Ad Hoc Networks 174 (2025) 103843

Contents lists available at ScienceDirect

Ad Hoc Networks

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

RPL-TSCH cross-layer design for improve quality of service in LLNs of IIoT

Mehdi Zirak, Yasser Sedaghat *, Mohammad Hossein Yaghmaee Moghaddam

Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Keywords:
Industrial internet of things
Low-power and lossy networks
Cross-layer
Routing protocol for low-power and lossy networks
Time Slotted Channel Hopping
Quality of Service

ABSTRACT

The Internet of Things (IoT) facilitates mutual interaction by establishing a connection between humans and objects. The adoption of IoT is rapidly expanding across various aspects of human life. Industrial IoT (IIoT) is the intersection point between IoT and industry, serving as a key driver of the Fourth Industrial Revolution, also referred to as Industry 4.0 (I4.0). Low-Power and Lossy Networks (LLNs), which contain many resource-constrained nodes, as the basic block in IoT and IIoT, are an essential part of this revolution. Given the resource constraint of LLNs and the mission, safety, and business criticality of IIoT, enhancing Quality of Service (QoS) has become a significant challenge. Addressing this challenge requires the optimal utilization of resources and fully coordinated decision-making across different layer protocols, which can be achieved through cross-layer design. Routing Protocol for Low-Power and Lossy Networks (RPL) and Time Slotted Channel Hopping (TSCH) are two widely used standard protocols in the IIoT protocol stack. In this paper, we investigate the fundamental factors affecting QoS within the RPL and TSCH and propose an RPL-TSCH Cross-Layer (RTCL) design for LLNs in IIoT, motivated by equations derived from queuing theory. The equations and simulation results reveal that the inconsistency between the arrival rate and service rate of the node's queue leads to increasing *queue overflow* and delay and decreasing reliability and throughput. The proposed RTCL design, structured as a six-step solution that integrates RPL and TSCH information, enhances QoS by employing two primary strategies: *controlling the arrival rate* and *increasing the service rate* of the nodes' queues. Evaluation results from various simulation scenarios demonstrate that RTCL improves QoS parameters while also enhancing load balancing and network stability.

1. Introduction

The Internet of Things (IoT) is a network consisting of many sensors, devices, and agents, collectively called "things," and accessible through the internet. The primary goal of IoT is to gather and share environmental information via the things around us. By facilitating a things-oriented approach to understanding and fulfilling our needs, IoT enables communication between us and the things to realize the slogan, "a better world for human beings." [1]. Being things-oriented means establishing a connection between things and the internet through protocols independently and without human intervention [2]. The development of the IoT has significantly advanced in the domains of buildings, business, health, and industry, leading to the creation of various applications such as smart homes, smart cities, smart health, and smart factories in recent years [3]. IoT can be divided into three main categories: Consumer IoT, Commercial IoT, and Industrial IoT (IIoT). Consumer IoT encompasses applications intended for general users in home environments, such as connected home appliances, smartphones, watches, and entertainment systems. Commercial IoT applications extend beyond home environments, including smart health systems, commercial office buildings, hotels, stores, etc. IIoT focuses on efficiently managing devices and industrial assessments through cost-effective solutions [4]. The goal of IIoT is to enhance industrial operations through the interconnection of devices at all times and locations within the manufacturing process, thereby improving efficiency, productivity, and reliability. IIoT differs from IoT in several aspects, including the types of devices, the development environment, Quality of Service (QoS) requirements, and the network scale [3]. IIoT often has a more extensive network size than IoT and requires stricter QoS due to mission, safety, and business criticality environment [5–7].

Low-power and Lossy Networks (LLNs) are a specific type of Low-power Wireless Personal Area Network (LoWPAN) consisting of sensors, agents, and routers that operate under energy, memory, processing, and communication constraints. LLNs, as a fundamental block of IoT and

* Crresponding author.
E-mail addresses: mehdi.zirak@mail.um.ac.ir (M. Zirak), y_sedaghat@um.ac.ir (Y. Sedaghat), yaghmaee@ieee.org (M.H.Y. Moghaddam).

https://doi.org/10.1016/j.adhoc.2025.103843
Received 21 October 2024; Received in revised form 16 February 2025; Accepted 25 March 2025
Available online 31 March 2025

IIoT, facilitate the collection of environmental data and device information through multi-hop communication over wireless links. For further processing, the collected data are sent to servers, control centers, or the cloud through an LLN Border Router (LBR), which is the root of LLN and does not have the limitations of nodes [8]. LLNs play a vital role in IIoT automation applications, which are categorized into process control and discrete manufacturing. Process control applications are used in fluid products, including those produced in refineries and petrochemical plants, such as greases, oil, gas, and other similar substances. In contrast, discrete manufacturing applications are typically employed in factories that produce individual goods such as televisions, screws, and tires. The primary objective of both IIoT application types is to collect the factory environment data that was either previously collected manually or not collected at all [9].

With the expanded use of LLNs, an adaptive layer for enabling IPv6 packet transmission over IEEE 802.15.4 was required to integrate LLNs into IP networks. Consequently, in 2004, the Internet Engineering Task Force (IETF) established a working group named IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) to address this challenge. The efforts of this group led to the creation of a 6LoWPAN adaptation layer between the network and data link layers in the IP stack. In 2008, the IETF formed a new working group, Routing over Low-power and Lossy Networks (RoLL), to develop appropriate routing solutions for LLNs. This group began researching existing routing protocols to meet LLN requirements. Additionally, RoLL published four Request For Comments (RFCs) to clarify the LLN requirements in various environments: industrial, urban, home, and building. The results of RoLL's research indicated that existing routing protocols were unsuitable for LLN requirements. Consequently, RoLL introduced the Routing Protocol for Low-power and Lossy Networks (RPL) as the standard routing protocol for LLNs in 2012 [10].

The Medium Access Control (MAC) layer controls communication scheduling between nodes. Using the IEEE 802.15.4 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as a standard MAC protocol cannot fulfill the strict requirements of IIoT. As discussed in [11], Time Division Multiple Access (TDMA) protocols can more efficiently address the sensitivity of industrial applications to reliability, delay, and predictability. Strict QoS parameters and the interfering industrial environment prompted the IEEE to release the Time Slotted Channel Hopping (TSCH) MAC protocol [12]. The IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) working group, under the supervision of IETF, standardized TSCH in IEEE 802.15.4e [8].

The MAC and network layers are two principal layers in the IIoT protocol stack that significantly affect QoS parameters in LLNs [13]. However, choosing protocols for these layers is not the sole influencing factor; their interaction also plays a critical role [14]. Although layer separation schemes provide modularity, they can lead to isolated decision-making between layers due to factors such as information separation. For example, creating routes using the RPL protocol without considering MAC layer scheduling can degrade QoS parameters. Addressing these issues has been the primary motivation for a shift from traditional separate-layer designs to cross-layer approaches. Cross-layer designs improve network performance by enabling decision-making based on shared information and multi-layer collaboration [11,15–17].

This paper proposes a cross-layer design of the RPL and TSCH protocols to enhance QoS parameters in LLNs for IIoT applications. In the proposed RPL-TSCH Cross-Layer (RTCL) design, we utilize consolidated information from both RPL and TSCH to make informed decisions in managing these protocols. In summary, the main contributions of this paper are as follows:

1. Investigating the relationship between the components of the RPL and TSCH protocols, focusing on their mutual interactions and effects.
2. Identifying the fundamental factors affecting QoS parameters and analyzing these factors using simulation and queuing theory-based equations.
3. Developing a cross-layer design for RPL and TSCH to manage fundamental factors and improve QoS parameters.
4. Performance evaluation of RTCL versus the separation-layer and cross-layer methods, using various evaluation criteria and scenarios through simulation.

The rest of this paper is organized as follows: Sections 2 and 3 review background and related works, respectively. Section 4 presents the motivation behind the proposed method. Section 5 describes the node model in an RPL-TSCH network, along with its corresponding queue equations, and details the proposed RTCL design. Section 6 presents the simulations conducted to evaluate the performance of RTCL design and discusses the results. Finally, Section 7 concludes the paper and outlines future work.

2. Background

2.1. RPL

RPL is a tree-based routing protocol that organizes the network into one or more Destination-Oriented Directed Acyclic Graphs (DODAGs). It supports three traffic models: Point-to-Point (P2P), Point-to-Multipoint (P2MP), and Multipoint-to-Point (MP2P) [18]. The LBR, which serves as the DODAG's destination, initiates DODAG creation and gathers data. RPL organizes DODAGs by assigning node ranks using an Objective Function (OF). OF zero (OF0) and Minimum Rank with Hysteresis OF (MRHOF) are two standard OFs supplied by RPL. OF0 calculates a node's rank based on hop count to create the shortest path between nodes and the LBR, while MRHOF uses the Expected Transmission Count (ETX) metric, which prioritizes link quality. ETX for a link represents the number of expected transmissions needed to deliver a packet and its acknowledgment successfully. Nodes link as parent and child based on rank, with each child maintaining a Candidate Parent List (CPL). However, each child selects one parent from the CPL as its preferred parent for time synchronization and data forwarding [19–21]. The left part of Fig. 1 shows an LLN organized according to RPL rules. RPL introduces four ICMPv6 control messages to create and manage changes within the DODAG. These messages include: (1) DODAG Information Object (DIO), which is initially broadcast by the LBR and periodically repeated by all joined nodes using a trickle timer, used to create and maintain the DODAG, CPL, and carry information such as rank and objective function; (2) Destination Advertisement Object (DAO), which establishes P2P and P2MP communication patterns; (3) Destination Advertisement Object Acknowledgement (DAOACK), which is unicast by the DAO recipient to the sender; and (4) DODAG Information Solicitation (DIS), which is broadcast by a node to solicit DAO packets from its neighbors [8].

2.2. TSCH

TSCH is a TDMA-based MAC protocol that improves energy efficiency, reliability, and delay predictability while also using channel hopping to mitigate multi-path fading effects [8]. As shown on the right side of Fig. 1, which indicates a sample slotframe corresponding to the routing scheme proposed by RPL on the left side, TSCH has two essential components: the queue and the slotframe. The TSCH queue is a limited space for storing received packets before forwarding. TSCH divides time into slotframes, which repeat periodically. The slotframe can be represented as a two-dimensional scheduling matrix. Rows of the slotframe represent channel offsets, and columns represent time offsets. Each slotframe cell indicates a state for the MAC layer: send, receive, or idle. Slotframe contains a fixed number of timeslots, known as the SlotFrame Length (SFL). Timeslot length is typically between 10 and 15 ms, sufficient to transmit a packet and receive the corresponding

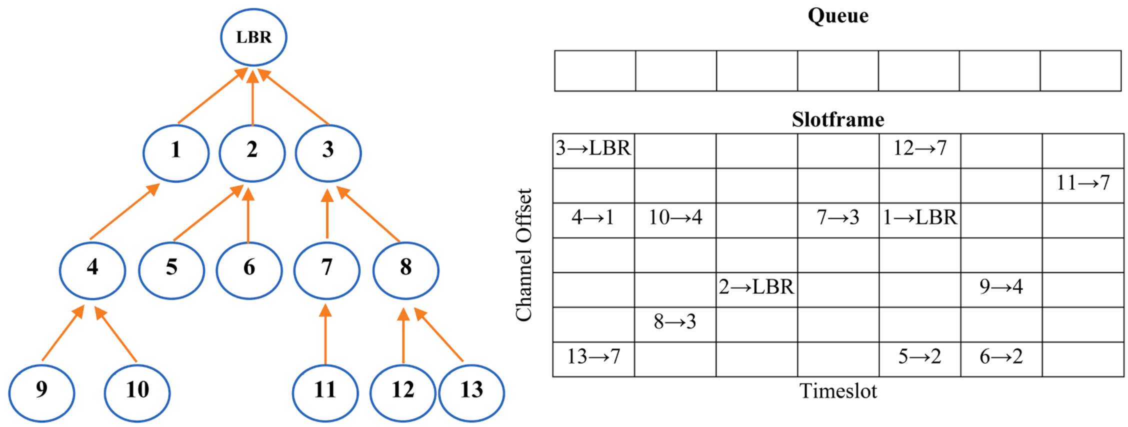


Fig. 1. RPL DODAG and TSCH queue and slotframe.

acknowledgment [22]. In each timeslot, nodes can send packets simultaneously but on different channels using the channel hopping mechanism. The Absolute Slot Number (ASN) acts as the global clock in a TSCH network, incrementing by one at the end of each timeslot. The coordinator node (usually the LBR node) initializes the network by setting the ASN to zero. ASN synchronization is critical, as nodes calculate the current timeslot and channel based on the ASN. The coordinator and the joined nodes broadcast Enhanced Beacon (EB) messages periodically, for ASN synchronization and new nodes joining the network. The current timeslot and channel are calculated using Eqs. (1) and 2 [22].

$$current\ time\ slot = ASN \ \%\ \mathrm{SFL} \quad (1)$$

$$Current\ channel = CHL[\ (\ ASN + ChannelOffset\ )\ \%\ |CHL|] \quad (2)$$

In Eq. (1), the current timeslot is between 0 and SFL. Eq. (2) identifies the current channel as a physical channel using the Channel Hopping List (CHL). Due to the channel hopping mechanism in TSCH, channel offsets do not directly correspond to the IEEE 802.15.4 physical channels. The CHL lists the IEEE 802.15.4 physical channels to translate the channel offset to the physical channel.

3. Related work

The advantage of cross-layer over separation-layer approaches lies in their simultaneous use of information from multiple layers during decision-making. This section reviews the existing cross-layer approaches proposed on RPL and TSCH and analyzes the shared information used in these systems. To this end, in Table 1, we investigate the TSCH parameters, including queue length, link quality, slotframe state (i.e., scheduling between nodes), and queue state (i.e., queue packet arrangement), as well as RPL parameters, including node rank, DODAG structure (i.e., parent-child relationship), and EXT, that each related work utilizes in their proposed system designs. Table 1 presents the differences between RTCL and other reviewed cross-layer approaches on the RPL and TSCH protocols.

The study by Estepa et al. [23], showed that transmission power directly influences the neighbor set detection and RPL parent selection processes, thereby indirectly affecting preferred parent selection, especially when link-quality parameters like ETX are used to determine the OF. They have suggested a scheme to create a dependency between the preferred parent selection of RPL and transmit power adjustment. In the proposed cross-layer on RPL and IEEE 802.15.4, nodes select the preferred parent by identifying neighbors and collecting statistical data on each transmit power level.

In the radio duty cycling mechanism of the MAC layer, strobe packets synchronize the sender and receiver before data transmission. Safaei et al. [24] introduced the Strobe Per Packet Ratio (SPR) measure in a cross-layer scheme called ELITE, which combines RPL and MAC layer. ELITE reduces energy consumption by using the RPL objective function based on SPR. Each node calculates a different SPR for each neighbor according to the wake-up time and channel check interval. Then, RPL uses this information and routes on neighbors with the lowest SPR to minimize the number of strobe packets needed to transmit.

In [25], the authors examined the performance of 6TiSCH networks under heavy traffic and identified issues related to RPL's bandwidth allocation. They proposed a cross-layer, traffic-aware solution for RPL and TSCH in IIoT environments called Traffic Aware RPL (TA RPL) to address load balancing. In this approach, RPL determines node bandwidth based on TSCH slotframe scheduling. The TA RPL objective function calculates node ranks based on bandwidth, hop count, and ETX parameters. By considering the bandwidth requirements of each traffic flow, TA RPL establishes routes with sufficient bandwidth.

Table 1
Summary and comparison of related works.

Reference	TSCH				RPL		
	Queue length	Link quality	Slotframe state	Queue state	Node rank	DODAG structure	ETX
[23]		✔				✔	
[24]		✔				✔	
[25]			✔		✔		✔
[28]			✔		✔	✔	
[32]			✔			✔	
[35]	✔				✔		✔
[37]	✔					✔	
[38]	✔		✔			✔	✔
[39]	✔					✔	
[40]		✔			✔	✔	✔
[41]	✔		✔		✔	✔	
RTCL	✔	✔	✔	✔	✔	✔	✔

Vera-Pérez et al. [26] presented an analytical model to study the interactions between RPL and TSCH protocols and analyze their mutual effects. This model examines network behavior during deployment, initializes scheduling, and establishes RPL connections. The validation was through simulations using the Cooja simulator [27] and real-world testbeds. The results suggest that RPL and TSCH should collaborate to determine optimal network configurations, enhancing network lifetime and reducing energy consumption across various operational stages, from deployment to maintenance.

Dilution-based Convergecast Scheduling (DCS) [28], designed based on scheduling presented in [29,30] and dilution procedure [31], provides a convergecast and distributed scheduling algorithm for TSCH with infrastructure provided by RPL. The network is divided into grid boxes through a dilution process. In boxes, nodes exchange messages, form local trees, and select a leader. DCS obtains a convergecast tree of the entire network by calculating the rank of the leader nodes and connecting them by RPL. The links between the leader and the box members are called internal links, while the links between leaders of different boxes are termed external links. The leader in each box is responsible for scheduling the internal links, whereas the external links are scheduled based on the box number.

In [32], Jenschke et al. improved energy consumption in the Common Ancestor (CA) method [33] to create multipaths on RPL and proposed a new approach called On-Demand Selection (ODeSe). In ODeSe, packets are routed to the LBR through multiple paths to enhance reliability. Additionally, a two-hop control approach is employed to mitigate flooding in CA, wherein each node designates a preferred parent and an alternative parent for itself and the next node. The functions of Packet Automatic Repeat reQuest, Replication and Elimination, and Overhearing (PAREO) [34] are utilized to determine the preferred and alternative parents. PAREO functions leverage the information of the TSCH protocol for these calculations.

Congestion control under dynamic traffic loads in RPL is addressed in [35] through parent selection based on minimal congestion. The Queue Occupancy Factor (QOF) is a metric that evaluates the congestion level of each parent by measuring the number of packets in the TSCH queue. The Parent Score (PS) represents the desirability of each parent for being selected as the next hop and is determined based on ETX, rank, and QOF parameters. Each node selects a parent based on the PS to ensure packets forward through less congested paths.

Orchestra [36], a well-known scheduling algorithm for TSCH, does not support burst traffic. Diana Deac et al. [37] proposed a traffic-aware approach to enhance the Orchestra's performance to address this limitation. In the proposed method, which focuses on the resources required by the root's child nodes, TSCH queue length was the traffic load measure. Each child node determines the size of its subtree using DAO messages. When the number of packets in the TSCH queue exceeds a predefined threshold, the child node requests additional resources from the root.

The hysteresis-free on-the-fly (HF-OTF) [38] is a scheduling function designed to manage TSCH resources under dynamic traffic loads by addressing network traffic variations through over-provisioning. At the end of each time interval, the total queue occupancy, average cell utilization, and the number of child nodes are determined. If the slot utilization is below a fixed threshold of 70 %, it is considered under-utilized. The difference between the current and required bandwidth determines the number of time slots to be added or removed. The required bandwidth is estimated based on the average number of packets received from the upper layer in each slotframe and the ETX between the child node and its parent.

The paper [39] addresses unforeseen traffic in an RPL and TSCH-based network. In the proposed method, the node cell requirement is disseminated throughout the network using the RPL control message option embedded in DIO and DAO messages. Each node extracts information from DIO messages, which travel from the LBR towards leaf nodes, to learn about the upstream nodes' requirements, and from DAO messages, which travel from leaf nodes to the LBR, to understand the downstream nodes' requirements. The required number of cells for each node is determined based on the length of its TSCH queue.

The reliable and delay-efficient multi-path RPL (RDMP-RPL) [40] is an RPL-based routing algorithm in which the preferred parent is selected based on mobility, hop count to the root, and ETX parameters. If a child node cannot meet the required reliability through its preferred parent, it adopts a multi-parent strategy. Each child node selects k parents based on reliability and end-to-end delay parameters and enhances reliability by duplicating data packets for these k parents. The end-to-end delay is divided by the hop count to the sink to support delay constraints, providing a permissible link delay metric. Consequently, only parents with links ensuring an acceptable delay are selected.

Wijayasekara et al. [41] propose a centralized approach leveraging parameters such as the network topology formed by RPL and the queue lengths to develop a scheduling algorithm based on Integer Linear Programming (ILP). The proposed approach, inspired by the Traffic-Aware Scheduling Algorithm (TASA), employs three ILP techniques: ILP1, which uses a preselected parent; ILP2, which applies joint optimization for parent selection; and ILP3, which focuses on fast parent selection for parent assignment, packet scheduling, and frequency selection. By emphasizing the queue lengths of the root's child nodes, the authors aim to distribute the load among them effectively, ultimately achieving their primary objective of minimizing the average delay.

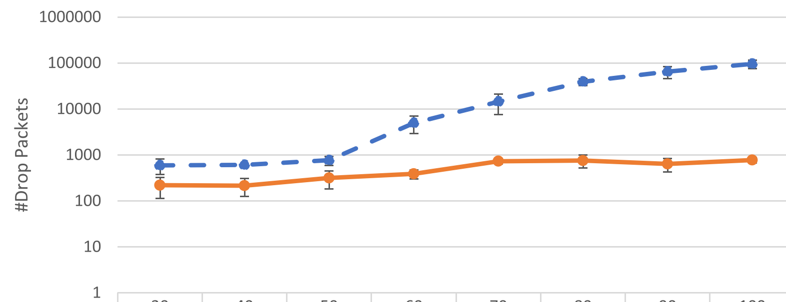
4. Motivation

The RPL protocol, as a standard protocol, is compatible with various MAC and PHY protocols. Among these, TSCH is considered a suitable option for cooperation with RPL in critical environments, including IIoT. TSCH provides substantial information on neighboring nodes, scheduling, and queue status, all of which can significantly impact the performance of RPL routing [42–44]. Creating a routing topology with RPL without considering the topology established by TSCH can negatively affect network QoS parameters.

QoS parameters in an LLN are interdependent, meaning that changes in one parameter can directly influence others due to common underlying factors impacting multiple QoS metrics simultaneously. One of the primary factors is packet loss. When packet loss occurs, it reduces network reliability. To address this challenge, lost packets need to be retransmitted, either end-to-end or hop-by-hop, which increases latency, consumes additional energy, and uses up network bandwidth. Consequently, this reduces the network's lifetime and throughput. Therefore, reducing reliability in an LLN is like toppling the first domino, negatively affecting other QoS parameters such as latency, throughput, and network lifetime. In critical IIoT environments, where applications are typically loss-sensitive, such losses can have catastrophic consequences for industrial operations [45].

To address this issue, increasing reliability, throughput, and network lifetime while reducing latency requires careful attention to packet loss. We conducted experiments using the 6TiSCH simulator [46] to measure packet loss in a standard RPL-TSCH network and layer separation design. In these simulations, nodes with a TSCH buffer size of 10 packets transmitted collected data to the LBR at a rate of one packet per slotframe. The traffic pattern used was MP2P, typical for data collection in IIoT networks. Packet loss can be categorized into two main types: *queue overflow* and *maximum retry*. *Queue overflow* occurs when the TSCH protocol at the receiving node lacks sufficient buffer space, leading to packet drops. *Maximum retries* occur when the MAC layer exhausts its attempts to transmit a packet without success, often due to a lack of a scheduled cell or link failure caused by interference or bit errors.

For results confidence, we simulated the network for 30 runs, with



	30	40	50	60	70	80	90	100
Queue Overflow	591.5	606.25	765.75	4935.75	14428.5	39061.75	64793	95568.25
Max Retries	219.75	214.75	315.25	387.75	728.75	753.75	637.5	774.25

Network size (Nodes)

Fig. 2. Number of lost packets due to *queue overflow* and max *retries* for different RPL-TSCH network sizes.

each simulation running for 90 min. Fig. 2 plots the average number of lost packets (on a log scale) against the network size and demonstrates how packet loss rates escalate drastically in an RPL-TSCH network as the network size increases.[1] For instance, with 50 nodes, the network loses approximately 1081 packets; with 100 nodes, this number increases to around 96,313 packets. This indicates that a 2-fold increase of nodes results in an 88-fold increase in lost packets, highlighting the exponential growth of packet loss with network size. This point is highlighted when we realize that IIoT networks are often large sizes [5–7,47]. The figure also shows that most lost packets (between 71 % and 99 %) are due to *queue overflow*. As network size increases, so does the likelihood of packet loss caused by *queue overflow*. For networks with fewer than 50 nodes, approximately 73 % of packet loss is due to *queue overflow*, whereas for networks with over 80 nodes, this figure rises to >98 %. This behavior can be attributed to the significant increase in the number of transmitted packets as the network grows while buffer sizes remain fixed, resulting in a higher probability of *queue overflow*. In the next section, we will explore the causes of *queue overflow* through equations derived from queuing theory, and based on these equations, we will present our proposed solution.

5. Proposed method

5.1. Node model

In the RPL-TSCH network, RPL organizes nodes into a DODAG structure, and the nodes send packets through a schedule created by TSCH in an interference-free manner. These packets include data generated by the nodes, forwarded data from child nodes, and control packets from different layers. Fig. 3 illustrates the node model in an RPL-TSCH network with 13 nodes and an LBR. On the right side of Fig. 3, the internal mechanism of a node is depicted. Upon receiving a packet, RPL determines the next node based on the DODAG structure and the transmission type (unicast or broadcast) before forwarding it to TSCH. TSCH stores all packets in its buffer, a queue of limited size, before sending them. When a timeslot is scheduled for a neighbor, TSCH searches the queue for the first packet destined for that neighbor and sends a copy of the packet. As illustrated in Fig. 2, TSCH *queue overflow* emerges as a critical factor contributing to packet loss. In the following section, we will examine the performance of the TSCH queue using queuing theory equations.

5.2. TSCH queue model

The queuing model is specified using arrival rate, service rate, number of servers, and system capacity. In the RPL-TSCH network, the arrival rate of the node's TSCH queue is influenced by several factors, including the number of packets generated by the node, the number of child nodes, the transmission rate of these child nodes, and the quality of the link between the parent and child nodes. Therefore, we consider the arrival rate a random variable with a Poisson distribution, denoted by parameter λ. The service rate is equivalent to the number of packets successfully sent to the parent node and received acknowledgment or removed from the queue due to multiple unsuccessful transmission attempts by TSCH, which reach the MAC layer's *maximum retries* limit. Due to factors such as the lossy nature of wireless links in LLNs, the dynamic allocation of timeslots in TSCH, and frequent changes in network topology by RPL, the service rate cannot be considered constant. Hence, we consider the service rate a random variable with a Poisson distribution, denoted by parameter μ. The system capacity represents the maximum number of packets that can be held simultaneously. In a system with a limited queue size, the system capacity equals the sum of the queue size and the number of servers. In short, the TSCH queue model can be described as an M/M/c/K system, where M, c, and K represent the Poisson distribution, the number of servers, and the system capacity, respectively. Let N(t) denote the number of packets in the system at time t, and π_n represent the probability of having n packets in the system, then [48]:

$$\pi_n = \underset{t\to\infty}{limp}\{ N(t) = n\} \tag{3}$$

In an M/M/c/K queue model with arrival rate λ and service rate μ, the birth-death process is represented using a continuous-time Markov chain, where each state represents the number of packets in the system. Fig. 4 illustrates the birth-death process for the queue model. The notations used are summarized in Table 2.

As depicted in Fig. 4, the arrival and service rates are not fixed in this model and are defined based on system capacity and the number of servers. The arrival rate remains λ as long as the system has available capacity (i.e., $n < K$). However, when the system is full (i.e., $n \geq K$), the arrival rate becomes 0. Similarly, the service rate is affected by the number of servers. Suppose the number of packets in the system is less than that of servers (i.e., $n < c$); the service rate equals nμ. If the number of packets in the system exceeds the number of servers (i.e., $n \geq c$), the service rate equals cμ. Eqs. (4) and 5 express the arrival and service rates based on the number of packets in the system [49].

$$\lambda_n = \begin{cases} \lambda \; 0 \leq n < K \\ 0 \; K \leq n \end{cases} \tag{4}$$

[1] In this chart, all packets, including data packets, RPL ICMP (DIO, DIS, DAO, DAOACK), TSCH (EB, Keep-Alive), and IP(6P), are considered.

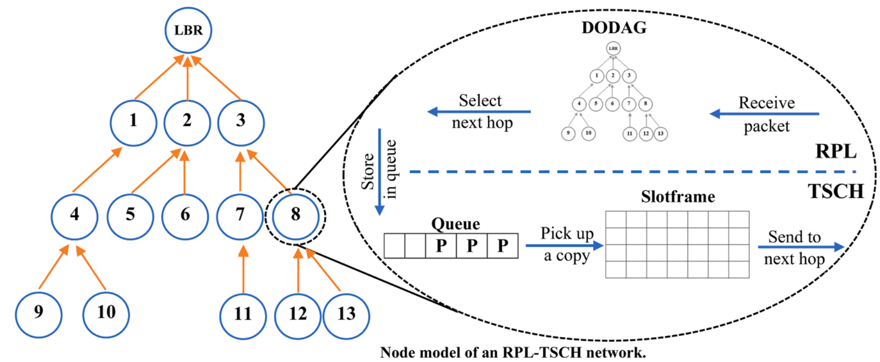


Fig. 3. Node model of an RPL-TSCH network.

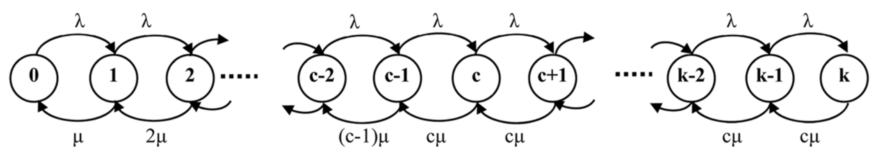


Fig. 4. Birth-death process for M/M/c/k queue [50].

Table 2
List of notations.

Notation	Description	Notation	Description
λ	Queue arrival rate	μ	**Queue service rate**
n	Number of packets in the system	π_n	**Probability of existing n packets in the system**
c	Number of servers	K	**System capacity**
QL_i	Queue length of node i	QL^T	**Queue length threshold**
$ETX_{i,j}$	ETX of link between i and j	ETX^T	**ETX threshold**

$$\mu_n = \begin{cases} n\mu \ 0 \le n < c \\ c\mu \ c \le n \le K \end{cases} \tag{5}$$

The steady-state equations of an M/M/c/K queuing model led to a stochastic balance concept, indicating that the input rate of a state equals the output rate of that state [51]:

$$\lambda\pi_0 = \mu\pi_1$$

$$(\lambda+\mu)\pi_1 = 2\mu\pi_2 + \lambda\pi_0$$

$$(\lambda+2\mu)\pi_2 = 3\mu\pi_3 + \lambda\pi_1$$

$$\dots$$

$$(\lambda+(c-2)\mu)\pi_{c-2} = (c-1)\mu\pi_{c-1} + \lambda\pi_{c-3}$$

$$(\lambda+(c-1)\mu)\pi_{c-1} = c\mu\pi_c + \lambda\pi_{c-2}$$

$$(\lambda+c\mu)\pi_c = c\mu\pi_{c+1} + \lambda\pi_{c-1}$$

$$(\lambda+c\mu)\pi_{c+1} = c\mu\pi_{c+2} + \lambda\pi_c$$

$$\dots$$

$$(\lambda+c\mu)\pi_{K-2} = c\mu\pi_{k-1} + \lambda\pi_{k-3}$$

$$(\lambda+c\mu)\pi_{K-1} = c\mu\pi_K + \lambda\pi_{K-2}$$

$$c\mu\pi_K = \lambda\pi_{K-1}$$

π_1 can be calculated based on π_0 using the first equation of this set. By substituting π_1 into the second equation, π_2 can also be calculated based on π_0:

$$\pi_1 = \frac{\lambda}{\mu}\pi_0$$

$$\pi_2 = \left(\frac{\lambda}{\mu}\right)^2\left(\frac{1}{2}\right)\pi_0$$

π_n can be expressed in terms of π_0 by continuing this process. Defining $\rho=\lambda/\mu$ [49]:

$$\pi_n = \begin{cases} \dfrac{\rho^n}{n!}\ \pi_0 \ 0 \le n < c \\ \dfrac{\rho^n}{c!c^{n-c}}\pi_0 \ c \le n \le K \end{cases} \tag{6}$$

There is one server in the TSCH queue model, so the M/M/c/K model can be simplified to an M/M/1/K model by setting $c = 1$. Hence, for π_n:

$$\pi_n = \rho^n \ \pi_0 \ 0 \le n \le K \tag{7}$$

Given that $\sum_{n=0}^{K} \pi_n = 1$ and replace π_n from Eq. (7), for $\neq\rho1$:

$$\sum_{n=0}^{K} \pi_n = 1 \Rightarrow \pi_0 + \sum_{n=1}^{K}\rho^n \ \pi_0 = 1 \Rightarrow \pi_0\left[1+\sum_{n=1}^{K}\rho^n\right] = 1$$

$$\Rightarrow\pi_0 = \left[1+\sum_{n=1}^{K}\rho^n\right]^{-1} \Rightarrow \pi_0 = \left[1+\frac{\rho(1-\rho^K)}{1-\rho}\right]^{-1} \Rightarrow\pi_0 = \frac{1-\rho}{1-\rho^{K+1}} \tag{8}$$

And for ρ=1:

$$\sum_{n=0}^{K} \pi_n = 1 \Rightarrow \sum_{n=0}^{K}\rho^n \ \pi_0 = 1 \Rightarrow (K+1)\pi_0 = 1 \Rightarrow\pi_0 = \frac{1}{K+1} \tag{9}$$

Queue overflow occurs when the system's capacity is full, meaning the system is in state K. The probability of *queue overflow* is equal to π_K, which can be calculated using equations 7, 8, and 9:

$$P(\text{QueueOverflow}) = \pi_K = \rho^K \ \pi_0 = \begin{cases} \dfrac{\rho^K(1-\rho)}{1-\rho^{K+1}} \ \rho \neq 1 \\ \dfrac{1}{K+1} \ \rho = 1 \end{cases} \tag{10}$$

In Eq. (10), the value of π_K is influenced only by the parameters K and ρ. Therefore, in Fig. 5, we have plotted π_K for different values of ρ and K.

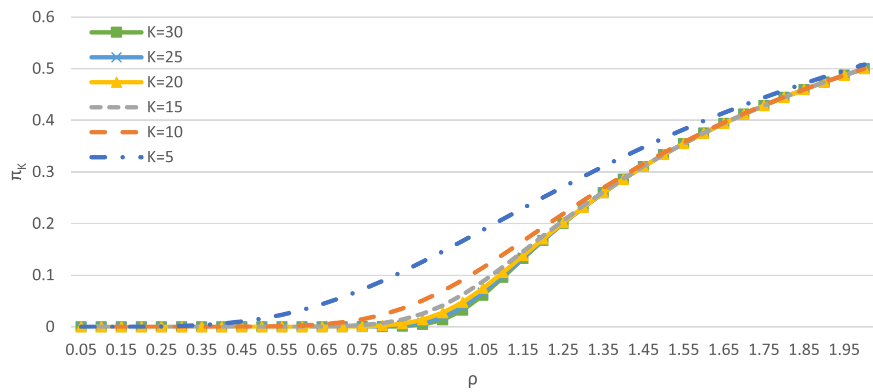


Fig. 5. π_K versus different values of ρ and K.

Fig. 5 demonstrates that to reduce the probability of *queue overflow* in TSCH, we should focus on ρ. As ρ decreases, the probability of *queue overflow* approaches zero. However, as ρ increases and approaches 1, the *queue overflow* probability rises rapidly. The probability of *queue overflow* increases more rapidly for smaller than larger queue sizes. Nevertheless, for ρ≥1, the *queue overflow* probability increases quickly across systems with different queue sizes. In such cases, increasing the queue size cannot compensate for the increase in ρ and prevent *queue overflow*, effectively neutralizing the effect of increasing K.

In summary, increasing the queue size can reduce the probability of *queue overflow* as long as ρ<1. However, if ρ≥1, even a TSCH with a larger queue size cannot prevent *queue overflow*. Therefore, the best approach to prevent *queue overflow* is to control ρ. To control ρ, according to the equation $\rho=\lambda/\mu$, we must *control the arrival rate* while also *increasing the service rate* at the nodes. This issue is particularly critical for nodes where the Queue Length (QL) is growing. It should be noted that TSCH does not have any flow control mechanism [22]. Therefore, in the proposed method introduced in the next section, we implement two strategies to control ρ in the TSCH queue: *controlling the arrival rate* and *increasing the service rate*. The first strategy goal is to adjust the arrival rate to prevent *queue overflow*, while the second strategy aims to *increase the service rate* to reduce the TSCH queue length. In the following sections, we describe the proposed method's steps and each step's performance in achieving these two goals and improving QoS parameters.

5.3. RPL-TSCH cross-layer design

The proposed cross-layer design aims to leverage RPL and TSCH information to improve QoS parameters for IIoT networks in a distributed manner. RPL provides information such as ETX, preferred parent, and CPL, while TSCH offers information such as link quality, queue status and transmission schedule. By combining this information in a cross-layer approach, the proposed design in a distributed manner seeks to improve reliability, throughput, and delay, which are among the most critical QoS parameters in the IIoT environment [3,5,6,52–54]. Fig. 6 presents a general overview of the proposed design, which includes six steps.

As mentioned in Section 1, each non-root node in RPL has a CPL but typically uses only the preferred parent to forward packets. However, in the multi-parent approach, the proposed method suggests using a Refined Candidate Parent List (RCPL) instead of relying solely on a single parent. In the first step, RTCL receives the CPL from RPL and generates an initial RCPL. In the RCPL, the preferred and candidate parents are treated equally, with no distinction. The proposed multi-parent approach distributes the child node's output traffic among RCPL by considering at least one timeslot per slotframe for each parent in the RCPL. Consequently, the number of packets transmitted per slotframe boosts, which *increases the service rate*, ensures load balancing and enhances overall network throughput. Additionally, the *control of the arrival rate* is achieved by distributing the arrival rate across multiple parents. This step also reduces the probability of *queue overflow* and queue waiting times by decreasing the average TSCH queue length, leading to increased reliability.

To better illustrate the steps of the RTCL design, we use the example shown in Fig. 7. This figure depicts a section of the DODAG network formed by RPL. Directional arrows indicate the links between child nodes and their CPL. Solid arrows represent the links between child nodes and their RCPL, while dashed arrows indicate links with candidate

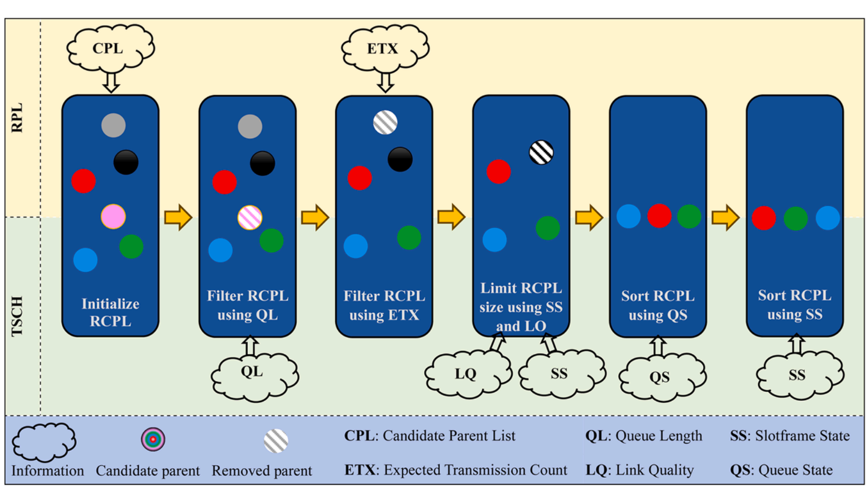


Fig. 6. Steps of the proposed method.

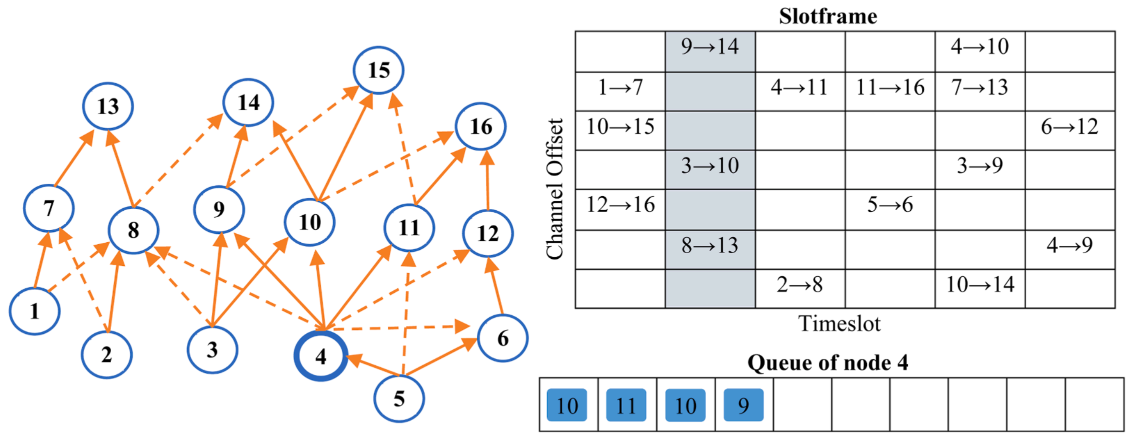


Fig. 7. A sample of the RPL-TSCH network with CPL, RCPL, slotframe, and queue.

parents not selected as part of the RCPL. The right side of the figure displays the slotframe created by TSCH, highlighting the current timeslot. Additionally, as an example of a TSCH queue, a snapshot of the TSCH queue of node 4 is shown, where the next hop is inserted into each packet.

Let us follow the RTCL process for node 4, which has the largest CPL and whose TSCH queue status is also shown in Fig. 7. In the first step of Fig. 6, by receiving the CPL from RPL, the initial RCPL is formed, consisting of {6, 8, 9, 10, 11, 12}. The second step requires the TSCH QL of the candidate's parents, which is unavailable by default in the RPL. However, the periodic DIO messages can piggyback QL, ensuring updated information is consistently available for child nodes. QL increases for parents who cannot maintain their service rate above the arrival rate. Therefore, in the second step, we apply a QL Threshold (QL^T), filter out parents whose QL exceeds QL^T, and remove them from the RCPL. Since *queue overflow* is the primary cause of packet loss, this step helps *control the arrival rate* and reduces *queue overflow* by avoiding sending packets to parents with high QL. In the example of Fig. 7, suppose $QL_8 \geq QL^T$; consequently, the RCPL becomes {6, 9, 10, 11, 12}.

In the third step, the ETX of RPL is used with a filter to remove parents with ETX links below a specified threshold (ETX^T) to *increase the service rate*. By sending packets to parents with higher link quality, the likelihood of packet loss due to link loss is reduced, thereby *increasing the service rate* and throughput. Assuming that in Fig. 7, $ETX_{4,6} < ETX^T$, node 6 is removed from the RCPL, resulting in {9, 10, 11, 12}.

In the fourth step, we use TSCH protocol information to limit the size of the RCPL. Our experiments showed that excessive growth of the RCPL leads to slotframe congestion, increasing the probability of collisions and packet loss, violating the objective of *increasing the service rate*. Therefore, the RTCL seeks a balance between RCPL size and collision probability in slotframe, limiting the RCPL size based on slotframe crowding. Additionally, using MAC performance metrics embedded in IEEE 802.15.4e [12], TSCH calculates link quality for each parent and removes those with the lowest link quality from the RCPL. Suppose the TSCH protocol of node 4 limits the RCPL size to three, and the link between nodes 4 and 12 has the lowest quality; hence, the RCPL is reduced to {9, 10, 11}. At this point, TSCH updates the slotframe and allocates a cell in the slotframe for each parent in the RCPL.

In the second step of RTCL, we use the queue length of parent nodes, but in the fifth step, we use the TSCH queue status of the node itself. In the TSCH queue, packets await transmission to the parent nodes, forming sub-queues for each parent in the RCPL. If packet distribution among parents is unbalanced, the benefits of multi-parenting are lost, and most packets are sent to a single parent. Consequently, the number of packets sent per slotframe decreases, violating the goal of *increasing the service rate*. For example, in Fig. 7, ifa new packet is scheduled for parent 10, and there are already two packets queued for this parent, the packet must wait for 2 slotframes and 3 timeslots (2 slotframes for sending previous packets and 3 timeslots until the first available timeslot of parent 10). However, if the new packet is scheduled for parent 9, it will wait only for 1 slotframe and 4 timeslots. Therefore, in step 5, using the TSCH queue state of the node and a sorting operation, the RCPL is arranged in ascending order based on the number of packets in the queue, resulting in {9, 11, 10}. This step reduces packet delay by prioritizing parents with shorter sub-queues.

However, an additional sorting step is required if two or more parents have the same sub-queue length. In step 6, parents with identical sub-queue lengths are sorted based on the distance between the current timeslot and next timeslot of each parent. For example, in Fig. 7, the distance between the current timeslot and the next timeslot of parents 9 and 11 is 4 and 1 timeslot, respectively. Therefore, parent 11 can transmit the new packet sooner. Finally, the RCPL becomes {11, 9, 10}, and parent 11 is the best choice to send the new packet in the current timeslot. Steps 5 and 6 aim to expedite packet transmission, reduce delay and queue length, and ultimately *increase the service rate*.

Steps 5 and 6 of the proposed method operate online in each time slot to sort the RCPL parents. However, Steps 1 to 4, responsible for constructing the RCPL, employ a periodic and event-driven approach to adapt to the dynamic conditions of IIoT networks, such as topology changes, node failure, and varying traffic. Specifically, at the end of each slot frame, the RCPL is updated periodically based on the piggybacked information from DIO messages exchanged during the slot frame. Additionally, any event such as a rank change, LQ variation, ETX dropping below the ETX^T threshold, or QL exceeding the QL^T threshold for any RCPL parent triggers the re-execution of Steps 1 to 4 to update the RCPL. If no CPL members qualify for inclusion in the RCPL, the proposed system activates a fallback mechanism, initializing the RCPL with the preferred parent.

RTCL imposes minimal overhead on the system, as the RPL and TSCH protocols already provide all the necessary information for RTCL execution at each node, including CPL, ETX, LQ, SS, and QS. The only additional information needed is the parents' QL, which piggybacks on DIO messages using the reserved 1-byte field specified in RFC 6550, avoiding any bandwidth and energy overhead on the network. Furthermore, from the computational perspective, if n=|CPL|, Steps 2 to 4 of RTCL have a complexity of $O(n)$, while Steps 5 and 6 have a complexity of $O(nlogn)$. As a result, the overall computational overhead of RTCL is $O(nlogn)$.

6. Evaluation

In this section, we evaluate the efficiency and effectiveness of the proposed method on QoS parameters through simulations using the 6TiSCH simulator [46]. The 6TiSCH simulator is an open-source tool that implements a fully standard 6TiSCH protocol stack, enabling researchers and developers to evaluate their optimizations. The simulator, developed in Python, simulates the behavior of 6TiSCH network protocols, including network formation, RPL routing, and TSCH scheduling. The rest of this section is structured as follows: We review the evaluation setup and criteria, followed by a presentation of the results.

6.1. Evaluation setup and criteria

Table 3 summarizes the RPL, TSCH, RTCL, and general settings used in the evaluations. Each simulation was run for 5400 timeslots, equivalent to 90 min, and repeated 30 times to increase the confidence level of the results. The packet sending rate was set to one packet per slotframe for each node. We evaluated two scenarios: the first focused on network size, with the number of nodes ranging from 50 to 100, and the second focused on TSCH queue size, varying between 5 and 20 packets. In the RPL protocol, the OF0 determined the DODAG structure and candidate parents and the trickle timer was set to 20 slotframes. The Minimum Scheduling Function (MSF) [55] is the standard scheduling function, and the MAC *maximum retries* equal to 5 used by TSCH. Each slotframe contains 100 timeslots with a length of 10 ms, and sixteen different channels are considered for the TSCH slotframe, resulting in a slotframe dimension of 16×100.

The optimal performance of RTCL depends on selecting appropriate values for the ETX^T and QL^T parameters. A low ETX^T value increases the number of parents in the RCPL, enabling the benefits of multi-parenting. However, the link quality between children and RCPL parents is reduced, negatively affecting QoS. Conversely, a high ETX^T value ensures better link quality for RCPL parents but reduces their number, limiting the advantages of multi-parenting. A similar trade-off applies to QL^T. A low QL^T value results in fewer selected parents, leading to lower queue congestion and reduced packet drop probability. On the other hand, a high QL^T value increases the number of RTCL parents, but these parents have longer queue lengths, raising the likelihood of packet drops. To analyze RTCL's sensitivity to ETX^T and QL^T and determine their optimal values, we conducted experiments based on the network's Packet Delivery Ratio (PDR) using the simulation settings in Table 3, a network of 100 nodes, a queue size of 10, and various ETX^T and QL^T values. The results in Table 4 indicate that RTCL achieves optimal performance with ETX^T and QL^T values of 0.8 and 70 %, respectively.

The evaluation criteria are as follows:

1. **PDR:** The ratio between the number of packets generated by the nodes and the number of packets received by the LBR, reflecting the network's reliability in packet delivery [56]. This metric is critical for IIoT applications due to their critical nature. As there is a direct relationship between PDR and throughput, we focus solely on PDR in our evaluations and pass-over throughput.

Table 3
Simulation settings.

General	Simulation time	5400 slotframes
	Number of runs	30
	Packet rate	1 data packet per slotframe
	Number of scenarios	2
	Network size	50–100 nodes
RPL	Objective function	OF0
	Trickle time period	20 slotframes
TSCH	Scheduling function	MSF
	MAC maximum retries	5 times
	Timeslot duration	10 ms
	Slotframe length	100 timeslots
	Number of channels offset	16
	Queue size	5–20 packets
RTCL	QL^T	70 % of queue size
	ETX^T	0.8

2. **Network lifetime:** Network lifetime is a fundamental criterion that impacts QoS parameters. Although there are many definitions of network lifetime [57], in line with the proposed method's node-based design, we define it as the time until the energy of any node is depleted.
3. **Average end-to-end delay:** The end-to-end delay of a packet is the time between its sending by the source node and its reception by the LBR. Since IIoT applications often use control data and environmental information for decision-making and action, they are time-sensitive, and excessive delays can lead to safety issues or system failures [3].
4. **Maximum end-to-end delay:** While the average end-to-end delay provides a general overview of packet delay, it does not capture the range of delays, which is important for IIoT applications that require bounded delay [47]. Therefore, in addition to the average end-to-end delay, the maximum end-to-end delay is checked in the evaluations.

6.2. Results

This section presents the evaluation results for the two scenarios introduced in the previous section. In both scenarios, nodes send data packets to the LBR using an MP2P communication pattern. We compared the proposed RTCL design with two other methods: RPL-TSCH Standard (RTS) and ILP3+TASA [41]. RTS represents the standard implementation of RPL and TSCH based on the IETF 6TiSCH Protocol stack [58]. The goal of selecting RTS for comparison is to assess the performance of the proposed cross-layer approach against a standard separation-layer method, allowing for better identification and evaluation of the advantages of using a cross-layer approach. Additionally, compared with the cross-layer approach, ILP3+TASA, one of the most recent papers in this research field, evaluates RTCL's performance against another cross-layer method.

6.2.1. First scenario: network size

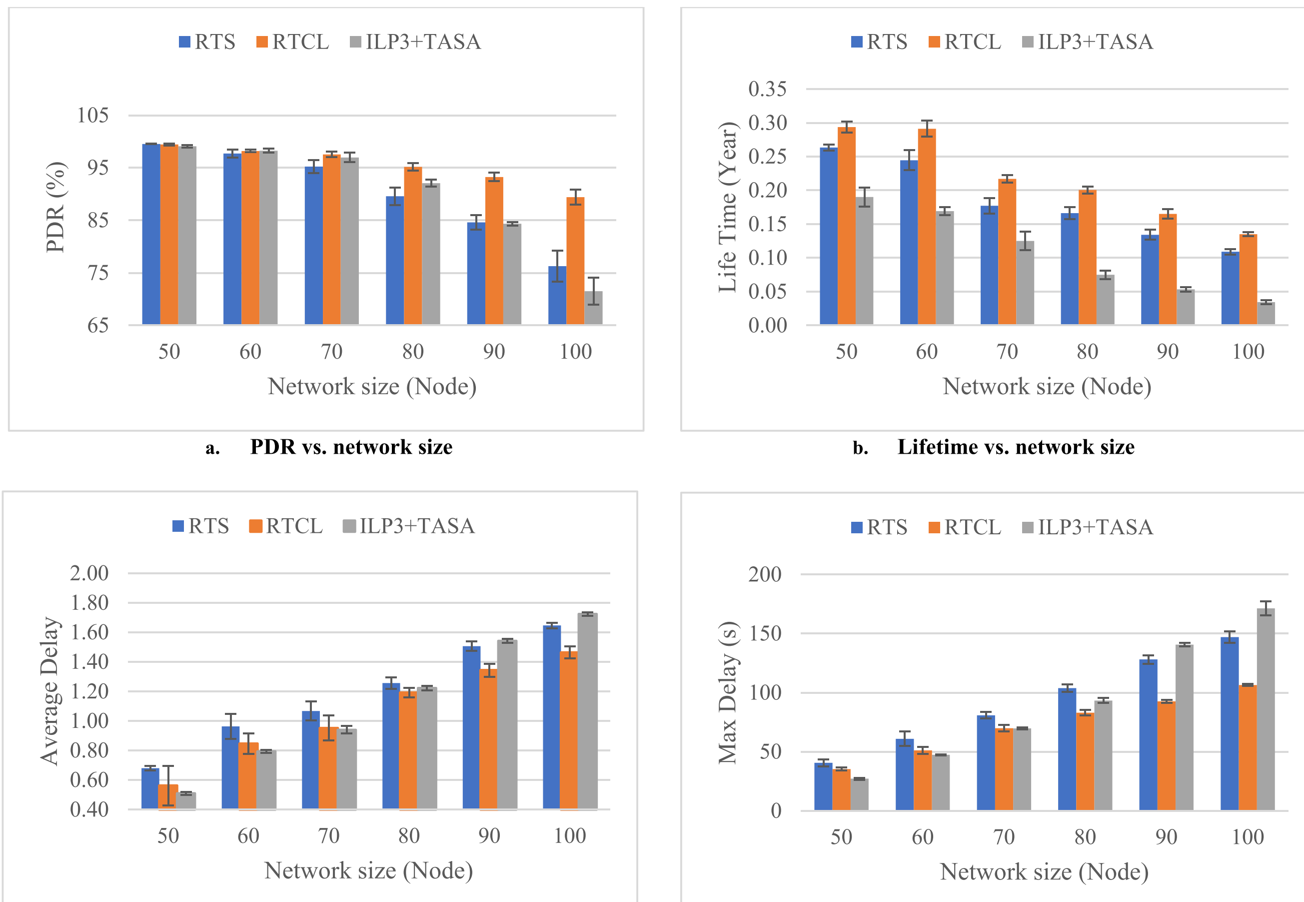
Given that IIoT networks can vary in size, this scenario focuses on the impact of network size on the evaluated methods. We consider network sizes ranging from 50 to 100 nodes, with a fixed TSCH queue size of 10 packets. Fig. 8 presents the results based on the evaluation criteria.

Fig. 8a shows the PDR results and indicates that PDR decreases as the network size increases in all three methods. However, the proposed method consistently outperforms the RTS and ILP3+TASA methods, and the advantage grows with increasing network size. This superiority can be explained by the multi-parent approach introduced in steps 1 to 4 of our method. The RTCL provides child nodes with multiple options for packet transmission, thereby improving queue utilization, increasing the service rate in child nodes, and controlling the arrival rate at parent nodes. As depicted in Fig. 5 and Eq. (10), *increasing service rate* and *controlling node arrival rates* reduce ρ and the *queue overflow* probability, enhancing PDR. Moreover, RTCL selects the RCPL using filtering and limiting operators based on QL and ETX to reduce *queue overflow* and link loss probabilities. In contrast, the RTS and ILP3+TASA methods, which employ a single-parent approach, cannot efficiently utilize the network's queue capacity, leading to a significant PDR decline as network size increases. The RTCL design, however, mitigates this issue, resulting in a growing disparity in PDR between RTCL with the RTS and ILP3+TASA as network size increases.

Fig. 8b highlights the network lifetime across different network sizes. The first notable observation is the reduction in network lifetime with increasing network size. As the network size grows, the number of packets forwarded by intermediary nodes, particularly those near the LBR, increases significantly, leading to higher energy consumption and a reduced network lifetime. The second observation is the improvement in network lifetime achieved by the proposed method compared to the RTS and ILP3+TASA methods. This improvement is attributed to two primary factors: first, the load balancing achieved by the multi-parent approach in RTCL, which distributes transition packets among several

Table 4
Sensitivity Analysis of RTCL to QL^T and ETX^T.

ETX^T \ QL^T	50%	60%	70%	80%	90%
0.5	65.28	68.40	67.69	63.94	60.19
0.6	75.37	77.94	84.90	84.72	72.37
0.7	81.63	84.68	85.11	86.61	84.77
0.8	80.15	82.47	**88.65**	84.61	78.69
0.9	73.26	80.04	85.43	83.69	78.95



a. **PDR vs. network size**

b. **Lifetime vs. network size**

c. **Average end-to-end delay vs. network size**

d. **Maximum end-to-end delay vs. network size**

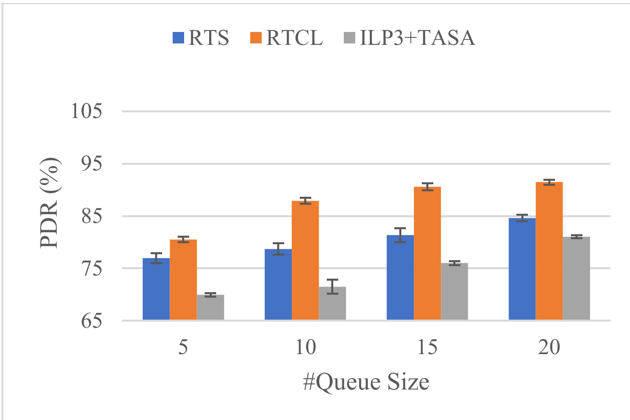
Fig. 8. Evaluation results in the first scenario.

parents, alleviating pressure on the preferred parents' energy resources; second, the improved PDR (Fig. 8a), which reduces packet loss and retransmissions, leading to better energy conservation and extended network lifetime. Due to its centralized approach, the ILP3+TASA method requires data collection from across the network, which leads to increased energy consumption, particularly in large-scale networks. As a result, it exhibits the worst network lifetime performance among the methods compared.

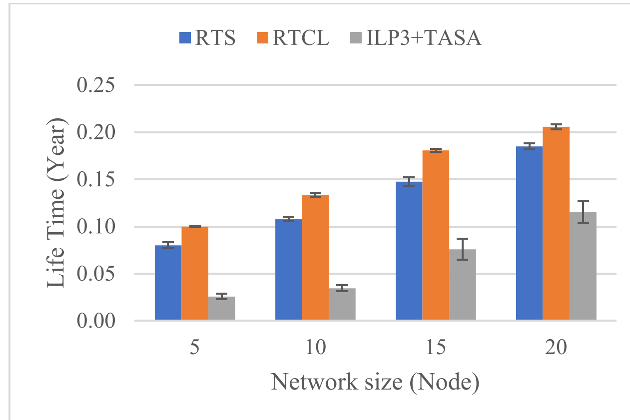
Fig. 8c and d show the average and maximum end-to-end delays as network size changes. In these figures, as the network size increases, the average and maximum end-to-end delays grow. Given that queue waiting has the greatest impact on end-to-end delay, two primary factors drive the increase in these delays as the network expands: the growth in the number of route queues and the increase in queue lengths due to the higher packet count. Despite this, RTCL outperforms the RTS in all network sizes and the ILP3+TASA in networks with size >80 nodes. The filtering operator in step 3 eliminates parents with long queues, while RTCL distributes packets across RCPL parents. Additionally, steps 5 and 6 of the proposed method ensure that newly received packets are scheduled with minimal queue waiting. Hence, RTCL reduces queue length and waiting time, lowering average and maximum end-to-end delays. ILP3+TASA aims to reduce the delay by scheduling all the packets in the queue of the root's child nodes within a single slotframe. However, as the network size increases, the proportion of packets in the queues of the root's child nodes decreases relative to the entire network, which reduces this method's efficiency.

6.2.2. *Second scenario: Queue size*

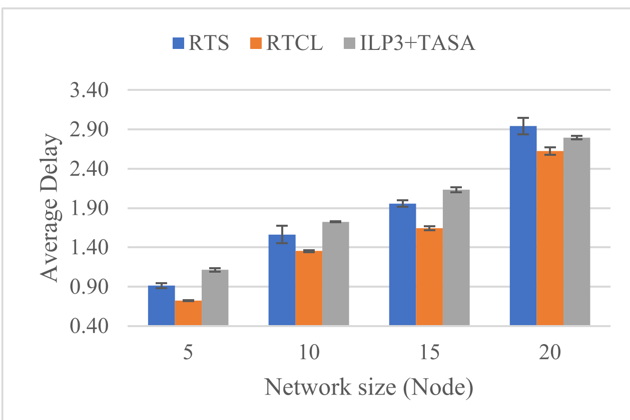
This scenario aims to investigate the effects of varying queue sizes on QoS parameters and the effectiveness of the techniques employed by RTCL to enhance queue efficiency. The tests were conducted with a fixed network size of 100 nodes and queue sizes ranging from 5 to 20 packets.



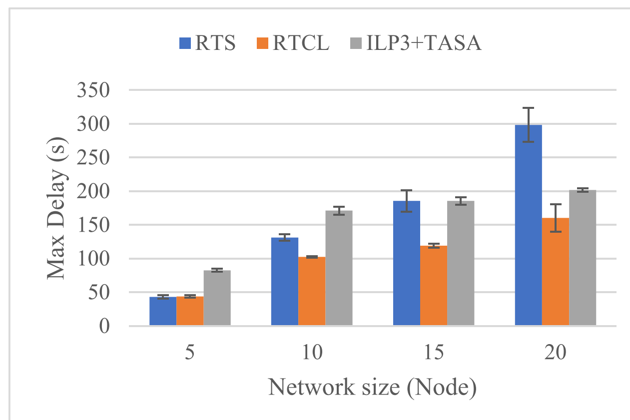
a. **PDR vs. queue size**



b. **Lifetime vs. queue size**



c. Average end-to-end delay vs. queue size



d. Maximum end-to-end delay vs. queue size

Fig. 9. Evaluation results in the second scenario.

Fig. 9 presents the evaluation results for this scenario.

Fig. 9a shows that increasing queue size improves PDR. With a fixed sending rate and node count, increasing queue size reduces *queue overflow*, thereby improving PDR. The multi-parent approach suggested in steps 1 to 4 of the RTCL design enables better queue utilization and a higher PDR than the RTS and ILP3+TASA. A noteworthy observation in Fig. 9a is the diminishing growth rate in PDR with increasing queue size. Specifically, the PDR values for RTCL are 80.5, 87.9, 90.6, and 91.4 percent for queue sizes 5, 10, 15, and 20, respectively, showing 7.4, 2.7, and 0.8 percent increments. This trend is consistent with the graph in Fig. 5, confirming that increasing queue size can only partially mitigate packet loss and is not a definitive solution. The ILP3+TASA method does not consider link quality when selecting parents, which results in poor PDR performance. As a result, this method exhibits the lowest PDR among the compared methods.

Fig. 9b demonstrates that all three methods improve network lifetime with increasing queue size. As shown in Fig. 9a, improving PDR by increasing queue size reduces packet retransmissions, thereby lowering node energy consumption. A secondary reason for this improvement is the RPL protocol, where reduced PDR often triggers preferred parent changes. With improved PDR, parent changes and the associated ICMP messaging decrease, allowing nodes to remain in sleep mode more frequently and conserve energy. This reasoning also applies to the RTCL design, as shown in [14], where using multiple parents instead of a single parent significantly reduces the frequency of parent changes. However, the centralized approach of ILP3+TASA has led to the highest energy consumption associated with this method.

While increasing queue size improves PDR and network lifetime, it also increases average and maximum end-to-end delays, as shown in Fig. 9. Queue waiting and packet retransmissions due to lost packets contribute to increased delays. However, the RTCL mitigates these issues by sending packets to parents with shorter queues (steps 2, 5, and 6 of the proposed method) and eliminating parents with poor link quality (steps 3 and 4), thereby reducing queue waiting and retransmissions, minimizing both average and maximum end-to-end delays compared to the RTS and ILP3+TASA methods.

7. Conclusion

This paper aims to improve QoS parameters in an IIoT network by proposing a cross-layer approach over RPL and TSCH protocols. Our analysis, aimed at identifying fundamental factors affecting QoS parameters, revealed that packet loss can be the initial link in a chain of effects impacting QoS. Packet loss not only reduces reliability but also leads to increased delay, bandwidth, and energy wastage due to the need for retransmissions. Tracking the causes of packet loss directed us to two main factors: *queue overflow* and *maximum retries*, with *queue overflow* playing a significantly greater role in packet loss. Through an analysis based on queuing theory and simulation, we concluded that preventing *queue overflow* requires *controlling the arrival rate* and *increasing the service rate* of the node's queue. Meeting these requirements necessitates the consideration of cross-layer information, such as candidate parent list, RPL DODAG structure, communication link quality, scheduled slotframe, and queues state; data is often scattered across different protocol layers.

Motivated by these insights, we proposed the RTCL design, a cross-layer solution for RPL and TSCH protocols. RTCL employs a multi-parent strategy, refining candidate parents based on queue length, ETX, and link quality, while prioritizing them based on scheduled slotframe to *control arrival rates* and *increase service rates*. The key objectives of RTCL are as follows:

- Reduce packet loss due to *queue overflow* and link issues, increasing reliability and throughput.
- Minimize the average queue length, reducing queue waiting times and packet delays.
- Distribute the load across multiple parents, balancing the network load more effectively.
- Improve energy consumption distribution among nodes, extending the network's lifetime.

Simulation results under two scenarios, varying network size and queue size, demonstrated that RTCL outperforms the RTS and ILP3+TASA regarding PDR, average end-to-end delay, maximum end-to-end delay, and network lifetime. Additionally, the solution enhances load balancing and network stability, contributing to better network scalability, flexibility, and energy efficiency. Another notable outcome from the simulations, previously predicted by queuing theory equations, is the low sensitivity of *queue overflow* to queue size. In future work, we aim to propose a dynamic queue size adjustment approach based on ρ, adapt RTCL to support various traffic patterns and evaluate the performance of the RTCL method through real-world implementation and testing.

CRediT authorship contribution statement

Mehdi Zirak: Writing – review & editing, Writing – original draft. **Yasser Sedaghat:** Writing – review & editing, Writing – original draft. **Mohammad Hossein Yaghmaee Moghaddam:** Writing – review & editing, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

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

References

[1] C. Perera, C.H. Liu, S. Jayawardena, The emerging internet of things marketplace from an industrial perspective: a survey, IEEE Trans. Emerg. Top. Comput. 3 (4) (2015) 585–598.
[2] C. Perera, et al., A survey on internet of things from industrial market perspective, IEEE Access 2 (2014) 1660–1679.
[3] H. Xu, et al., A survey on industrial internet of things: a cyber-physical systems perspective, IEEE Access 6 (2018) 78238–78259.
[4] S. Munirathinam, *Industry 4.0: Industrial Internet of Things (IIOT)*, in *Advances in Computers*, Elsevier, 2020, pp. 129–164.
[5] E. Sisinni, et al., Industrial internet of things: challenges, opportunities, and directions, IEEE Trans. Industr. Inform. 14 (11) (2018) 4724–4734.
[6] W.Z. Khan, et al., Industrial internet of things: recent advances, enabling technologies and open challenges, Computers & electrical engineering 81 (2020) 106522.
[7] M. Younan, et al., Challenges and recommended technologies for the industrial internet of things: a comprehensive review, Measurement 151 (2020) 107198.
[8] B. Ghaleb, et al., A survey of limitations and enhancements of the ipv6 routing protocol for low-power and lossy networks: a focus on core operations, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1607–1635.
[9] Pister, K., et al., *Industrial routing requirements in low-power and lossy networks*. 2009.
[10] J.V. Sobral, et al., Routing protocols for low power and lossy networks in internet of things applications, Sensors 19 (9) (2019) 2144.
[11] K. Ovsthus, L.M. Kristensen, An industrial perspective on wireless sensor networks—A survey of requirements, protocols, and challenges, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1391–1412.
[12] H. Kurunathan, et al., IEEE 802.15. 4e in a nutshell: survey and performance evaluation, IEEE Commun. Surv. Tutor. 20 (3) (2018) 1989–2010.
[13] G. White, V. Nallur, S. Clarke, Quality of service approaches in IoT: a systematic mapping, J. Syst. Softw. 132 (2017) 186–203.
[14] O. Iova, et al., The love-hate relationship between IEEE 802.15. 4 and RPL, IEEE Commun. Magaz. 55 (1) (2016) 188–194.
[15] D. Resner, G.M. de Araujo, A.A. Fröhlich, Design and implementation of a cross-layer IoT protocol, Sci. Comput. Prog. 165 (2018) 24–37.
[16] M. Ghribi, A. Meddeb, Survey and taxonomy of MAC, routing and cross layer protocols using wake-up radio, J. Netw. Comput. Appl. 149 (2020) 102465.
[17] D. Xu, K. Yu, J.A. Ritcey, Cross-layer device authentication with quantum encryption for 5G enabled IIoT in industry 4.0, IEEE Trans. Industr. Inform. 18 (9) (2021) 6368–6378.
[18] Winter, T., et al., *RPL: iPv6 routing protocol for low-power and lossy networks*. 2012.
[19] Thubert, P., *Objective function zero for the routing protocol for low-power and lossy networks (RPL)*. 2012.
[20] Gnawali, O. and P. Levis, *The minimum rank with hysteresis objective function*. 2012.
[21] Vilajosana, X., K. Pister, and T. Watteyne, *Minimal ipv6 over the tsch mode of ieee 802.15. 4e (6tisch) configuration*. 2017.
[22] Watteyne, T., M. Palattella, and L. Grieco, *Using IEEE 802.15. 4e time-slotted channel hopping (TSCH) in the internet of things (IoT): problem statement*. 2015.
[23] R. Estepa, et al., RPL cross-layer scheme for IEEE 802.15. 4 IoT devices with adjustable transmit power, IEEE Access 9 (2021) 120689–120703.
[24] B. Safaei, A.M.H. Monazzah, A. Ejlali, ELITE: an elaborated cross-layer RPL objective function to achieve energy efficiency in Internet-of-things devices, IEEE Internet Things J. 8 (2) (2020) 1169–1182.
[25] Y. Ha, S.-H. Chung, Traffic-aware 6TiSCH routing method for IIoT Wireless networks, IEEE Internet Things J. 9 (22) (2022) 22709–22722.
[26] J. Vera-Pérez, J. Silvestre-Blanes, V. Sempere-Payá, TSCH and RPL joining time model for Industrial Wireless Sensor networks, Sensors 21 (11) (2021) 3904.
[27] F. Osterlind, et al., Cross-level sensor network simulation with cooja, in: Proceedings. 2006 31st IEEE conference on local computer networks, IEEE, 2006.
[28] F. Assis, et al., DCS: dilution-based convergecast scheduling in a TSCH network, Ad Hoc Netw. 146 (2023) 103173.
[29] J.C. da Silva, F. Assis, Convergecast in a TSCH network under a physical interference model, in: Ad-Hoc, Mobile, and Wireless Networks: 18th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2019, Luxembourg, Luxembourg, October 1–3, 2019, Proceedings 18, Springer, 2019.
[30] J.C. da Silva, F. Assis, A distributed algorithm to schedule TSCH links under the SINR model, Des. Autom. Embed. Syst. 23 (2019) 21–39.
[31] T. Jurdzinski, D.R. Kowalski, G. Stachowiak, Distributed deterministic broadcasting in uniform-power ad hoc wireless networks, in: Fundamentals of Computation Theory: 19th International Symposium, FCT 2013, Liverpool, UK, August 19-21, 2013. Proceedings 19, Springer, 2013.
[32] T.L. Jenschke, et al., ODeSe: on-demand selection for multi-path RPL networks, in: Ad Hoc Netw, 114, 2021 102431.
[33] T.L. Jenschke, et al., Alternative parent selection for multi-path RPL networks. 2019 IEEE 5th World Forum On Internet of Things (WF-IoT), IEEE, 2019.
[34] R.-A. Koutsiamanis, et al., Meet the PAREO functions: towards reliable and available wireless networks, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020.
[35] K. Chakraborty, et al., A novel technique to parameterize congestion control in 6TiSCH IIoT networks, in: GLOBECOM 2023-2023 IEEE Global Communications Conference, IEEE, 2023.
[36] S. Duquennoy, et al., Orchestra: robust mesh networks through autonomously scheduled TSCH, in: Proceedings of the 13th ACM conference on embedded networked sensor systems, 2015.
[37] D. Deac, et al., Traffic aware scheduler for time-slotted channel-hopping-based IPv6 wireless sensor networks, Sensors 22 (17) (2022) 6397.
[38] H. Wang, A.O. Fapojuwo, Design and performance evaluation of a hysteresis-free on-the-fly scheduling function for 6TiSCH, IEEE Internet Things J. 8 (13) (2021) 10499–10508.
[39] I.F. Vieira Junior, J. Granjal, M. Curado, RT-ranked: towards network resiliency by anticipating demand in TSCH/RPL communication environments, J. Netw. Syst. Manage. 32 (1) (2024) 19.
[40] S.N. Mishra, M. Khatua, Reliable and delay efficient multi-path RPL for mission critical IoT applications, IEEE Trans. Mob. Comput. (2023).
[41] S. Wijayasekara, et al., Optimization of packet transmission scheduling and node parent selection for 802.15. 4e time slotted channel hopping (TSCH), ICT Exp. 10 (2) (2024) 442–450.
[42] M. Zhao, et al., A comprehensive study of RPL and P2P-RPL routing protocols: implementation, challenges and opportunities, Peer. Peer. Netw. Appl. 10 (2017) 1232–1256.
[43] A.J. Witwit, A.K. Idrees, A comprehensive review for RPL routing protocol in low power and lossy networks, in: International conference on new trends in information and communications technology applications, Springer, 2018.
[44] K.A. Darabkh, et al., RPL routing protocol over IoT: a comprehensive survey, recent advances, insights, bibliometric analysis, recommendations, and future directions, J. Netw. Comput. Appl. 207 (2022) 103476.
[45] P.A.M. Devan, et al., A survey on the application of WirelessHART for industrial process monitoring and control, Sensors 21 (15) (2021) 4951.
[46] E. Municio, et al., Simulating 6TiSCH networks, Trans. Emerg. Telecommun. Technol. 30 (3) (2019) e3494.
[47] L. Bommisetty, T. Venkatesh, Resource allocation in time slotted channel hopping (TSCH) networks based on phasic policy gradient reinforcement learning, Internet of Things 19 (2022) 100522.
[48] U.N. Bhat, An Introduction to Queueing Theory: Modeling and Analysis in Applications, Springer, 2008. Vol. 36.

[49] J.F. Shortle, et al., Fundamentals of Queueing Theory, John Wiley & Sons, 2018. Vol. 399.
[50] C.-H. Ng, S. Boon-Hee, Queueing Modelling Fundamentals: With Applications in Communication Networks, John Wiley & Sons, 2008.
[51] R. Nelson, Probability, Stochastic Processes, and Queueing Theory: The Mathematics of Computer Performance Modeling, Springer Science & Business Media, 2013.
[52] S.B. Yaala, F. Théoleyre, R. Bouallegue, Cooperative Resynchronization to Improve the Reliability of Colocated IEEE 802.15. 4-TSCH Networks in Dense Deployments, 64, Ad Hoc Networks, 2017, pp. 112–126.
[53] X. Huang, Quality of service optimization in wireless transmission of industrial internet of Things for intelligent manufacturing, Int. J. Adv. Manuf. Technol. 107 (2020) 1007–1016.
[54] M. Zhu, et al., A smart collaborative routing protocol for delay sensitive applications in industrial IoT, IEEE Access 8 (2020) 20413–20427.
[55] Chang, T., et al., *6TiSCH minimal scheduling function (MSF)*. Internet Engineering Task Force, Internet-Draft draft-ietf-6tischmsf-02, 2019.
[56] K. Kritsis, et al., A tutorial on performance evaluation and validation methodology for low-power and lossy networks, IEEE Commun. Surv. Tutor. 20 (3) (2018) 1799–1825.
[57] H. Yetgin, et al., A survey of network lifetime maximization techniques in wireless sensor networks, IEEE Commun. Surv. Tutor. 19 (2) (2017) 828–854.
[58] X. Vilajosana, et al., Ietf 6tisch: a tutorial, IEEE Commun. Surv. Tutor. 22 (1) (2019) 595–615.