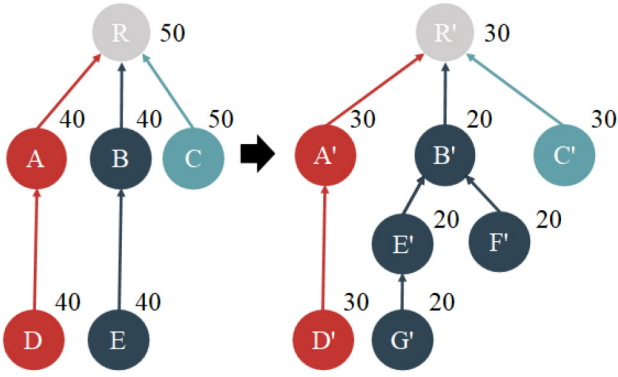


Fig. 7. Example of available bandwidth as path cost.

traffic ratio can be assumed

$$mB_r = mCpF * (1 - D). \quad (9)$$

Equation (9) shows the maximum available bandwidth for the uplink traffic of the network.  $D$  is a predefined downlink traffic ratio and  $mB_r$  indicates the number of cells of root node that can be allocated to the Rx cells for uplink traffic

$$mB_{sr_i} = \lfloor mB_r / 2 \rfloor. \quad (10)$$

Equation (10) provides the calculation of the maximum available bandwidth of the subroot nodes, which is half the maximum bandwidth of the root. Moreover, it is assumed that all factors of (5) and (9) are determined before the network is formed and can be propagated to the network via control messages, such as EB and DIO. Thus, all nodes in the network can independently compute  $mB$ .

The calculation of the available bandwidth  $B$  according to the attribute of each node is as follows:

$$B_r = mB_r - nRx(r) \quad (11)$$

$$B_{sr_i} = \min(mB_{sr_i} - nRx(sr_i), B_r) \quad (12)$$

$$B_{n_i} = B_{P(n_i)}. \quad (13)$$

The root node  $r$  calculates the available bandwidth using (11), which then subtracts its Rx cells from the maximum network bandwidth. The subroot node  $sr_i$  can also calculate the available bandwidth by subtracting its Rx cells from the  $mB_{sr_i}$ . However, because the available bandwidth of the subroots depends on the root, if the available bandwidth of the root is smaller, the same would hold for the value of the subroot, as represented by (12). All nodes that belong to the subtree of subroot  $n_i$  assume the parent value according to (13). Therefore, the  $B$  value refers to the bandwidth of the entire path, not the available bandwidth of a specific node.

A network using the available bandwidth is shown in Fig. 7 as the path cost. In the network, the maximum bandwidth is 100, and all nodes except the root generate ten traffic loads each. The number next to each node represents the available bandwidth,  $B$ , of the node. Subroot node A has an available bandwidth of 40 according to (12), and child node D has a value given by (13). The root node handles 50 traffic loads of the network and has an available bandwidth of 50. After the new nodes F' and G' join the network, node B' and

the root, R', have available bandwidths of 20 and 30, respectively. Nodes A' and C' have sufficient available bandwidth, but the bandwidth limit of the path is determined by the root node. Therefore, nodes A' and C' follow the  $B_r$  of the root as given by (12).

Each node can determine the path cost based on the available bandwidth metric that propagates through the DIO message, and the problem of selecting a path with insufficient bandwidth is mitigated. However, if only the available bandwidth is used as a metric, the nodes in the same subtree have the same priority. Therefore, an additional metric is required.

### B. ETX Metric in TA-RPL

ETX is a key metric for calculating the cell requirement in link scheduling as well as in routing methods. In an IIoT wireless network, the possibility of packet loss must be considered, and most link-scheduling methods adopt *over-provisioning*, considering such losses

$$\text{CellReq}(n_i) = \left( t(n_i) + \sum_{\text{Child}_{n_i}}^c t(c) \right) * \text{ETX}_{n_i, P(n_i)}. \quad (14)$$

Equation (14) shows the concept of simple over-provisioning. Node  $n_i$  must handle and transmit the sum of the traffic it generates, including the traffic received from the child nodes. By representing the ETX for the preferred parent of node  $n_i$  in the total traffic, the cell requirement considering packet loss can be calculated. Therefore, the selection of a parent node with a low ETX helps reduce the required bandwidth of the TSCH.

In Fig. 6, when node X selects the preferred parent, the parent candidate nodes, C and D, have already measured and know the ETX of the preferred parents A and B. To improve the decision making, the proposed method uses the ETX of the grandparent node as a secondary metric. In IIoT wireless networks, nodes close to the root node always handle more traffic than do their child nodes. Therefore, improvement of the bandwidth efficiency of the upper node can improve the overall efficiency of the network.

This policy can cause parent selection to focus on specific nodes having good parent links. However, even if all traffic in the subtree is directed to a single node, the limit is eventually determined by the subroot node. Note that ETX is the secondary metric in the proposed technique. Therefore, load concentration in a subtree does not significantly affect the overall efficiency of the network, and the load cannot exceed that of the subroot.

### C. Rank and Metric Calculation

The proposed method uses the available bandwidth and ETX of the parent node as metrics, and this information must be propagated to child nodes via DIO messages. Metrics are not encoded in the rank information but are delivered using a *metric container* [10], an option field of the DIO message

$$\text{Rank}(n_i) = h(n_i) + 1. \quad (15)$$

Parameter  $h(n_i)$  represents the hop count of node  $n_i$  and the rank calculation follows (15). Although the hop-count metric is simple, it intuitively includes distance information for the entire path, and there are numerous advantages associated with the reduction of the number of transmitted packets. The proposed method adopts the hop count used in many previous studies as the rank calculation metric

$$M(n_k) = (IF - B(n_k)) + ETX_{n_k, P(n_k)}. \quad (16)$$

Equation (16) is a calculated metric used to evaluate the parent candidate node  $n_k$ . Node  $n_k$  propagates its available bandwidth  $B(n_k)$  and the ETX of its parent node using DIO.  $B$  is then subtracted from the slotframe length so that the metric function decreases monotonically. Because the ETX metric is used to evaluate the path inside the subtree, it is designed as a summation.

The unit of available bandwidth is a cell, and ETX refers to the number of cells required for packet transmission. All nodes have an ETX limit during parent selection [27], and the value does not exceed three. Therefore, the ETX metric has a low priority. The proposed method prioritizes load balancing between subroots based on the available bandwidth metric and improves the efficiency within the subtree based on the ETX metric.

#### D. Parent Selection and Change

Equation (17) defines the parent candidate set  $P_{n_i}$  of node  $n_i$

$$P_{n_i} = \{n_k \in N_{n_i} | h(n_k) < h(n_i), ETX_{n_i, n_k} < \tau\} \quad (17)$$

where  $P_{n_i}$  is defined as neighbors having a lower hop count than node  $n_i$  in  $N_{n_i}$ , which is the neighboring set of node  $n_i$ , whose ETX does not exceed  $\tau$ . Threshold  $\tau$  for parent selection is set to three according to Table I presented for RFC8180 [27]

$$P(n_i) = \min\{R(p_k) | p_k \in P_{n_i}\} \quad (18)$$

$$R(n_k) = \{h(n_k) * (IF + \tau)\} + M(n_k). \quad (19)$$

The preferred parent  $P(n_i)$  of node  $n_i$  is selected according to (18) and is the node with the lowest value determined using the evaluation function (19). The evaluation function for parent candidate node  $n_k$ ,  $R(n_k)$  is calculated using the metric calculation function (16) with a hop count of  $h$ . To suppress loops and minimize the number of transmissions, the hop count is multiplied by the maximum value of  $M$ .

The node receiving a better evaluation than the existing preferred parent node according to (19) is changed to a new preferred parent. However, in a 6TiSCH network, changing the RPL parent has a large overhead. Therefore, each node should consider the impact of parent change on the network, and the preferred parent should not be changed for a slight benefit. First, the traffic load that will migrate to the new parent must be considered

$$T'(n_i) = nTx(n_i) / ETX_{n_i, P(n_i)}. \quad (20)$$

Equation (20) can be used to estimate the total traffic of node  $n_i$  that represents the ETX to the parent.  $nTx(n_i)$  indicates

the number of Tx cells of node  $n_i$ . Note that  $nTx$  is the number of over-provisioned cells considering the ETX

$$R(P(n_i)) - T'(n_i) > R(n_k) \quad (21)$$

$$B(n_k) > T'(n_i). \quad (22)$$

Equations (21) and (22) represent the conditions for parent change. When node  $n_i$  changes the parent, the available bandwidth of the old parent increases according to (20), and when the available bandwidth of the new candidate node  $n_k$  is greater, the parent is changed. The available bandwidth is converted into (16) and (19) in the parent evaluation step, and (21) represents this change. In addition, each node must select a preferred parent that satisfies (22) to select a path with sufficient bandwidth.

Because all nodes belonging to the same subroot have the same available bandwidth metric, there is a high probability of determining a parent change when this change is required. In this case, a herding effect may occur in which multiple nodes change their parents simultaneously. To solve this problem, the proposed method uses a stochastic parent change

$$\sigma(n_i) = h(n_i) * \varepsilon \quad (23)$$

$$\sigma(n_i) > \rho \mid \rho \in [0, 100]. \quad (24)$$

The probability threshold  $\sigma(n_i)$  for determining the parent change is calculated using (23) with the hop count of node  $n_i$ . When the random variable  $\rho$  having a range of [0, 100] satisfies (24), the preferred parent is changed. In a 6TiSCH network, a node close to the root must handle more traffic and allocate more cells. Therefore, a node with a lower hop count should use a lower change probability, and the hop count coefficient  $\varepsilon$  is 10 in the proposed method.

## V. PERFORMANCE EVALUATION

### A. Simulation

The 6TiSCH Simulator [38] was used to evaluate TA-RPL for a large number of nodes and various topologies. The simulator implemented key technologies for the IoT network in Python, including IEEE 802.15.4 TSCH, MSF, 6P, CoJP [39], 6LoWPAN, and RPL. Most of the simulator settings follow the recommendations of IETF RFC 8180 [27]. Table III lists the relevant parameters. Each node configures the network based on OF0 (ETXOF) using ETX as a metric, QU-RPL (QU) using QU, NIAP using energy consumption as a metric, and TA-RPL (TAR), which is the proposed method.

ETXOF is set using the values shown in Table I and follows the recommendations in [27]. Specifically, the ETXOF scenario is the minimal standard configuration proposed by the IETF 6TiSCH WG and has important meaning as a baseline of the 6TiSCH network.

QU-RPL is selected as a comparative method because the suitability of QU as a metric in the reactive method is evident. Additionally, the QU method used in the simulation was modified to record and use the lowest value during the slotframe period.

NIAP uses energy consumption, which is a representative metric of load balancing. Moreover, unlike reactive



TABLE III  
PARAMETERS OF THE 6TiSCH SIMULATOR

Features	Description
RPL	EXTOF, QU, NIAP, TAR
Scheduling function	MSF
Slotframe length	101 timeslots
Timeslot length	10 ms
MaxRetries	5
numMotes	100
Topology size	$2 \times 2$ km
PDR mean	80%
Pkt queue length	10
numSlotframePerRun	4,000
numRuns	20
Propagation model	Pister–Hack model

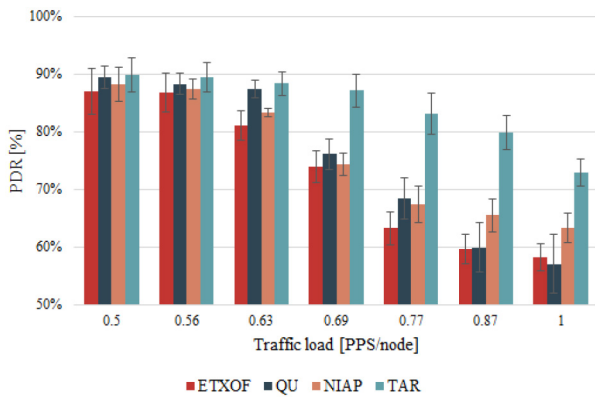


Fig. 8. Data-packet PDR versus traffic load.

approaches, NIAP calculates the path cost, and proactively balances the traffic load.

The link between each node was established based on the Pister–Hack model [9]. The model is the default propagation model used in the 6TiSCH Simulator and tunes the RSSI levels to better match empirical results. The RSSI is randomly selected to be between the values predicted by the Friiss model.

Each node transmitted packets at a predetermined period when a link schedule was allocated after synchronization. The simulation was repeated 20 times for each scenario. To evaluate the performance of the proposed method under different traffic loads, various performance indicators were measured by changing the *pps* of each node. Using (5) and (7), the theoretical network bandwidth required for the total traffic can be estimated. For example, if 99 nonroot nodes generate 1 pps of traffic, a bandwidth of 99 cells per second is required; considering ETX-based over-provisioning, the required bandwidth is approximately 124 cells per second. However, the maximum available bandwidth based on (5) is 98 cells, and given that traffic exceeding 100% of the bandwidth is generated, packet drops are unavoidable. The simulation evaluates each method by changing the traffic load from 0.5 to 1 pps. The error bars in all the graphs indicate 95% confidence intervals, except for Fig. 14.

The packet delivery ratio (PDR) of the data packet according to the traffic load is shown in Fig. 8. For all methods,

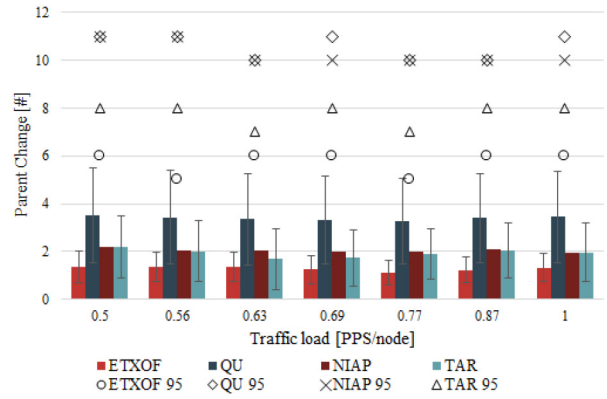


Fig. 9. Average number and 95% percentile of parent changes versus traffic load.

the performance decrease as the traffic load increases. Performance degradation is insignificant at low traffic loads (i.e., 0.5–0.63 pps), but the PDR of ETXOF and QU rapidly decreases from 0.69 pps. As a proactive method, NIAP provides relatively good performance even when the traffic exceeds 0.77 pps. However, because the energy consumption metric indirectly reflects traffic, the performance improvement is limited. Although there may be differences that depend on the number of nodes or link quality, it is evident that the load-balancing problem of the 6TiSCH network is exacerbated around the 0.69 pps setting, which generates approximately 70% of the traffic load compared with the maximum available bandwidth. As the traffic load increases, the maximum available bandwidth of the network is utilized; thus, reduction of the PDR is inevitable. The TA-RPL method shows superior performance compared with the other methods, even below a traffic load of 1 pps. This is attributable to the TA-RPL method minimizing the number of path changes and selecting a path that can improve bandwidth utilization by using the available bandwidth metric. We describe the operation and impact of the method through the analysis of other performance factors.

The average number and the 95% percentile of RPL parent changes according to the increase in the traffic load are shown in Fig. 9. The results show a similar trend regardless of the change in traffic load. 0.5 pps per node, which is the lowest load in the simulation, is calculated as a load of approximately 62% of the total bandwidth, considering the average ETX of simulation. This result signifies that each method works for load balancing in all heavy-traffic scenarios. The bottleneck node causes most parent changes; hence, the average value is low, but the error bar is long. The average cannot show the state of the bottleneck node, but the 95% percentile can be used to estimate the number of changes in the bottleneck node. Additionally, the total number of parent changes has a similar trend. The QU and TA-RPL methods were measured to change parents approximately 425 and 300 times during the simulation.

ETXOF and QU methods have reactive policies. For ETXOF, a node determines the parent change only when the ETX of the parent node is sufficiently high. The nodes using ETXOF do not trigger parent changes easily and are slow to

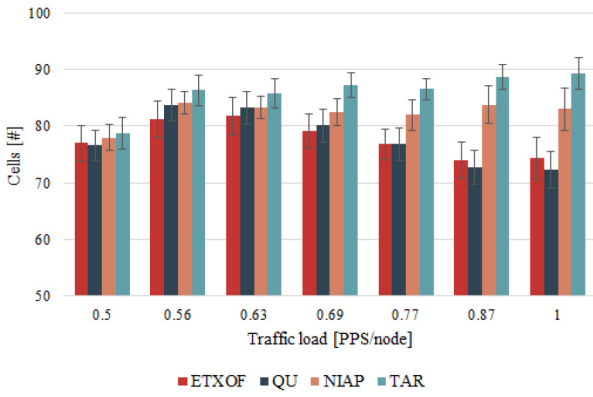


Fig. 10. Number of cells in the root node versus traffic load.

respond. However, because the QU method adapts to changes in queue level, it changes its parent relatively frequently. On the other hand, TA-RPL and NIAP have a proactive policy and select the parent based on the information available. Therefore, the average values of the two methods are similar. However, the characteristics of the metric produce different 95% percentile values, and TA-RPL provides high PDR performance while reducing the number of parent changes.

The number of cells allocated by the root node according to traffic changes is shown in Fig. 10. In a 6TiSCH network, the maximum throughput is the same as the allocated bandwidth of the root node, and the traffic throughput of each method can be estimated based on this result. If the traffic load is not efficiently distributed, many packets will be dropped at the subroot node, and the number of allocated cells of the root node will decrease. For a light traffic-load setting, each method exhibited a slight difference up to 0.63 pps, after which the number of allocated cells increased with the load. However, at a traffic load of 0.69 pps or greater, the performance of the ETXOF and QU methods decreased substantially. This result shows that the traffic load is not properly distributed and the problem in Section III-B occurs at traffic loads of 0.69 pps or more. The NIAP method provides the path cost mechanism and energy consumption metric. The characteristics of NIAP do not exacerbate the 6TiSCH bandwidth problem, even under heavy traffic loads. However, unlike the TA-RPL, which increases the number of allocated cells as the traffic load increases, NIAP has a lower limit of increase because it does not reflect cross-layer characteristics.

The proposed method aims to prevent the concentration of traffic and minimize unnecessary changes. Figs. 8–10 show that the proposed method directly and indirectly improves the efficiency of TSCH MAC. However, the functioning of TA-RPL for each purpose is explained by analyzing the records of dropped packets.

The number of dropped packets according to the traffic load is shown in Fig. 11. Colored areas indicate packet drops caused by the full Tx queue, and empty areas indicate drops caused by maximum retransmission. The performance is similar to that of the data-packet PDR. However, for the 1 pps setting, the PDR of ETXOF and TA-RPL has a difference of 125%, but the number of dropped packets has a difference of 216%.

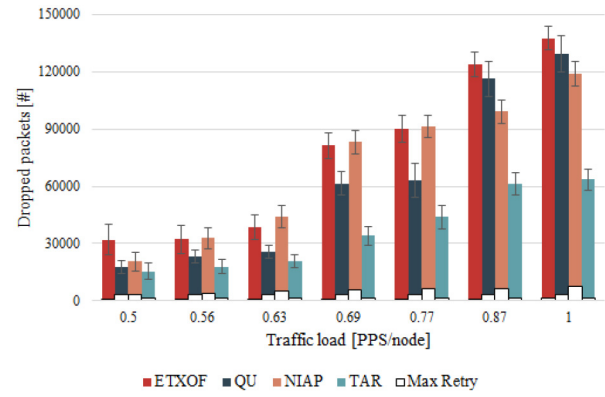


Fig. 11. Number of dropped packets versus traffic load.

Nodes continuously drop packets owing to paths having insufficient available bandwidth, and it can be determined that more retransmissions and control messages have occurred. The drop caused by the maximum retransmission is a predictable link loss and accounts for a very small percentage. Therefore, only packet drops caused by a full Tx queue is considered in the following.

The distribution of dropped packets over time for 0.5, 0.69, and 1 pps, respectively, is shown in Fig. 12. We noted the extreme points in these results. In the 6TiSCH network, when a node changes its routing path, many packets are temporarily dropped, and it appears as a minimum point on the graph. Thereafter, during link scheduling, the number of dropped packet decreases when bandwidth is secured, and it appears as a maximum point on the graph. Note that the trend line in the graph only connects and filters the midpoints of each point for illustrative purposes. Not all extreme points indicate this.

Generally, ETXOF, which does not consider load balancing, has a small number of extreme points regardless of the traffic load. By contrast, the QU, NIAP, and TA-RPL methods generate a larger number of extreme points. The results presented in Figs. 8, 10, and 11 show that the QU method performed meaningful load balancing up to 0.77 pps, which can also be confirmed in Fig. 12. However, at 1.0-pps traffic load, the QU method draws a gradual trend line. NIAP provides slightly lower performance than the other methods in scenarios of 0.69 pps or less; however, the decrease in performance is small at very heavy traffic loads. This trend can also be confirmed in Fig. 12. Although not all dropped packets imply the overhead caused by the parent change, this result can show the effect of the problem and the operation of each metric.

The ratio of dropped packets per hop is shown in Fig. 13. In most scenarios, more than 90% of packets are dropped from the subroot, which is a one-hop node. Because the load balancing methods induce a parent change at the child node of the bottleneck, packet drops occurring in more than two hops can be inferred as the impact of the parent changes. First, at a relatively low traffic load (i.e., 0.5 and 0.56 pps), the load balancing methods drop about 10% of packets over two hops. This means that each method changed the parent to distribute the traffic load, and the results of dropped packets and PDR show that the overhead is small. Second, at a moderate traffic

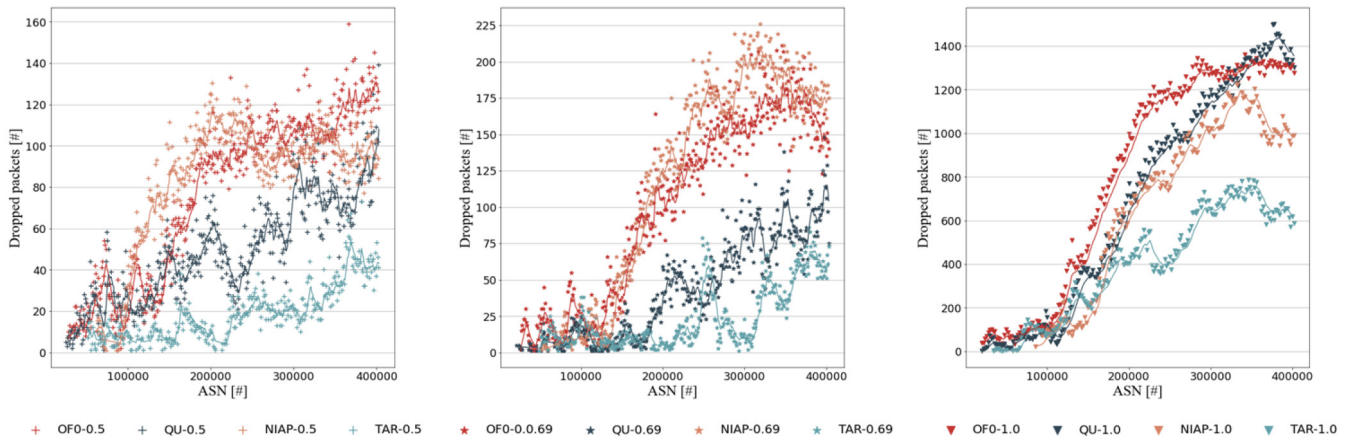


Fig. 12. Distribution of dropped packets over time (left: 0.5 pps, middle: 0.69 pps, and right: 1.0 pps).

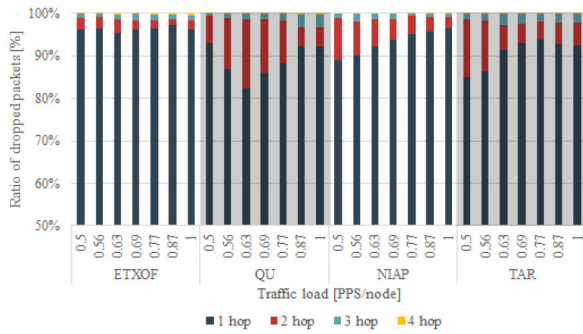


Fig. 13. Ratio of dropped packets per hop versus traffic load.

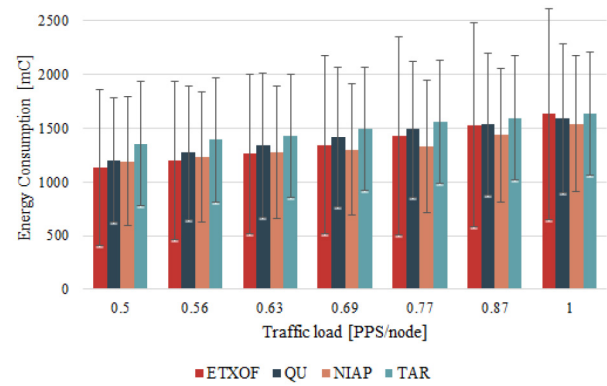


Fig. 14. Average energy consumption versus traffic load.

load (i.e., 0.63 to 0.77 pps), the QU method has a larger drop ratio of two hops or more. Considering that the performance of PDR, bandwidth, and dropped packets of the QU method decrease rapidly in those scenarios, it can be inferred that this is the overhead of frequent parent changes. Lastly, the heaviest traffic loads (i.e., 0.87 and 1 pps) exceeding the total throughput of the network significantly increase the number of dropped packets of the subroot, and the weight of the overhead is lowered owing to parent change. The NIAP method shows a similar trend as TA-RPL; however, the number of dropped packets of the subroot is larger in all scenarios. Nevertheless, the number of dropped packets in TA-RPL is half that of the other methods.

The average energy consumption of all the nodes is shown in Fig. 14. TA-RPL consumes slightly more energy than the other methods, which may be caused by the difference in traffic throughput. TA-RPL facilitates efficient load-balancing performance and can handle more traffic at the root node. In the simulation at 1-pps traffic load, TA-RPL handled about 200 000 packets, which is approximately 25% more than the other methods. Considering the results of PDR and bandwidth, the throughput of the ETXOF, QU, and NIAP methods decreases significantly as the traffic load increases. Although the actual packet throughput is greatly reduced, energy consumption is not. This is analyzed as an overhead caused by packet drop and control messages. The error bars in the graph denote the standard deviation of the energy consumption of all

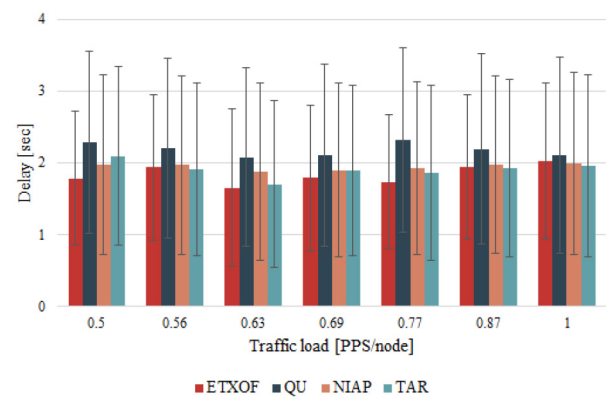


Fig. 15. End-to-end delay versus traffic load.

the nodes. TA-RPL consumes slightly more energy than the other methods, but the standard deviation is reduced owing to the efficient load balancing. Although the standard deviation of the ETXOF and QU methods greatly increases as the traffic load increases owing to an imbalance in throughput, the standard deviation of TA-RPL remains very low. The NIAP method shows good performance in all scenarios because it directly optimizes the network to minimize energy consumption.

The end-to-end delay of each method according to the traffic load is shown in Fig. 15. In all scenarios, the performance

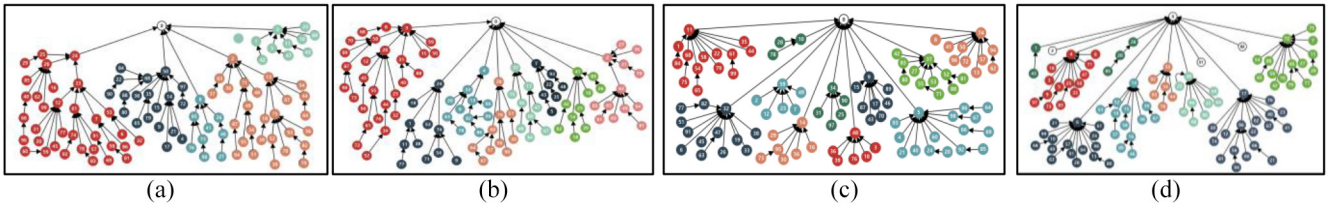


Fig. 16. Topology snapshot of each method. (a) ETXOF. (b) QU. (c) NIAP. (d) TAR.

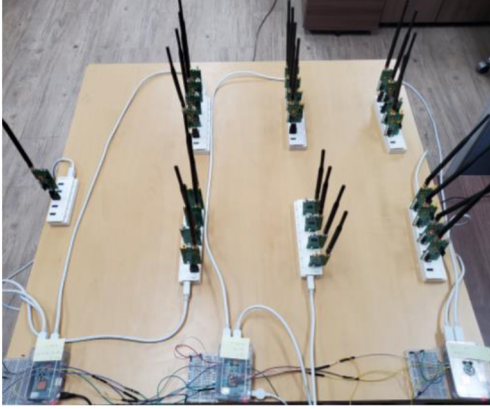


Fig. 17. Testbed using 25 motes based on the OpenWSN platform.

is almost the same, and the confidence intervals are similar. Although each method is slightly different, an error within the length of the slotframe (i.e., 1 s) is inconsequential. Furthermore, the error range slightly exceeds 1 s, which is the length of the slotframe, and it can be attributed to the hop distance. This result is obtained because the delay is recorded only for successful transmission. Furthermore, it is a characteristic of TSCH MAC that guarantees stable communications within the allocated bandwidth.

Topology snapshots of each method after 1-pps traffic load simulation are shown in Fig. 16. This snapshot implies the connection of the RPL network rather than the physical location. In ETXOF, two subroots handle excessive child nodes. Although the QU method constructs a somewhat balanced network, it appears that it was not balanced until the simulation was finished during the running time where the snapshot was taken. The NIAP and TAR methods construct balanced topologies with parent selection based on path cost.

### B. Testbed Evaluation

OpenWSN [40], RIOT [41], and Contiki-NG are open-source projects that implement the 6TiSCH network stack and provide compatible firmware for various hardware platforms. To verify the performance trend of the proposed method, a testbed comprising the OpenWSN platform was used. In this study, OpenMote-B hardware was used, and a network was constructed with 25 motes, including the root node. Because the number of nodes decreased more than the simulation, the experiment was performed by increasing the traffic load. The runtime of each experiment was 60 min, and each scenario was repeated ten times.

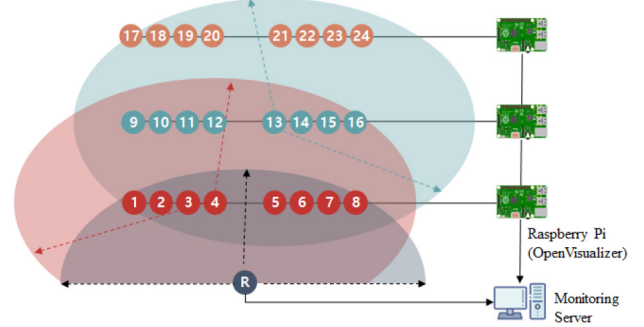


Fig. 18. OpenWSN testbed structure.

The testbed used for the experiment is shown in Fig. 17, and Fig. 18 represents the structure of the testbed. The nodes were placed at a maximum distance of three hops from the root node, with eight nodes in each hop. To limit the communication range of each hop, the nodes used three different dedicated channels for each hop. Channel 11 was used for shared cells for EB and DIO; however, to construct a three-hop topology, control packets from a distance of two hops were discarded. As shown in Fig. 18, the nodes of each hop were linked using OpenVisualizer running on Raspberry Pi.

The PDR, number of allocated cells, and average number of parent changes for the testbed evaluation are illustrated in Figs. 19–21, respectively. The overall performance is similar to that of the simulation. Based on (5) and (7), a 4-pps traffic load requires approximately 98% of the network bandwidth. The ETXOF method has a slight performance degradation compared with the simulation result, and the TA-RPL method has slightly better performance.

The testbed evaluation used a smaller number of nodes compared with the simulation. This feature reduces the wasted bandwidth owing to over-provisioning and facilitates easier load balancing. However, because each node must handle more traffic on average, conventional RPL methods that gradually adapt to traffic are expected to have a larger overhead of parent changes for load balancing. For the same total traffic amount, over-provisioning for a light traffic load generated by a large number of nodes requires more bandwidth than does a heavy traffic load generated by a small number of nodes.

## VI. CONCLUSION

IIoT wireless networks must direct heavy traffic throughout industrial sensors and facilities. This article introduced a TA-RPL method that provides efficient load-balancing performance using cell allocation information in a 6TiSCH

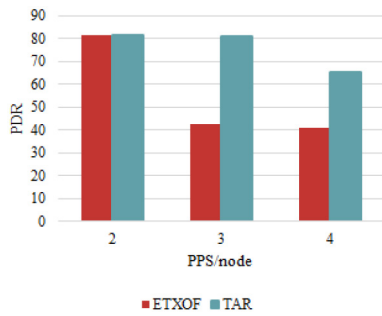


Fig. 19. PDR on testbed.

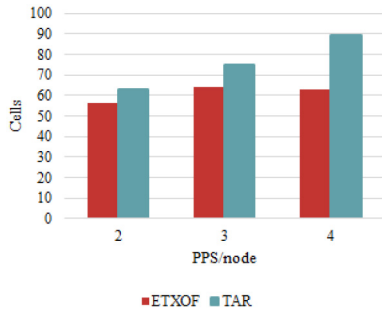


Fig. 20. Cell allocation on testbed.

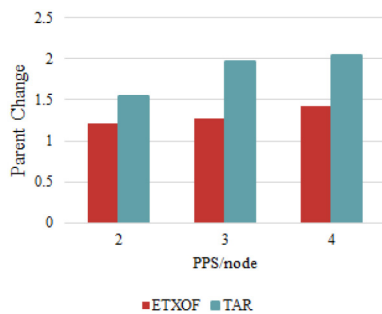


Fig. 21. Parent changes on testbed.

network. The proposed method facilitates high data-packet PDR, even when most of the network bandwidth is utilized. It further contributes to balanced energy consumption and improved network stability. The proposed method prevents the concentration of traffic that exceeds the maximum bandwidth of the path and improves the efficiency of TSCH MAC by minimizing unnecessary path changes. The cross-layer approach that utilizes the available bandwidth information of the TSCH MAC improved the data-packet PDR by up to 31% and the bandwidth utilization by up to 20%. Additionally, the proposed method provides available bandwidth for each path. Based on these features, it is possible to predict the impact of network events (i.e., node joining, node leaving, path change, etc.)

As shown in Fig. 10, TA-RPL did not achieve a bandwidth that matches the traffic load. This is a limitation of link scheduling. Because the proposed method acquires traffic information based on the bandwidth measured by the scheduling function, the traffic-awareness performance depends on the traffic prediction accuracy of the MSF. Conservative traffic

prediction can provide network stability but wastes bandwidth, and TA-RPL is affected by these characteristics.

For TA-RPL to achieve optimal performance and improve the deterministic characteristics of the 6TiSCH network, such a consideration is required; this is left for future research.

## REFERENCES

- [1] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, 2012.
- [2] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
- [3] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *J. Netw. Comput. Appl.*, vol. 97, pp. 48–65, Nov. 2017.
- [4] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Apr. 2014.
- [5] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
- [6] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.
- [7] G. Micoli *et al.*, "ASAP: A decentralized slot reservation policy for dynamic 6TiSCH networks in Industrial IoT," in *Proc. 4th Conver. Internet Things (C-IoT)*, 2019, pp. 1–6.
- [8] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: Deterministic IP-enabled Industrial Internet (of Things)," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 36–41, Dec. 2014.
- [9] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: Why channel hopping makes sense," in *Proc. 6th ACM Symp. Perform. Eval. Wireless Ad Hoc Sensor Ubiquitous Netw.*, 2009, pp. 116–123.
- [10] T. Winter *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Eng. Task Force, RFC 6550, Mar. 2012.
- [11] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial routing requirements in low-power and lossy networks," Internet Eng. Task Force, RFC 5673, 2009.
- [12] "Cisco IPv6 mesh for field area networks." [Online]. Available: <https://developer.cisco.com/docs/ipv6-mesh/> (Accessed: Mar. 9, 2022).
- [13] X. Vilajosana, T. Watteyne, M. Vucinic, T. Chang, and K. S. Pister, "6TiSCH: Industrial performance for IPv6 Internet-of-Things networks," *Proc. IEEE*, vol. 107, no. 6, pp. 1153–1165, Jun. 2019.
- [14] T. Watteyne, M. Palattella, and L. Grieco, "Using IEEE 802.15.4e time-slotted channel hopping (TSCH) in the Internet of Things (IoT): Problem statement," Internet Eng. Task Force, RFC 7554, 2015.
- [15] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial IoT applications using IEEE 802.15.4 e TSCH," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct. 2013.
- [16] 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.
- [17] 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, pp. 100–114, Feb. 2018.
- [18] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 964–979, Apr. 2017.
- [19] Z. Yang, H. Liu, J. Liu, H. Yang, W. Yang, and J. Li, "SC-RPL: A social cognitive routing for communications in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7682–7690, Dec. 2020.
- [20] E. Aydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsström, and M. Gidlund, "A central intrusion detection system for RPL-based Industrial Internet of Things," in *Proc. 15th IEEE Int. Workshop Factory Commun. Syst. (WFCS)*, 2019, pp. 1–5.
- [21] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2502–2525, 4th Quart., 2017.
- [22] A. Dunkels, "The contikimac radio duty cycling protocol," SICS, Kista, Sweden, Rep. SICS T2011:13, 2011.

- [23] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LRWPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4-2015, Oct. 2015.
- [24] R. T. Hermeto, A. Gallais, and F. Theoleyre, “Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey,” *Comput. Commun.*, vol. 114, pp. 84–105, Dec. 2017.
- [25] D. Barthel, J. Vasseur, K. Pister, M. Kim, and N. Dejean, “Routing metrics used for path calculation in low-power and lossy networks,” Internet Eng. Task Force, RFC 6551, Mar. 2012.
- [26] P. Thubert, “Objective function zero for the routing protocol for low power and lossy networks (RPL),” Internet Eng. Task Force, RFC 6552, Mar. 2012.
- [27] X. Vilajosana, K. Pister, and T. Watteyne, “Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) configuration,” Internet Eng. Task Force, RFC 8180, May 2017.
- [28] T. Chang, M. Vucinic, S. Duquennoy, D. Dujovne, and X. Vilajosana, “6TiSCH minimal scheduling function (MSF),” Internet Eng. Task Force, Fremont, CA, USA, draft-ietf-6tisch-msf-06, Aug. 2019.
- [29] Q. Wang, X. Vilajosana, and T. Watteyne, “6TOP protocol (6P),” Internet Eng. Task Force, RFC 8480, Aug. 2018.
- [30] D. Pancaroglu and S. Sen, “Load balancing for RPL-based Internet of Things: A review,” *Ad Hoc Netw.*, vol. 116, May 2021, Art. no. 102491.
- [31] B. Safaei, A. Mahdi, H. Monazzah, and A. Ejlali, “ELITE: An elaborated cross-layer RPL objective function to achieve energy efficiency in Internet of Things devices,” *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1169–1182, Jan. 2021.
- [32] T. B. Oliveira, P. H. Gomes, D. G. Gomes, and B. Krishnamachari, “ALABAMO: A load balancing model for RPL,” in *Proc. Braz. Symp. Comput. Netw. Distrib. Syst. (SBRC)*, 2016, p. 9.
- [33] O. Gnawali and P. Levis, “The minimum rank with hysteresis objective function,” IETF, RFC 6719, 2012.
- [34] H. Farag, P. Österberg, and M. Gidlund, “Congestion detection and control for 6TiSCH networks in IIoT applications,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.
- [35] H. Pereira, G. L. Moritz, R. D. Souza, A. Munaretto, and M. Fonseca, “Increased network lifetime and load balancing based on network interface average power metric for RPL,” *IEEE Access*, vol. 8, pp. 48686–48696, 2020.
- [36] Y. Ha, and S. H. Chung, “Enhanced 6P transaction methods for industrial 6TiSCH wireless networks,” *IEEE Access*, vol. 8, pp. 174115–174131, 2020.
- [37] 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.*, 2015, pp. 337–350.
- [38] E. Municio *et al.*, “Simulating 6TiSCH networks,” *Wiley Trans. Emerg. Telecommun.*, vol. 30, no. 3, 2019, Art. no. e3494.
- [39] M. Vucinic, J. Simon, K. Pister, and M. Richardson, “Constrained join protocol (CoJP) for 6TiSCH,” Internet Eng. Task Force, Fremont, CA, USA draft-ietf-6tisch-minimal-security-15, Dec. 2019.
- [40] T. Watteyne *et al.*, “OpenWSN: A standards-based lowpower wireless development environment,” *Trans. Emerg. Telecommun. Technol.*, vol. 23, no. 5, pp. 480–493, 2012.
- [41] E. Baccelli *et al.*, “RIOT: An open source operating system for low-end embedded devices in the IoT,” *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4428–4440, Dec. 2018.