People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University Mohamed KHider, BISKRA
Faculty of Exact, Natural, and Life Sciences
Computer Science Department

N° of order:

Series:

A thesis submitted in fulfilment of the requirements for the degree of PhD in Computer Science

Optimisation des performances des réseaux du futur

By:

Ahlam Hannachi

Abdelhamid Djeffal	PROF	President
Saber Benharzallah	PROF	Examiner
Sofiane Zaidi	MCA	Examiner
Salim Bitam	PROF	Supervisor

Abstract

The Internet of Things (IoT) is driving the evolution of low-power, scalable, and reliable networks to meet the growing demand for large-scale deployments in diverse applications such as industrial automation, healthcare, and smart cities. Within this context, the 6TiSCH standard emerges as a promising framework by integrating IPv6 for scalability and Time-Slotted Channel Hopping (TSCH) for deterministic, energy-efficient communication in low-power and lossy networks (LLNs). However, despite its advantages, 6TiSCH networks face challenges related to energy consumption and network reliability, particularly due to the overhead introduced by control packet exchanges between the 6P protocol, the scheduling function, and the routing protocol.

This thesis presents two novel contributions aimed at optimizing the performance of 6TiSCH networks by addressing challenges related to the number, size, and content of exchanged control packets. We reduce the number of 6P control packets transmitted during cell reservation through an optimized scheduling function, decreasing energy consumption while maintaining efficient resource allocation and improving overall network performance. Additionally, we minimize the size of control packets by leveraging cross-layer interactions between the scheduling function and the Routing Protocol for Low-Power and Lossy Networks (RPL), enhancing data transmission reliability and reducing communication overhead without compromising network stability or scalability.

Through detailed theoretical analysis and extensive simulations using the 6TiSCH simulator, the proposed solutions demonstrate significant improvements in energy efficiency and reliability, addressing key limitations of the existing 6TiSCH framework. These findings highlight the importance of optimizing control packet interactions and resource management in achieving high-performance IIoT networks. Ultimately, this research contributes to the development of scalable, energy-efficient, and reliable networks, forming a critical foundation for the future of IoT communication systems.

Keywords: IIoT, TSCH, 6TiSCH, SF, RPL , 6P, Energy consumption, Reliability.

ملخص

تدفع إنترنت الأشياء (IoT) تطور الشبكات منخفضة الطاقة، القابلة للتوسع، والموثوقة لتلبية الطلب المتزايد على التوزيعات واسعة النطاق في تطبيقات متنوعة مثل الأتمتة الصناعية، والرعاية الصحية، والمدن الذكية. في هذا السياق، يظهر معيار 6TiSCH كإطار واعد من خلال دمج IPv6 للتوسع و Time-Slotted Channel Hopping (TSCH) للتواصل الحتمي والموفر للطاقة في الشبكات منخفضة الطاقة وعالية الفقد (LLNs) . ومع ذلك، على الرغم من مزاياه، تواجه شبكات 6TiSCH تحديات تتعلق باستهلاك الطاقة وموثوقية الشبكة، خاصة بسبب التحميل الناتج عن تبادل حزم التحكم بين بروتوكول 6P ، وظيفة الجدولة، وبروتوكول التوجيه.

تقدم هذه الرسالة اثنين من المساهمات الجديدة تهدف إلى تحسين أداء شبكات 6TiSCH من خلال معالجة التحديات المتعلقة بعدد وحجم ومحتوى حزم التحكم المتبادلة. نقوم بتقليل عدد حزم التحكم 6P المرسلة أثناء حجز الخلايا من خلال وظيفة جدولة محسنة، مما يقلل من استهلاك الطاقة مع الحفاظ على تخصيص الموارد بكفاءة وتحسين الأداء العام للشبكة. بالإضافة إلى ذلك، نقوم بتقليل حجم حزم التحكم من خلال الاستفادة من التفاعلات بين الطبقات بين وظيفة الجدولة وبروتوكول التوجيه للشبكات منخفضة الطاقة وعالية الفقد (RPL) ، مما يعزز موثوقية نقل البيانات ويقلل من التحميل الاتصالي دون التضحية باستقرار الشبكة أو قابليتها للتوسع.

من خلال تحليل نظري مفصل ومحاكاة موسعة باستخدام محاكي 6TiSCH ، تظهر الحلول المقترحة تحسينات كبيرة في كفاءة الطاقة والموثوقية، مما يعالج القيود الرئيسية للإطار الحالي 6TiSCH . تبرز هذه النتائج أهمية تحسين التفاعلات بين حزم التحكم وإدارة الموارد لتحقيق شبكات IIoT

عالية الأداء. في النهاية، تسهم هذه الأبحاث في تطوير شبكات قابلة للتوسع، موفرة للطاقة، وموثوقة، مما يشكل أساسًا حيويًا لمستقبل أنظمة اتصالات إنترنت الأشياء.

الكلمات المفتاحية: IIoT ، 6TiSCH ، TSCH ، SF ، (RPL) ، 6P استهلاك الطاقة، الموثوقية.

Abstract in Frensh

L'Internet des Objets (IoT) stimule l'évolution de réseaux à faible consommation d'énergie, évolutifs et fiables pour répondre à la demande croissante de déploiements à grande échelle dans des applications diverses telles que l'automatisation industrielle, la santé et les villes intelligentes. Dans ce contexte, la norme 6TiSCH émerge comme un cadre prometteur en intégrant l'IPv6 pour l'évolutivité et le Time-Slotted Channel Hopping (TSCH) pour une communication déterministe et économe en énergie dans les réseaux à faible consommation et sujets à des pertes (LLNs). Cependant, malgré ses avantages, les réseaux 6TiSCH font face à des défis liés à la consommation d'énergie et à la fiabilité du réseau, en particulier en raison de la surcharge introduite par les échanges de paquets de contrôle entre le protocole 6P, la fonction de planification et le protocole de routage.

Cette thèse présente deux contributions novatrices visant à optimiser les performances des réseaux 6TiSCH en abordant les défis liés au nombre, à la taille et au contenu des paquets de contrôle échangés. Nous réduisons le nombre de paquets de contrôle 6P transmis lors de la réservation de cellules grâce à une fonction de planification optimisée, diminuant ainsi la consommation d'énergie tout en maintenant une allocation efficace des ressources et en améliorant les performances globales du réseau. De plus, nous minimisons la taille des paquets de contrôle en exploitant les interactions inter-couches entre la fonction de planification et le Protocole de Routage pour Réseaux à Faible Consommation et Sujets à des Pertes (RPL), améliorant la fiabilité de la transmission des données et réduisant la surcharge de communication sans compromettre la stabilité ou l'évolutivité du réseau.

Grâce à une analyse théorique détaillée et à des simulations étendues utilisant le simulateur 6TiSCH, les solutions proposées démontrent des améliorations significatives en matière d'efficacité énergétique et de fiabilité, traitant les principales limitations du cadre 6TiSCH existant. Ces résultats soulignent l'importance d'optimiser les interactions des paquets de contrôle et la gestion des ressources pour atteindre des réseaux IIoT à haute performance. En fin de compte, cette recherche contribue au développement de réseaux évolutifs, économes en énergie et fiables, formant une base essentielle pour l'avenir des systèmes de communication IoT.

Mots-clés : IIoT, TSCH, 6TiSCH, SF, RPL, 6P, Consommation d'énergie, Fiabilité.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Salim Bitam, for his invaluable guidance, continuous support, and patience throughout the course of this research. I am especially grateful for his trust in my competence during such a challenging time. His support was instrumental in helping me move forward and complete my work during this difficult period.

I am also sincerely thankful to Professor Wael Jaafar from École de technologie supérieure (ÉTS) in Montreal for his valuable input and thoughtful discussions, which contributed meaningfully to the development of this research.

A special thanks goes to Professor Roland Stam from the University of Aix-Marseille for his warm encouragement and personal support throughout my academic journey. His kindness and generosity were deeply appreciated, especially during tough times.

I am profoundly grateful to my family for their unwavering support, encouragement, and love. Their belief in me has been a constant source of strength and motivation throughout this journey.

Finally, I extend my appreciation to all those who supported me, directly or indirectly, throughout this endeavor.

Contents

List of Figures

List of Tables

General Introduction

The ubiquity of internet is virtually inescapable in contemporary life (or we can say industry) regardless of the realm whether smart city, agricultural , industrial or medical. The Industrial Internet of Things (IoT) it is expected to connect several billions of devices to the traditional Internet. A sizeable part of these devices are small electronic objects that are low-cost and have limited capabilities in terms of energy reserves, computation, storage, and communication. The success of the IoT relies on the ability of these small objects to run reliably and sustainably while being connected to the Internet.

The Medium Access Control (MAC) layer plays a crucial role in controlling the reliability of the network. It is responsible for controlling access to the shared communication medium (a wireless channel). It determines how devices in the network can transmit data and it is critical for ensuring that data is transmitted accurately and efficiently. The IEEE.802.15.4e is defined to meet the industrial application requirements. However, it exhibits three operational MAC modes : Time Slotted Channel Hopping (TSCH), Deterministic and Synchronous Multi-channel Extension (DSME), and Low Latency Deterministic Network (LLDN). But TSCH mode is considered as the most effective mode [4]. The essence of TSCH is the merge of time division (TDMA) with the multichannelling (FDMA), which yields a coherent twofold(channel, time) medium access mechanism. The consistency of devices act reflects the TSCH efficiency, thus, the device have to do the appropriate act (listen, transmit, sleep) in the appropriate channel and at the appropriate time. The time is organized in $slotframe$ which is a set of time-slots that repeats automatically over time. The abstraction of the coalition channel, time is *cell*. Where the *cell* is the core of TSCH mode and it have ot be delivered to devices without inequity and according to the network reliability. However, TSCH only defined a multi-channel usage of the original IEEE 802.15.4 without providing a full solution to how cells are distributed, optimized and maintained.

Due to IPv4 addresses exhaustion and IPv6's user-friendliness, the transition to IPv6 has become fairly urgent. Considering the significant difference in the protocol format and behavior, the IPv4 and IPv6 are not inter-operable and their coexistence needs a large set of protocols and procedures. For that reason, the Internet Engineering Task Force (IETF) has define the 6TiSCH working group to set standard and specification to enable IPv6 over TSCH mode. The 6TiSCH working group standardizes a protocol stack that encompasses all the necessary components for an IPv6 network to function over IEEE 802.15.4 Time-Sensitive Channel Hopping (TSCH) links, where the minimal scheduling function (MSF) [5] is the default scheduling function. However, the issue of managing and distributing the TSCH cells (TSCH schedule) is not fully addressed. 6*top* [6] is the layer immediately adjacent to 802.15.4e TSCH phy layer, It outlines the function and duties of the scheduling algorithm (SF) and how to add/delete cells, whereas The scheduling function defines when to add or delete cells with regards the key performance factors.

MSF is based on random cell allocation according to the traffic demand, which makes it unsuitable in some aspects such as for large-scale and dense deployments due to internal collisions. The schedule have to be designed according to the specific requirements of the network application, such as the network capacity, the required latency. For example, in a low-latency and high-reliability industrial control system, the schedule may need to ensure that critical control messages are transmitted with a high priority and minimal delay. In contrast, a low-power building automation system may prioritize energy efficiency over latency. In short, the 6TiSCH schedule must be designed to meet the specific needs of the network application in order to ensure efficient and effective operation. In [2], the authors compare the TSCH schedule in terms of different parameter.

The choice to focus this research on 6TiSCH stems from its central role in enabling deterministic, reliable, and energy-efficient communications within the Industrial Internet of Things (IIoT). In contrast to traditional low-power wireless protocols such as IEEE 802.15.4 or Zigbee, which rely on contention-based channel access and consequently suffer from collisions and unpredictable latency, 6TiSCH combines TSCH mode of IEEE 802.15.4e with IPv6 networking. This integration ensures time-synchronized communication, channel diversity, and end-to-end IP interoperability, establishing 6TiSCH as one of the most promising standards for large-scale industrial automation and mission-critical IoT deployments.

In our proposals, we suggest a 6TiSCH schedule that uses the RPL control packet during the parent switch to reduce the number of the control packets. Then the node defines the slot frame length and the cell reservation autonomously. defacto, the slotframe length is a critical point in the 6tisch networks. it affects the latency, throughput.

The solution we have proposed saves energy, gives a considerable reduce the overhead of control packet.To validate those performances, we used the 6TiSCH simulator due it is well-known in this context.
We have organized the structure of the thesis in five chapters:
The first, presents generalities about IoT networks, definitions, protocol stack. In the second chapter we mention some challenges that face the performance of 6TiSCH networks.

Moreover, the third and the forth chapters, we include the development of our solutions in full details. While the last chapter, we evaluate by simulation the performance of our contributions in terms of the most important key performance factors including packet delivery ratio, jitter, E2E delay, join time, life time, energy drain. Finally, we conclude our work by a general conclusion.

Chapter 1

Future Networks: Bridging Challenges and Solutions

1.1 Introduction

The evolution of communication networks has sparked a transformative shift toward highly interconnected and intelligent systems, forming the backbone of the Internet of Things (IoT) and the broader vision of Future Networks [7]. These next-generation networks are designed to connect billions of devices seamlessly while tackling key challenges such as scalability, energy efficiency, and heterogeneity. As IoT applications continue to expand—from smart cities and autonomous systems to precision agriculture and healthcare—the demand for reliable, efficient, and adaptable network infrastructures has grown exponentially.

The Industrial Internet of Things (IIoT) refers to the use of interconnected sensors, devices, and machinery in industrial settings, allowing the collection, sharing, and analysis of data to improve efficiency, productivity, and reliability. Unlike traditional consumer IoT, IIoT often deals with mission-critical operations where downtime or errors can result in significant financial loss or safety concerns. Typical applications include factory automation, predictive maintenance, supply chain management, and energy optimization.

A critical pillar of IIoT is Low-Power and Lossy Networks (LLNs), which are tailored to support resource-constrained devices operating in dynamic and often unpredictable environments. LLNs are defined by their limited energy reserves, intermittent communication links, and constrained computational capabilities. Despite these limitations, LLNs serve as the foundation for IoT applications that require both energy-efficient and dependable connectivity, including environmental sensing, industrial automation, and wearable devices.

These constraints make it difficult to maintain reliable communication and achieve the required performance levels, especially as the scale of IIoT networks increases. Furthermore, the diversity of IIoT applications demands that networks be adaptable, scalable, and able to support heterogeneous devices with different capabilities.

This chapter introduces general aspects of LLN networks, beginning with the challenges they face and subsequently exploring potential solutions.

1.2 Future Network Challenges

Given these challenges, performance optimization in LLN networks is essential. Optimizing network performance addresses these issues by enhancing key metrics such as:

1. **Energy Efficiency:** Since devices are battery-powered or energy-constrained, minimizing power consumption is crucial for extending device lifetimes and ensuring sustainability.
2. **IP Scalability:** As IIoT networks grow, they must scale to handle a large number of devices without a significant degradation in performance. Efficient protocols and network architectures are essential to support this scalability.
3. **Latency and Reliability:** Real-time applications, such as industrial control systems or healthcare monitoring, require low-latency communication and high reliability to function effectively. Optimizing these metrics ensures that critical data is transmitted promptly and accurately.
4. **Throughput:** The ability to transmit a high volume of data quickly is important for applications such as video streaming or sensor data aggregation. Optimization techniques can help ensure that throughput is maximized without overwhelming the network.

5. **Adaptability:** IoT networks need to be flexible enough to handle dynamic changes in network topology, device availability, and network conditions. Optimization strategies can ensure that networks remain robust and responsive under varying conditions.

1.3 Solutions for Scalability and Energy-Efficient Networking

The design of IIoT networks poses significant challenges, requiring solutions that can scale to billions of devices while maintaining efficient and reliable communication in resource-constrained and lossy environments [8]. The networking community addresses these challenges by integrating two powerful technologies:

- **IPv6:** Resolves scalability issues through its vast address space and advanced networking features.
- **Time-Slotted Channel Hopping (TSCH):** Ensures reliable, low-power communication in lossy networks.

1.3.1 IPv6: Solving Scalability Challenges

The adoption of IPv6 in IIoT networks overcomes the address limitations of IPv4, allowing IoT systems to accommodate a vast number of devices. Additionally, IPv6 introduces features such as hierarchical routing, built-in mobility, and Quality of Service (QoS), which are critical for managing the complex, large-scale topologies of IoT networks [9], [10], [11].

1.3.2 TSCH: Ensuring Energy Efficiency in Low-Power, Lossy Networks

TSCH (Time Slotted Channel Hopping), a core component of the IEEE 802.15.4e standard [12], addresses the unique demands of low-power, lossy networks (LLNs). Through time-synchronized communication and frequency hopping, TSCH ensures

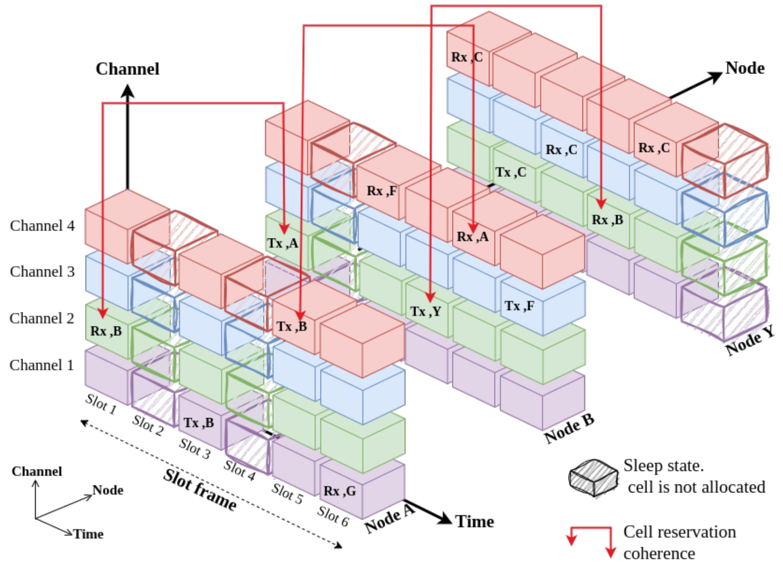


FIGURE 1.1: TSCH schedule of nodes A, B, and Y, illustrating how time-slotted and channel-based communication ensures synchronization and collision avoidance. Each timeslot is carefully coordinated to minimize interference, ensuring deterministic communication.

deterministic communication with minimal energy consumption, even in environments with significant interference or unreliable links. Its ability to reduce idle listening and retransmissions makes it ideal for energy-constrained IoT devices [13], [4].

The core principle of TSCH lies in the integration of Time Division Multiple Access (TDMA) with channel hopping techniques, resulting in a dual medium access mechanism based on both time and channel. In this mechanism, "time" is represented as a *time slot*, while "frequency" is represented as a *channel*. A combination of a specific time slot and channel is referred to as a *cell*. Within each time slot, as illustrated in Fig. 1.1, an IoT node determines its activity on a given channel, choosing to either transmit to a node (Tx), receive from a node (Rx), or remain in sleep mode. Notably, the sleep mode does not require the selection of any specific cell [14].

The efficiency of a TSCH schedule depends on the consistent and coordinated actions of devices. Each device must carefully select a combination of *(state, cell, neighbor)*, where the *state* $\in \{Tx, Rx, Sleep\}$. For example, in Fig. 1.1, node A's TSCH schedule assigns the cell *(slot 1, channel 2)* for receiving data from node B. Correspondingly, in node B's schedule, the same cell is allocated for transmitting data to node A. Additionally, node A enters sleep mode during time slots 2 and 4, while node B transitions to sleep mode during time slot 2.

1.4 The Synergy of IPv6 and TSCH

The exponential growth of IoT networks, fueled by the increasing adoption of connected devices, has underscored the importance of scalability as a core requirement for future networks. Traditional IPv4-based networks, limited by their address space and lack of advanced adaptability features, are ill-suited to handle the demands of large-scale IoT deployments. To overcome these limitations, the adoption of IPv6 has emerged as a foundational solution. With its vastly expanded address space and enhanced features for mobility, security, and QoS, IPv6 paves the way for next-generation IoT networks [15].

Building upon this framework, the Internet Engineering Task Force (IETF) established the 6TiSCH working group to develop protocols that enable the seamless integration of IPv6 over the TSCH mode of operation.

Within this context, **6TiSCH networks** emerge as a critical enabler for scalable, reliable, and efficient IoT deployments. 6TiSCH combines the Time-Slotted Channel Hopping (TSCH) mode of IEEE 802.15.4e with IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) to offer the following key advantages:

- **Massive Addressability with IPv6:** The adoption of IPv6 in 6TiSCH networks ensures that IoT systems can accommodate the billions of devices expected in the coming decades. This is especially crucial for Industrial IoT (IIoT) applications where large-scale, dense deployments are common.
- **Deterministic Networking at Scale:** The TSCH mechanism in 6TiSCH provides deterministic communication by organizing the network into time slots and channels. This prevents congestion and ensures predictable performance, even as the network scales up in size and complexity.
- **Hierarchical Routing with RPL:** By leveraging RPL, 6TiSCH networks effectively manage large-scale topologies through the use of Directed Acyclic Graphs (DAGs). This enables efficient routing and minimizes control overhead, ensuring scalability in terms of both device count and geographic coverage.

- **Energy Efficiency in Dense Networks:** As IoT networks grow, managing energy consumption becomes a significant challenge. 6TiSCH achieves energy efficiency through its time-synchronized communication and minimal idle listening, allowing networks to scale without compromising battery life.
- **Support for Heterogeneous Applications:** The modularity of the 6TiSCH stack, with its ability to integrate diverse scheduling functions and routing strategies, ensures that it can adapt to the needs of different IoT applications, from smart cities to industrial automation, making it a scalable solution across various domains.

By addressing these scalability and energy drain challenges, 6TiSCH networks provide a robust and future-ready foundation for IoT systems. Their ability to seamlessly integrate large numbers of devices while maintaining deterministic and energy-efficient communication positions them as a key building block for the **network of the future**.

1.5 6TiSCH Protocol Stack

The 6TiSCH network is composed of multiple protocol layers; as depicted in Fig.1.2 that operate in a cohesive stack to ensure reliable, low-power, and Internet-compatible communication. Below is a detailed description of the primary protocols involved in the 6TiSCH protocol stack:

1.5.1 Physical and MAC Layers: IEEE 802.15.4e TSCH

The basis of the 6TiSCH network is the MAC layer defined by the IEEE 802.15.4e amendment. TSCH introduces time-slotted access to the medium and channel hopping to avoid interference. The network is organized into a grid of timeslots, with each timeslot defining when and on which channel communication can occur. By allocating specific timeslots, TSCH ensures deterministic latency and reduces the likelihood of collisions.

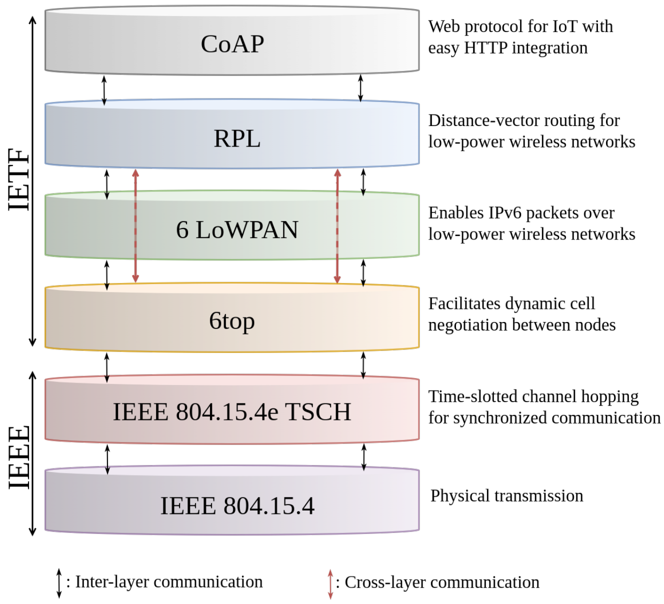


FIGURE 1.2: The 6TiSCH protocol stack, adapted from [1], showing the interplay of various layers, including IEEE standards and IETF protocols, to achieve a balance between reliability, scalability, and energy efficiency.

1.5.2 Adaptation Layer: 6LoWPAN

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) provides compression and fragmentation of IPv6 packets to fit the small frame size of IEEE 802.15.4. This adaptation layer allows efficient use of IPv6 in resource-constrained environments and supports seamless integration of 6TiSCH networks into existing IP networks.

1.5.3 Network Layer: RPL

RPL (Routing Protocol for Low-Power and Lossy Networks) is a proactive routing protocol designed specifically for Low-power and Lossy Networks (LLNs) [16]. In 6TiSCH, RPL creates a Destination Oriented Directed Acyclic Graph (DODAG), optimizing routing paths based on metrics such as energy and reliability. It supports multi-hop routing to ensure data reaches its destination in the most efficient way possible.

1.5.4 Scheduling Mechanisms: 6TiSCH Minimal and 6TiSCH Operation Sublayer

6TiSCH Minimal Configuration defines the default behavior for a basic operational schedule that can be used when the network is first established. This minimal schedule provides the basic level of connectivity needed for the initial communication and management of network nodes. The 6TiSCH Scheduling Function (e.g., SF0) allocates time slots to nodes dynamically based on application demands, optimizing performance for diverse scenarios.

1.5.5 Transport Layer

UDP (User Datagram Protocol) serves as the transport layer in 6TiSCH networks. Its lightweight nature and low overhead make it ideal for resource-constrained devices, enabling efficient transport of data without extensive session management.

1.5.6 Application Layer: CoAP

CoAP (Constrained Application Protocol) is a RESTful protocol optimized for low-power and lossy networks. It supports lightweight, asynchronous communication, making it suitable for monitoring and control applications in industrial IoT scenarios.

1.5.7 Management and Security Protocols

6P (6top Protocol) enables neighbor nodes to negotiate timeslot allocation autonomously. This management protocol helps maintain optimal link usage based on traffic conditions. Security in 6TiSCH is handled at various layers, including link-layer encryption and end-to-end encryption at the application layer using Datagram Transport Layer Security (DTLS).

1.5.8 Synergistic Protocol Layers in 6TiSCH Networks

In 6TiSCH networks, the protocol stack is characterized by an intricate interdependence among its components, rather than a strictly layered architecture. At the core of this structure is the Time-Slotted Channel Hopping (TSCH) MAC layer, which serves as a foundational element influencing the performance of higher-layer protocols. The TSCH layer not only manages channel access and time synchronization but also plays a crucial role in coordinating the activities of upper protocols such as RPL (Routing Protocol for Low-Power and Lossy Networks), 6top, and CoAP (Constrained Application Protocol).

For instance, the scheduling and neighbor management functionalities provided by 6top are directly affected by the TSCH layer's operation, which determines the availability of time slots and communication channels. Similarly, the RPL protocol relies on the TSCH layer to ensure timely data transmission and efficient routing of packets, as it must adapt its routing metrics based on the temporal and frequency characteristics dictated by TSCH. Moreover, CoAP, designed for resource-constrained environments, benefits from the reliability offered by TSCH's time-slotted architecture, enabling it to efficiently manage request-response interactions in a low-power context.

This interconnectedness is vital for optimizing network performance; the effective collaboration among these protocols, facilitated by the TSCH layer, enhances overall efficiency and reliability. Consequently, the success of a 6TiSCH deployment hinges on the harmonious interaction of its protocol stack rather than isolated functionalities, underscoring the need for a cohesive approach to protocol design and implementation in these networks. As illustrated in Figure 1.3, the 6TiSCH protocol stack emphasizes the strong interrelations between TSCH, 6LoWPAN, RPL, and other components, each contributing to a cohesive and reliable communication framework.

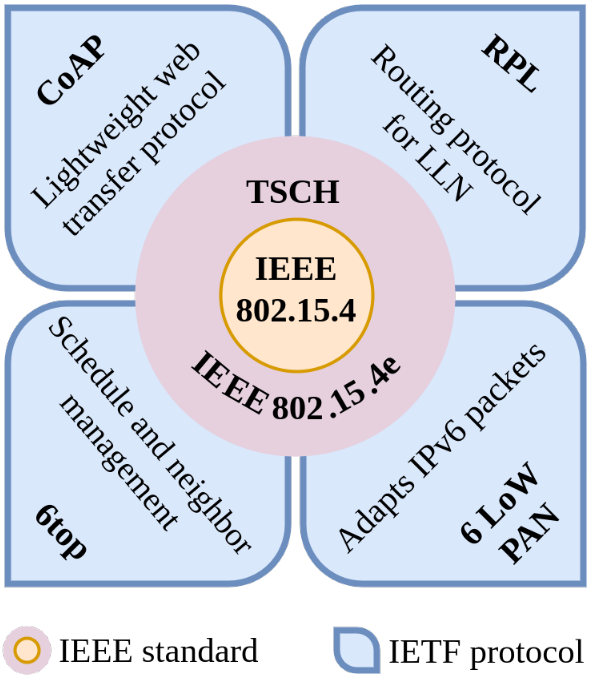


FIGURE 1.3: The 6TiSCH protocol stack emphasizing the strong interrelations between TSCH, 6LoWPAN, RPL, and other components. Each layer contributes to achieving a cohesive and reliable communication framework.

1.6 Conclusions

The 6TiSCH protocol faces several challenges that need to be addressed to optimize its performance in IIoT environments. These challenges, including parent switching, time slot allocation, control packet overload, and energy inefficiency, highlight the need for further improvements to enhance the protocol's scalability and reliability. In the following section, we will explore potential solutions to these issues, aiming to optimize the protocol for more efficient operation in large-scale networks.

Chapter 2

6TiSCH Overview: Background, Challenges, and Literature

2.1 Introduction

The Internet of Things (IoT) consists of thousands of interconnected nodes that communicate and exchange data. In most cases, the configuration of sensor nodes is primarily handled at the Medium Access Control (MAC) layer to optimize the transmission of monitoring data in terms of key performance metrics such as throughput, latency, and delay. The MAC layer plays a crucial role in ensuring the reliability and efficiency of wireless sensor networks, as it manages medium access, channel assignment, error control, energy consumption, latency, and throughput.

The performance of 6TiSCH networks relies heavily on the integration of efficient resource management strategies and reliable communication protocols. In this chapter, we provide a comprehensive review of the existing literature to establish the theoretical and technical background of this work. The chapter is organized into two main sections: the foundational background of 6TiSCH networks and an overview of related works aimed at addressing the challenges within this domain.

2.2 6TiSCH Background

The 6TiSCH standard defines a set of mechanisms to enable deterministic, low-power, and reliable communication in IoT networks. Key components, including

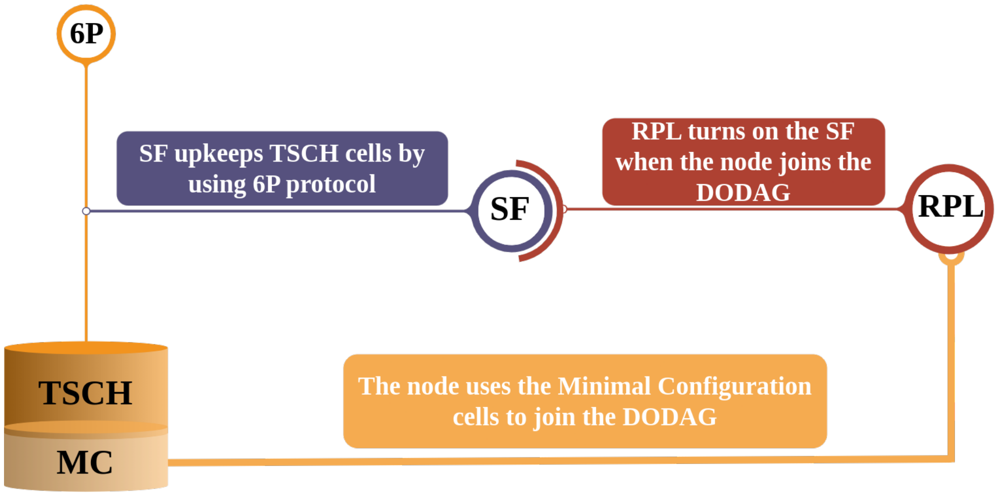


FIGURE 2.1: Interplay of 6TiSCH protocols for network management.

the Minimal Configuration (MC), the Scheduling Function (SF) and RPL , play pivotal roles in ensuring the seamless operation of 6TiSCH networks.

Fig. 2.1 illustrates the intricate relationships among the low-level protocols governing the operations of 6TiSCH networks. At the core of this architecture, the *TSCH mode* defines the Medium Access Control (MAC) behavior, encapsulated within the *Minimal Configuration (MC)* [17]. When an IoT node initiates its participation in the network, it transitions from a state of potential isolation to active engagement through a process known as "pledging." During this initial phase, the MC plays a crucial role in facilitating the node's integration into the network.

Once the node successfully joins the network, it must then become a member of the Destination-Oriented Directed Acyclic Graph (DODAG). This membership is vital for the node to engage in effective communication and network activities. The process of joining the DODAG triggers the scheduling function (SF), which is responsible for managing the allocation of time slots and channels that the node will utilize throughout its operational lifetime. The reservation of these resources is accomplished through the transmission of packets utilizing the 6P (6TiSCH Protocol) protocol, which ensures that the node's communication needs are systematically addressed.

In summary, the seamless integration of these protocols—MC for initial joining, RPL for DODAG participation, and 6P for resource allocation—enables IoT nodes to effectively operate within 6TiSCH networks, ensuring robust and efficient data exchange.

2.2.1 Minimal Configuration (MC)

The MC outlines the sequence of steps for a node to join and operate within a 6TiSCH network [17]. Nodes start in a "pledge" state, passively listening for Enhanced Beacons (EBs) broadcast by network nodes. Upon synchronization, nodes proceed to join the Directed Acyclic Graph (DAG) defined by the RPL protocol, as depicted in Fig.2.3. This process involves the exchange of control packets such as the DODAG Information Object (DIO) and DAO, which establish the topology and facilitate routing. Once synchronized, each node reserves a minimal cell and initiates 6P transactions to allocate additional resources based on its communication needs.

2.2.2 Scheduling Function (SF)

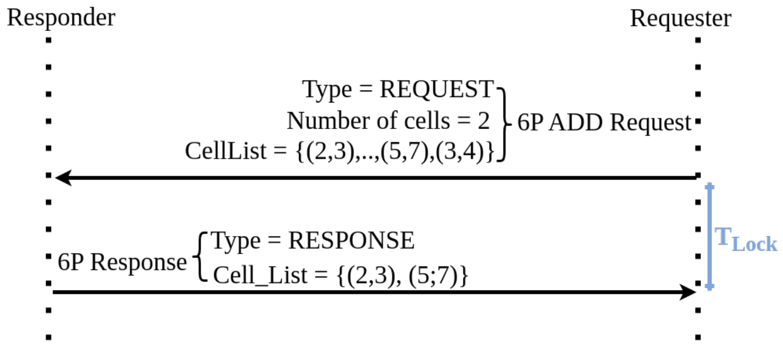
The SF manages the allocation of communication resources (time slots and channels) using 6P transactions [6, 18]. Two approaches, the 2-step and 3-step transactions, facilitate cell reservation. While the MSF ensures adaptability to traffic demands, it relies on random cell selection, which can lead to internal collisions in dense networks, thus raising concerns about scalability and QoS in demanding scenarios.

2.2.3 RPL - DODAG construction

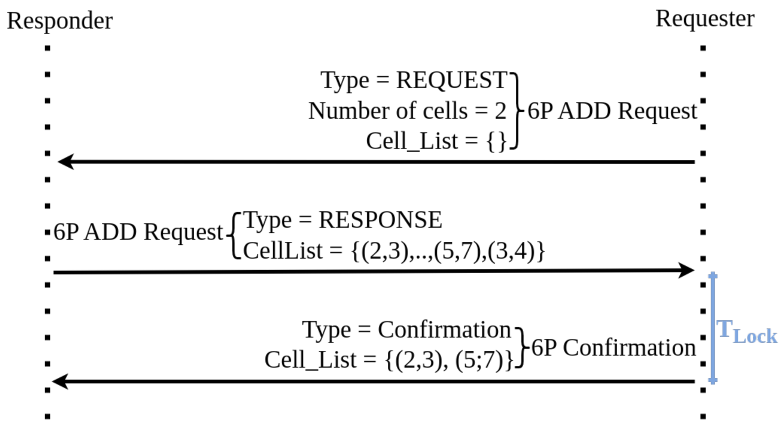
In 6TiSCH networks, RPL (Routing Protocol for Low-Power and Lossy Networks) is used as the routing protocol. RPL is responsible for establishing and maintaining routes between nodes in the network, enabling efficient communication and data transfer.

RPL in 6TiSCH networks operates in a hierarchical manner using a Directed Acyclic Graph (DAG) structure. The DAG is formed by a root node, often referred to as the DODAG (Destination-Oriented Directed Acyclic Graph) root, and a set of nodes that join the network and become part of the DODAG. Each node in the DAG has a specific role, either as a parent or a child node.

The RPL protocol governs the formation and maintenance of the DODAG, as well as the selection of optimal routes between nodes. It utilizes various metrics



(A) 2-Step 6P transaction.



(B) 3-Step 6P transaction.

FIGURE 2.2: Examples of 6P transactions.

and algorithms to determine the most efficient paths based on factors such as link quality, latency, energy consumption, and available resources.

The DODAG formation starts with the root node multi-casting DIO messages periodically, as illustrated in Fig.2.3. These DIO messages contain information about the DODAG, such as the DODAG ID, DODAG version, and the node's rank within the DODAG. The DIO messages are transmitted using multicast addresses to reach multiple nodes in the network.

When a node receives a DIO message, it evaluates whether to join the DODAG based on the information provided. The node considers factors such as the DODAG ID, the available resources, and the node's rank in relation to the advertised ranks in the DIO message.

If the node decides to join the DODAG, it becomes a member of the DODAG and selects a suitable parent node from the neighbors advertising in the DIO message. The parent selection process typically involves choosing a parent with a lower rank to ensure a downward flow towards the root. If the secure join is enabled, the node then sends a DAO (Destination Advertisement Object) message to inform its selected parent about its desire to join the DODAG.

The parent node acknowledges the DAO message by sending a DAO-ACK (Destination Advertisement Object Acknowledgment) message back to the joining node. This completes the joining process, and the new node becomes part of the DODAG and can participate in the routing and communication within the network.

The DODAG formation process ensures the construction of a hierarchical routing structure, where nodes are organized in a tree-like fashion with the root at the top and child nodes at lower levels. This structure enables efficient and scalable routing in 6TiSCH networks.

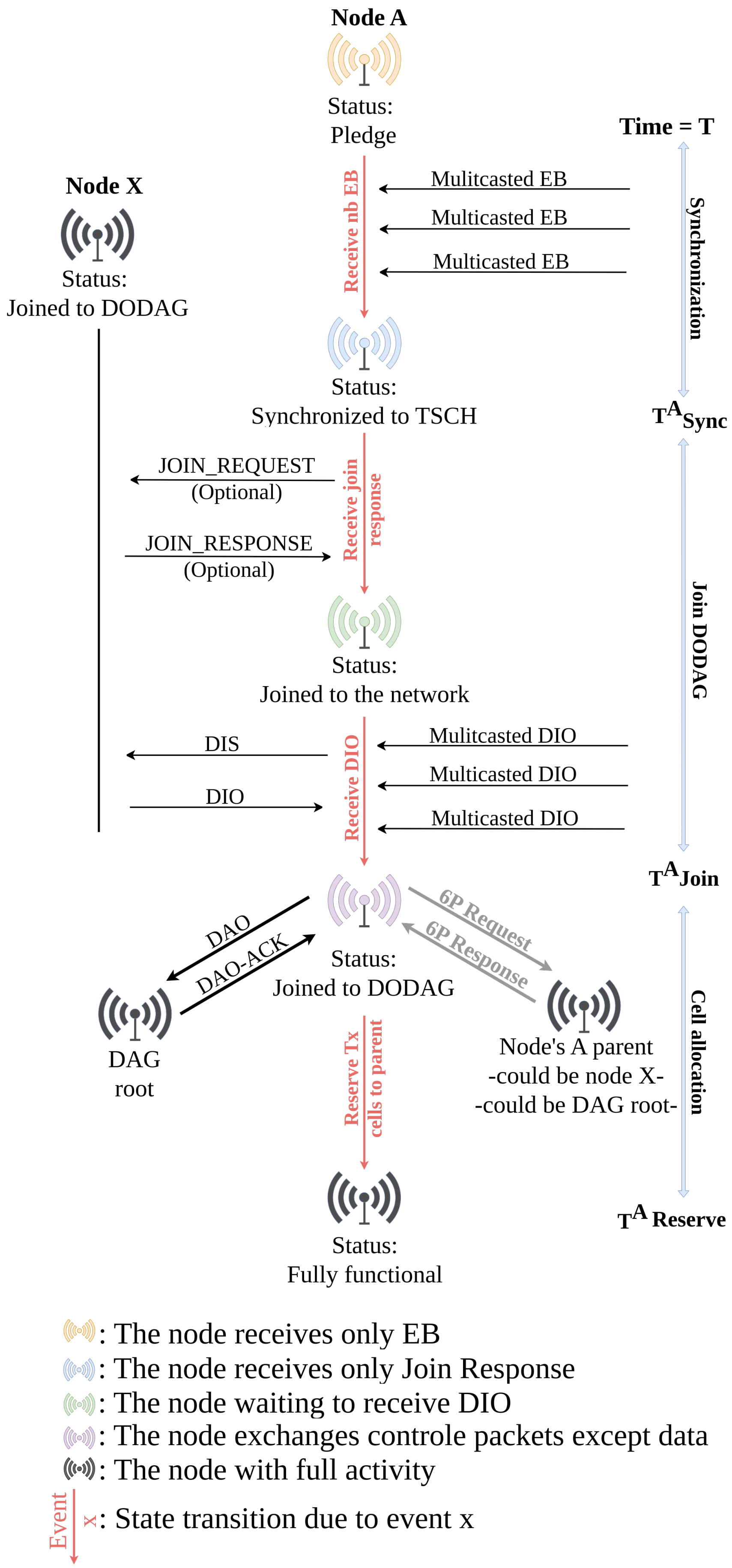


FIGURE 2.3: 6TiSCH network formation using MC.

2.3 Related Works

Research in 6TiSCH networks encompasses advancements in resource management, scheduling strategies, and routing protocols. Aiming to enhance fundamental aspects of 6TiSCH, particularly the TSCH configuration, the authors of [19] adopted heterogeneous timeslot lengths to minimize communication latency. However, the use of multiple minimal cells for multi-PHY discovery, dynamic slot duration adaptation, and the additional control overhead caused by frequent slot reconfiguration increase network overhead, memory usage, scheduling complexity, and energy consumption. In [20], the authors introduced a heuristic approach for parent selection in multi-PHY TSCH networks, demonstrating that slot bonding can efficiently adapt to different PHY data rates and achieve high Packet Delivery Ratios (PDRs). The Adaptive Static Scheduling (ASS) approach [21] optimizes resource utilization and preserves stable performance under varying network conditions. Below, we outline key contributions and highlight the challenges that persist in this field.

2.3.1 Advancements in Scheduling and Routing

In the realm of 6TiSCH networks, numerous studies have been devoted to enhancing Scheduling Functions (SFs), which can be categorized according to their methodologies and design objectives. Typically, these strategies are grouped into three main categories: centralized, distributed, and autonomous approaches. Each category presents unique advantages and trade-offs in terms of scalability, overhead, and adaptability.

In [22], the authors proposed a fully centralized scheduling strategy tailored for data-gathering applications. In this approach, each node must send an allocation request to the DAG root and await a response before transmission slots can be assigned. While this centralized mechanism enhances determinism, it incurs significant communication overhead, delays network formation, and increases energy consumption—particularly in dense networks. Similarly, the authors of [23] developed a centralized Recurrent Neural Network (RNN)-based model to predict transmission slots, aiming to minimize end-to-end latency. However, this approach suffers from high control packet overhead and imposes a considerable computational burden on nodes close to the root. Alternatively, [21] proposed the

Adaptive Static Scheduling (ASS) technique, which enhances energy efficiency by dynamically adjusting static schedules according to traffic patterns.

In contrast, distributed scheduling strategies aim to improve scalability and autonomy. For instance, the study in [24] introduced a distributed mechanism that jointly optimizes routing and scheduling decisions. Although this approach effectively reduced latency by allocating dedicated control cells to minimize collisions, it led to inefficient resource utilization in dense networks and under high mobility conditions. Moreover, the reliance on fixed downlink routes increased system complexity and generated substantial control overhead due to frequent route updates. Autonomous approaches, on the other hand, such as TESLA [25], introduced a traffic-aware elastic slotframe adjustment scheme, enabling each node to dynamically adapt its slotframe size based on real-time traffic load. This adaptive design improves resource utilization and maintains high energy efficiency without centralized coordination.

Beyond scheduling strategies, several works have focused on optimizing the Routing Protocol for Low-Power and Lossy Networks (RPL), which plays a critical role in establishing and maintaining network topology. For instance, the authors of [26] proposed a parent selection function that considers both the number of neighbors and the traffic load of potential parents, effectively reducing frequent parent switching and improving the Packet Delivery Ratio (PDR). Similarly, [27] presented a Q-learning-based algorithm to optimize the RPL objective function, while [28] introduced an early parent-switching scheme based on the remaining queue capacity of the parent node. Although these methods improved routing performance and PDR, they also introduced new challenges. The frequent parent switching and continuous queue monitoring required in [28] increase control overhead and fail to account for the possibility that congestion may arise from scheduling inefficiencies rather than actual traffic imbalance. Moreover, Q-learning-based solutions, such as [27], entail high computational complexity, significant memory requirements for Q-table storage, and frequent Q-value updates, making them less suitable for resource-constrained IIoT devices in 6TiSCH networks.

To overcome the limitations of single-layer optimization, cross-layer designs have been proposed to jointly optimize parameters across multiple layers. For example, the authors of [29] proposed a cross-layer scheduling and routing framework that enhances reliability by integrating alternative parent selection, MPLS-like source routing, and a Bounded Delay Packet Control (BDPC) mechanism. By leveraging multiple parent paths and implementing a deadline-aware packet forwarding policy, this approach significantly improved reliability and ensured bounded latency. However, it introduces increased computational and communication overhead, as each node must perform per-packet label switching, maintain multiple parent routes, and dynamically monitor packet delivery deadlines. Likewise, the Q-learning-based cross-layer approach presented in [30] improved both packet delivery and latency but required substantial computational resources, careful parameter tuning, and frequent communication for Q-table updates, thus limiting scalability.

A more recent development is the Improved Minimal Scheduling Function (IMSF), proposed in [31], which extends the standard MSF by dynamically adjusting the number of allocated cells based on queue occupancy and Expected Transmission Count (ETX). IMSF improves adaptability to varying traffic loads and reduces packet loss without introducing additional control messages beyond standard 6P transactions. Nevertheless, its dependence on ETX metrics and historical queue data may limit its responsiveness in rapidly changing network conditions.

Overall, this body of work illustrates the evolution of scheduling and routing optimization techniques in 6TiSCH networks, highlighting the trade-offs between centralization and autonomy, complexity and scalability, and performance and resource efficiency. Figure 2.4 summarizes the comparison of existing scheduling algorithms according to their design objectives, providing a comprehensive overview of the strengths and limitations of current approaches.

2.3.2 Network Formation and Parameter Optimization

Regarding 6TiSCH network formation, the study in [32] evaluated the network formation procedure and identified several limitations. The authors also proposed a set of recommendations and established guidelines for parameter adjustments, adapting them to various network topologies to improve formation efficiency. Similarly, [33] addressed performance degradation issues observed during 6TiSCH network formation, emphasizing the challenges faced by new nodes when joining the network. This degradation mainly stems from the policy of allocating only one shared cell per slotframe for control packet transmission, which restricts the transmission opportunities during the joining phase. The authors further underlined the

Algorithm	Delay Min	Slot-frame Length Min	Latency Min	Overhead Min	Reliability Max	Scalability	Parametric Evaluation						
							Packet Loss	Delay	Energy Consumption	Transmission Latency	Duty Cycle	Throughput	Signaling Overhead
TASA (Palattella et al., 2013a, 2012)	✗	✓	✗	✗	✗	✗		Low	Low		Low	High	High
MODESA(Soua et al., 2013b)	✗	✓	✗	✗	✗	✗		Low	Low			High	
Musika (Soua et al., 2013a)	✗	✓	✗	✗	✗	✗							
Queue-based algorithm (Farias and Dujovne, 2015)	✓	✗	✗	✗	✗	✗		Low	Low				
SSF$_{LC,DYN}$ (Hahm et al., 2016)	✗	✗	✗	✗	✗	✓	Low		Low	High			
AMUS (Jin et al., 2016)	✗	✗	✓	✗	✗	✗		Low	High	Low	High		
CLS (Choi and Chung, 2016)	✗	✗	✓	✓	✓	✗		Low	Low	High			
PS (Chen et al., 2016)	✗	✗	✓	✗	✗	✗		Low		Low			
Aloha and Reservation based (Tinka et al., 2010)	✗	✗	✗	✗	✗	✗							
DeTAS (Accettura et al., 2015, 2013)	✗	✓	✗	✗	✓	✗	Low	Low		Low	Low		
Label Switching (Morell et al., 2013)	✓	✗	✗	✗	✗	✗		Low	Low				
Wave(Soua et al., 2014)	✗	✗	✓	✗	✗	✗				Low			
DiSCA (Soua et al., 2015a)	✗	✗	✓	✗	✗	✗							
OTF (Palattella et al., 2016)	✗	✗	✗	✗	✓	✗	Low	Low		Low			
PID Based Scheduling (Domingo-Prieto et al., 2016)	✗	✓	✗	✗	✗	✗					High		High
DIS_TSCH (Wang and Hwang, 2015)	✓	✗	✗	✓	✗	✗		Low				Low	
Localized Scheduling (Hosni et al., 2016)	✓	✗	✗	✗	✗	✓	Low	Low				Low	
Random Scheduling with Housekeeping (Muraoka et al., 2016)	✗	✗	✗	✓	✗	✗	Low						Low
LLSF (Chang et al., 2016)	✗	✗	✓	✗	✗	✗				Low			
DeBRAS (Municio and Latré, 2016)	✗	✗	✗	✗	✗	✓			High	Low		High	
Orchestra (Duquennoy et al., 2015)	✗	✗	✗	✗	✓	✗		Low		Low	Low		
e-Queue MAC (Cao et al., 2015)	✓	✗	✗	✗	✗	✗		Low				High	
e-SF0 (Duy et al., 2017)	✗	✗	✗	✓	✗	✗	Low		Low	Low			
ReSF (Daneels et al., 2018)	✗	✗	✓	✗	✗	✗	Low		High	Low			
P-SBS (Lee et al., 2017)	✓	✗	✗	✗	✓	✗	Low						
DIVA (Demir and Bilgili, 2017)	✗	✗	✗	✗	✗	✗							
MSA (Meng et al., 2018)	✓	✗	✗	✗	✗	✗			Low				
EES (Ojo et al., 2017b)	✗	✗	✗	✗	✗	✗	Low		Low				
DeAMON (Aijaz and Raza, 2017)	✗	✗	✓	✗	✗	✗	Low			Low	Low		
LV (Kralevska et al., 2017)	✓	✗	✗	✗	✓	✗			Low	Low			
Heuristic Load Balancing (Kim et al., 2016).	✓	✗	✗	✗	✗	✗			Low				
DS-WBAN (Zou et al., 2016)	✗	✗	✗	✗	✗	✗							
Debt Based (Khoufi et al., 2017b)	✗	✓	✗	✗	✗	✗		Low		Low			
TMS-DC (Ojo et al., 2017a)	✗	✗	✗	✗	✗	✗							
HR-UT (Elsts et al., 2017)	✗	✗	✗	✗	✗	✗		Low					

FIGURE 2.4: Comparison on the Basis of Design Objectives of TSCH Scheduling Algorithms [2]

drawbacks of prioritizing Enhanced Beacons (EBs) during network formation and proposed an improved approach that more effectively integrates control packets within the network formation process.

In addition, the authors of [34] aimed to accelerate the network formation process through the adoption of active scanning procedures combined with a trickled beacon advertising strategy. In their method, nodes attempting to join transmit beacon request packets to increase their detectability by already associated nodes, thereby reducing the overall association time. Although this solution notably enhanced the network formation speed, it neglected energy efficiency aspects, as joining nodes remain in continuous listening mode during the formation phase.

More recently, [35] proposed the Enhanced Network Bootstrapping technique using Multi-Channel (EBMC) to further improve network formation performance. However, EBMC relies on a static channel selection algorithm derived from the nodes' EUI-64 addresses, which makes it vulnerable to persistent synchronization failures when constant interference affects the assigned channel. This limitation prevents certain nodes from successfully joining the network. Furthermore, since topology formation in EBMC does not include dynamic parent selection, inefficient multi-hop routes may occur, leading to increased end-to-end latency and routing overhead. Despite the benefits of multi-channel allocation in reducing collision probability, EBMC's static control packet assignments cause underutilization of channel resources during periods of low control packet activity.

2.4 6TiSCH Challenges

While the 6TiSCH protocol is widely recognized for enabling low-power and reliable communication in IIoT networks, several challenges hinder its full effectiveness. These challenges stem from issues related to parent switching, time slot allocation, control packet management, and energy consumption. The following subsections discuss these primary challenges and their impact on network performance.

2.4.1 Excessive Control Traffic

The 6TiSCH protocol relies heavily on control packets for both scheduling and routing operations. These packets are essential for managing the network's time slot allocation and ensuring proper routing through the DODAG. However, excessive reliance on control traffic leads to network congestion, particularly in large-scale deployments such as smart cities or IIoT systems.

The increasing volume of control traffic results in higher latency and reduces the network's reliability. As the application period lengthens (i.e., when data generation is less frequent), the number of control packets exchanged increases because the network attempts to schedule more data within the same time slots. This control traffic, while necessary for maintaining network operations, can overwhelm the network's capacity to efficiently manage data transmission, leading to reduced throughput and operational delays.

In order to approximate the number of the control packet during 7.5 hours, we run 50 simple simulations in the 6TiSCH simulator [3] where the slot frame length was 50 slots and time slot duration was 0.01 seconds and in fully meshed topology.

Figure 2.5 depicts the number of transmitted and received (Tx/Rx) 6P control packets in a 6TiSCH network consisting of 50 nodes, observed over a 7.5-hour simulation, with varying application/service data generation intervals of 5, 15, 45, and 60 seconds.

In a meshed topology, where nodes are close to one another, the increased control traffic exacerbates collision rates and reduces spatial reuse. This leads to further delays in data transmission and higher energy consumption. Optimizing the management of control packets is crucial to maintaining network efficiency, particularly in large-scale deployments where control traffic is expected to grow substantially.

As the application period increases, the network is required to manage a higher volume of data transmissions within each time slot. This results in a greater number of 6P control packets being exchanged for scheduling purposes. Despite 6P packets being given higher priority, the growing demand for scheduling as the application period extends leads to a substantial rise in 6P traffic. Therefore, optimizing control packet management is crucial for maintaining the network's efficiency.

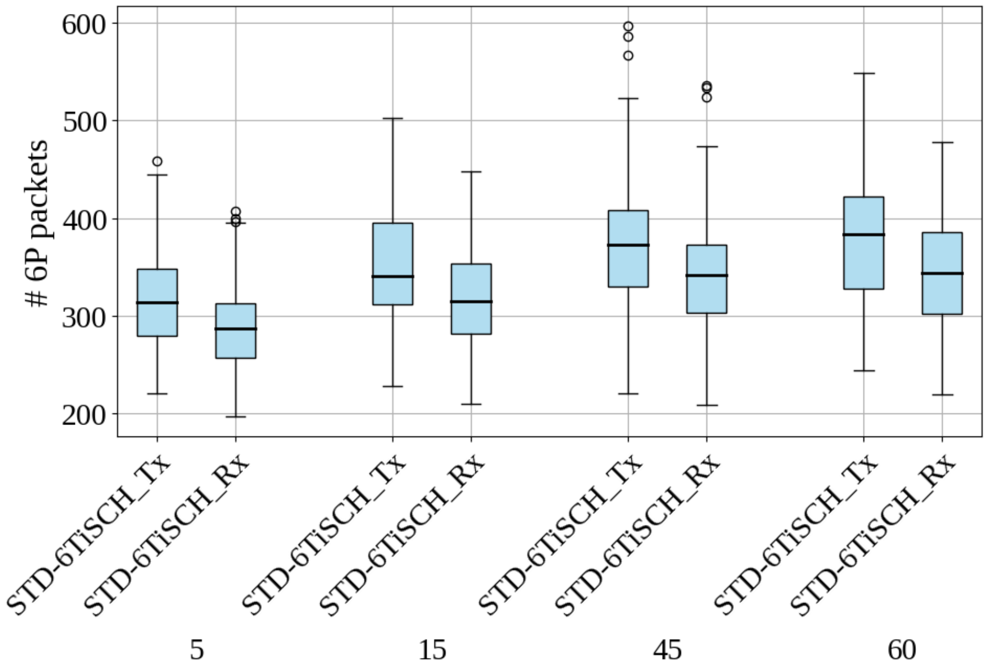


FIGURE 2.5: Number of 6P packets exchanged during 7.5 hours in a 50-node network.

The increase in control packet traffic not only exacerbates collision rates but also reduces spatial reuse, further depleting the energy of the nodes. These challenges are magnified in a meshed topology, where the proximity of nodes increases the likelihood of collisions, further impacting the overall network performance."

2.4.2 Complex Time Slot Allocation for External Nodes

Time slot allocation in 6TiSCH is another challenge, especially for nodes that are not directly part of the DODAG (Destination-Oriented Directed Acyclic Graph). These nodes require efficient integration into the network, which is particularly difficult when the topology changes due to node movement or the introduction of new nodes. The protocol struggles to provide sufficient time slots for such nodes, especially in mobile scenarios where node positions are constantly changing.

This challenge is compounded by the need for real-time reporting from nodes to maintain network synchronization. Without timely integration, these nodes face delays in joining the network, leading to increased energy consumption as they repeatedly attempt to access the network. The inability to efficiently allocate time slots for external or newly joining nodes affects network scalability and can lead to operational inefficiencies.

2.4.3 Inefficient Energy and Time Consumption during Parent Switching

In 6TiSCH networks, IoT nodes frequently switch their parent in search of the best link to the Directed Acyclic Graph (DAG)-root. Each time a node switches parents, it triggers a 6P request to reserve time slots with the new parent, and an RPL-based DAO packet is sent to inform the DAG-root of the new parent, allowing the routing table to be updated. However, this process generates a high volume of control packets, which can lead to network congestion.

The process of parent switching in 6TiSCH networks also contributes to inefficiency, particularly in terms of energy and time consumption. When a node switches parents, the RPL protocol does not account for the new parent's schedule saturation, leading to unnecessary energy expenditure and delays in packet delivery in case the new parent **schedule is full** while the switching child lacks awareness of the situation.

In IIoT environments, where devices operate on limited energy resources, this inefficiency can be detrimental. Each parent switch operation requires the node to expend energy in communication and re-synchronization. Moreover, the time required for the new child to update its routing table can cause delays in data transmission, affecting overall network performance and reducing the lifespan of devices.

2.4.4 Bottlenecks and Collisions

As the number of nodes in a 6TiSCH network increases, the risk of bottlenecks and packet collisions also rises. In a fully meshed network, nodes are in close proximity to one another, increasing the likelihood of collisions when transmitting control packets. Furthermore, nodes that share the same parent can cause congestion as they transmit and receive a high number of control packets.

The figure 2.6 illustrates this problem, showing how nodes C and G are responsible for forwarding multiple DAO packets and handling numerous 6P requests. This situation creates a bottleneck, as these nodes are tasked with handling the control traffic for multiple child nodes, leading to delays in packet delivery. When nodes become overwhelmed with traffic, delays increase, resulting in a less efficient network.

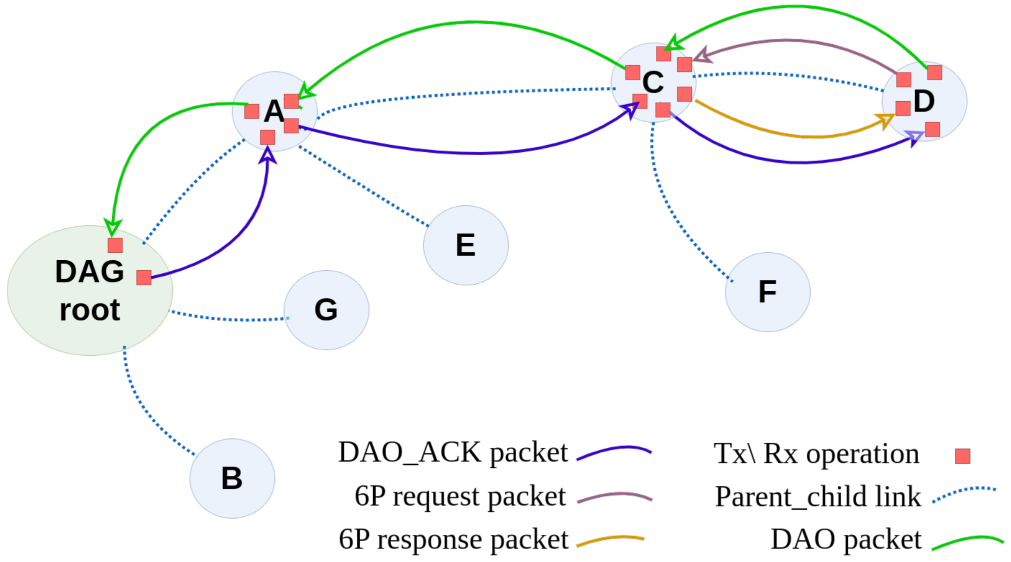


FIGURE 2.6: Control packets triggered upon parent switching.

2.4.5 Energy drain

In 6TiSCH networks, nodes are battery-powered, making energy efficiency a critical factor. Given the challenges of battery replacement in remote deployments, minimizing energy consumption is essential for ensuring long-term network sustainability and reliable operation.

Energy can be drained in several situations due to the unique characteristics and requirements of low-power and lossy environments. Here are details about the most important situations:

- Network Overhead: The control messages used for network maintenance, such as DODAG formation, route updates (DIO,DAO), and neighbor discovery, add to the network overhead. High overhead leads to more energy consumption and may reduce the effective communication time for data transmission.
- packet re-transmission: In densely deployed 6TiSCH networks or during periods of increased data traffic, devices may experience more frequent collisions and retransmissions. These activities consume additional energy and can contribute to faster battery depletion.

- Synchronization: The initial state of the device is *pledge*, at this stage the node is not synchronized and it keeps listening to randomly selected channels. The longer it takes for a node to become synchronized, the more energy it consumes.

- Join time: A node's effectiveness is dependent on its successful integration into a DODAG, defining its preferred parent, and allocating cells for communication. The negotiation mechanism between child nodes and their parents, used to reserve cells, can significantly impact energy consumption—the more packets required for this process, the higher the energy consumption incurred.

Minimizing energy consumption in 6TiSCH networks is essential to prolong the network lifetime, reduce maintenance costs, and ensure the proper functioning of battery-powered devices in various applications. To address these energy challenges, with keeping the eye on network efficiency, we propose; besides our pulse based mechanism; to reduce the number of packet used by the scheduling function and inject the needed information by the SF in the RPL packets DIO and DAO packets.

2.5 Conclusions

In conclusion, this chapter has examined the foundational aspects of 6TiSCH networks within the context of the Industrial Internet of Things (IIoT). As industries increasingly adopt IoT technologies, the need for reliable, low-power communication solutions becomes critical.

6TiSCH networks leverage the IEEE 802.15.4e TSCH standard to provide time-synchronized communication and channel hopping, ensuring high reliability and minimal interference. The integration of protocols such as 6LoWPAN, RPL, and CoAP further enhances their functionality and interoperability with IP networks. The reviewed literature reveals substantial progress in enhancing 6TiSCH networks. However, challenges such as scalability, energy efficiency, and adaptability to diverse traffic demands persist. These gaps underscore the necessity for innovative solutions that address the dynamic requirements of future IoT networks.

Chapter 3

Control Packet Balancing: A Novel Approach

3.1 Introduction

Our research delves deeply into the complex interactions between RPL and the Scheduling Function (SF), as well as the SF and packet queue management. The guiding principles of our proposed solution are:

- Reducing the transmission frequency of control packets during parent switching.
- Minimizing failed parent switching events caused by cell unavailability.
- Preventing packet loss due to queue saturation.
- Enhancing jitter and latency performance through optimized slot selection.

This approach is designed to conserve energy and reduce medium contention, ultimately improving spatial reuse efficiency.

To clarify our solution, we emphasize the following technical terms:

- The term "DAO" specifically refers to the DAO packet triggered during parent switching, unless stated otherwise. This focus is due to our emphasis on mechanisms and improvements related to parent-switching events, while the handling of periodic DAO packets remains unchanged in our approach.

- The term "DIO" may refer to either periodic transmissions or those triggered as a direct response to a DIS packet, depending on the context of the processes or mechanisms being discussed.

- "DAO-ACK" and "ACK" refer to two distinct messages. DAO-ACK is a packet sent by the DAG root to acknowledge DAO messages for reliable route registration or de-registration. On the other hand, "ACK," also known as a link-layer or TSCH acknowledgment (as shown in Fig. 3.1), is issued upon packet reception [36].

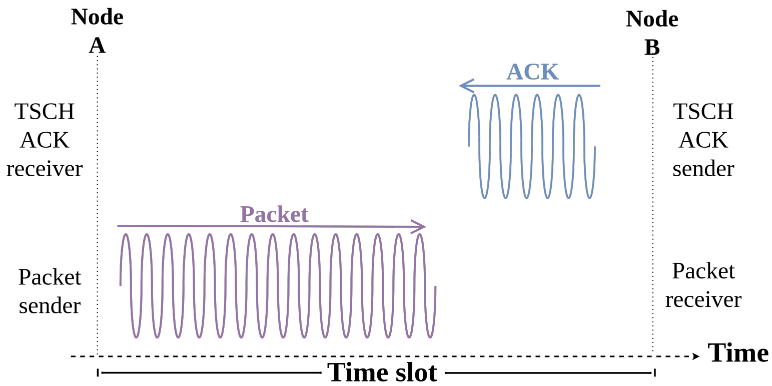


FIGURE 3.1: Packet TSCH acknowledgment.

3.2 Regulated Triggering of Parent Switching

In 6TiSCH networks, parent switching presents certain challenges. Specifically, the decision to switch to a potentially more optimal parent is complicated by the node's lack of knowledge regarding the availability of cells from the new parent. This uncertainty can result in inefficient energy use, as the node may continue transmitting control packets without securing proper cell allocation.

Furthermore, the RPL specifications [37, 38] do not specify how the protocol handles transmission interruptions to the previous parent during this transition. As a result, increased latency and potential packet loss may occur. Particularly if the node continues to receive packets but lacks sufficient cells for forwarding them. To address this issue, our solution modifies the DIO packet to include a list of available slots from the sender. This ensures that a node initiates parent switching only when a shared available slot is present, thereby improving the efficiency of the process. Algorithm 1 below outlines this mechanism, emphasizing the node's decision-making process during parent switching.

Algorithm 1: Parent switching procedure

```
Data: parents: DIO's senders
min_nb: Minimum number of slots to allocate
max_nb: Maximum number of slots to allocate
Result: Parent switching decision
nb: Number of slots to reserve
// ------ Initialization ------
nb_parentSlot ← Number of slots allocated to parent
child_slots ← Node's free slots
// ------ Parent Selection and Slot Reservation ------
selected_parent ← parent with minimum rank and available free slots
if child_slots ≥ min_nb then
    if Old parent is None then
        // Node is joining the DODAG
        nb ← min_nb
    else
        // Node is switching parent
        nb ← nb_parentSlot
        if nb > max_nb then
            // Ensure selected slots are not exceeding the max limit
            nb ← max_nb
    free_slots ← child_slots ∩ parent_free_slots
    slots ← Select nearest nb slots from free_slots
    child_slots ← child_slots − slots
    Append slots to DAO packet
    Append child_slots to DAO packet
    Transmit DAO packet
else
    Pass
    // Retain current situation since there is no slot availability
```

3.3 Minimizing 6P Transaction Packets

In the sequenced control packet transmission process of 6TiSCH networks, when a node decides to switch its parent, it signals this transition by sending a DAO packet to the DAG root and initiates the cell reservation process with the new parent.

This process involves sending a cell reservation request through a 6P transaction, as shown in Fig. 2.2.

Instead of dedicating an entire packet for this function, we propose embedding the necessary cell reservation information directly within the RPL control packet DAO. Since the next hop for each child packet is naturally its parent, all packets, including the modified DAO, will traverse the designated parent. Consequently, the parent can extract the embedded cell reservation information and allocate the appropriate cells. This method is elaborated below:

1. *Autonomous channel selection:* The transmission channel is autonomously determined based on the receiver's MAC address, allowing nodes to distribute only the slot offset to represent the cell. To mitigate interference and take advantage of multi-channeling, we follow the same strategy as MSF. Specifically, the current transmission/reception channel ($channel_{\text{ID}}$) is calculated using the Absolute Slot Number (ASN) and the channel offset ($cell_{\text{channel}}$) of the current cell, as shown below:

$$channel\_\text{ID} = (ASN + cell_{\text{channel}})\ \%\ nb\_\text{channels}, \tag{3.1}$$

   where nb_{channels} is the total number of channels and % is the remainder of the Euclidean division.

2. *RPL packet size management:* n 6TiSCH networks, where packet size is inherently limited, efficient management of RPL packets, including DIO, DAO, and DAO-ACK, is crucial. Therefore, we propose the following modifications:

   (a) *Selective data injection:* Conventionally, cells are represented as tuples consisting of slot-offset and channel-offset. Embedding these tuples into RPL packets results in increased packet size and processing time.

   To reduce packet size, we propose a simplified integer-only representation of available cells, avoiding the space-consuming tuple format.

   (b) *Slot ID encoding:* We propose the implementation of an 8-bit encoding scheme for slot IDs, thereby reducing the size of the slot ID list embedded in the packet.

(c) *Compact free slots notation:* In our proposed approach, the DAO and DIO packets must include the node's available slots. To optimize memory usage, we suggest that the node transmits either an ordered list of occupied slots or free slots, depending on which is shorter, and precedes it with a 1-bit flag. In the 6TiSCH context, slot '0' is reserved by nodes upon joining the network and, therefore, will never be part of the ordered list of free slots. This reserved status allows us to efficiently use slot '0' as the flag for the occupied/free slots list.

This flexible method ensures reduced memory usage.

The techniques outlined above align with the fundamental requirements of low-power and lossy networks (LLNs) as they aim to reduce transmission overhead, conserve energy, and improve network throughput. Additionally, they simplify demand processing, which is crucial for devices operating in constrained environments, such as those in 6TiSCH networks.

3. *TSCH ACK-based optimization of slot allocation:* For efficient time slot allocation, we propose a novel method that utilizes TSCH ACK for immediate confirmation of slot reservations upon receiving a DAO packet. This approach eliminates the conventional delays associated with reservation confirmation. By leveraging TSCH ACK, available time slots are communicated more rapidly, reducing the locking duration of time slots (denoted as T_{Lock} in Fig. 2.2) and enhancing the network's adaptability to topological changes. Our technique embeds a list of available slots (represented as integers) directly into TSCH ACK messages. To minimize data size, we limit the maximum number of allocated time slots at once to 5, adding only 5 bytes to the TSCH ACK payload.

   Furthermore, for efficient time slot reservation, we assume each node maintains an up-to-date variable containing its ordered list of available slots. This variable is dynamically updated after each slot allocation or de-allocation. Upon receiving a packet, the joining node compares the embedded list of available slots against its own available slots variable. As a result, the decision to reserve slots is based on the most current network state, improving the responsiveness and adaptability of the 6TiSCH network to real-time demands.

3.4 Reducing DODAG Joining Time

To reduce packet collisions and reduce the nodes' joining time, we propose that the DIO sender sets temporary time slots or cells, ensuring it is actively listening, and it designates one permanent cell to maintain its ongoing listening state. In scenarios where a DIO is concurrently received by multiple nodes, thus causing potential DAO collisions, the proposed mechanism that uses a combination of temporary and permanent cells can diminish the impact of this issue as follows: 1) The DAO sender temporarily allocates cells until it receives the TSCH ACK of the DAO packet. This strategy intentionally sacrifices idle listening to prioritize energy saving, given that the energy consumed during idle states is notably lower than that needed to manage collisions and trigger packet retransmissions [39]. 2) In the initial phase of network formation where nodes actively compete to join the network and parent switching events are frequent, our method sets a high number of proposed time slots in the DIO to facilitate rapid integration and reduced collision events. As the network stabilizes, we strategically reduce the number of proposed slots to preserve energy.

3.5 Scheduling Packet Update

In the context of TSCH, transmitting packets from higher-layer protocols is not an instantaneous process; rather, it entails a meticulous queuing and scheduling mechanism. As depicted in Fig. 3.2, incoming packets are first enqueued within the TSCH queue and await their transmissions. Since we inject the proposed time slots in DIO and DAO packets, they will be locked during time $T_q = T_{\text{de}} - T_{\text{en}}$, where T_{en} is the instant at which the packet entered the TSCH queue, and T_{de} is the time at which it left the queue. To avoid locking the proposed time slots for a long time T_q, we propose to inject them within the packet only before leaving the queue, i.e., at instant T_{de}, hence $T_q \approx 0$. By carefully orchestrating the RPL control packets' update, through adding details about the schedule state (i.e., the free and reserved slot IDs), we optimize time slot utilization.

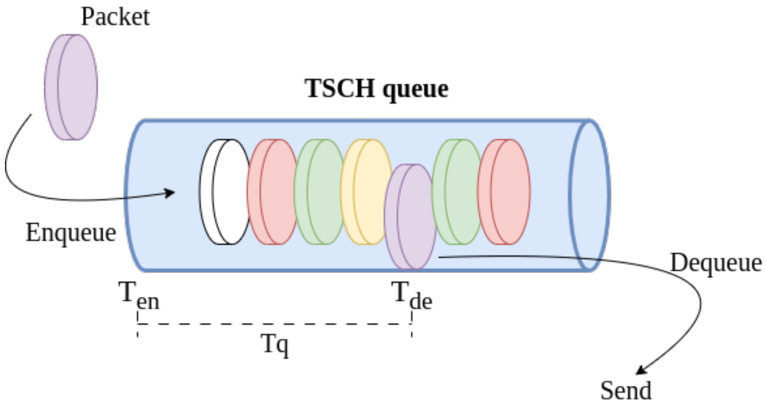


FIGURE 3.2: Illustration of a packet time in a 6TiSCH queue.

3.6 Avoiding TSCH Queue Overflow

In 6TiSCH networks, effective synchronization between SF and TSCH queue management is essential for network operation. With various protocols filling the queue with packets, timely processing during the SF-designated transmission cells is crucial. A mismatch between the queue and cell allocation can lead to packet loss. To address this issue, we propose the introduction of an early cell reservation technique. Specifically, as the TSCH queue begins to be filled and before reaching its limit, our system proactively initiates 6P transactions with the preferred parent for an additional cell reservation. This proactive approach offers several benefits: 1) *Prevent packet drops:* Early reservation of additional cells reduces the likelihood of packet loss due to queue overflow, and 2) *Mitigate resource urgency:* This method allows nodes to secure necessary resources in advance, thus avoiding the last-minute pressure of queue overflow and network instability.

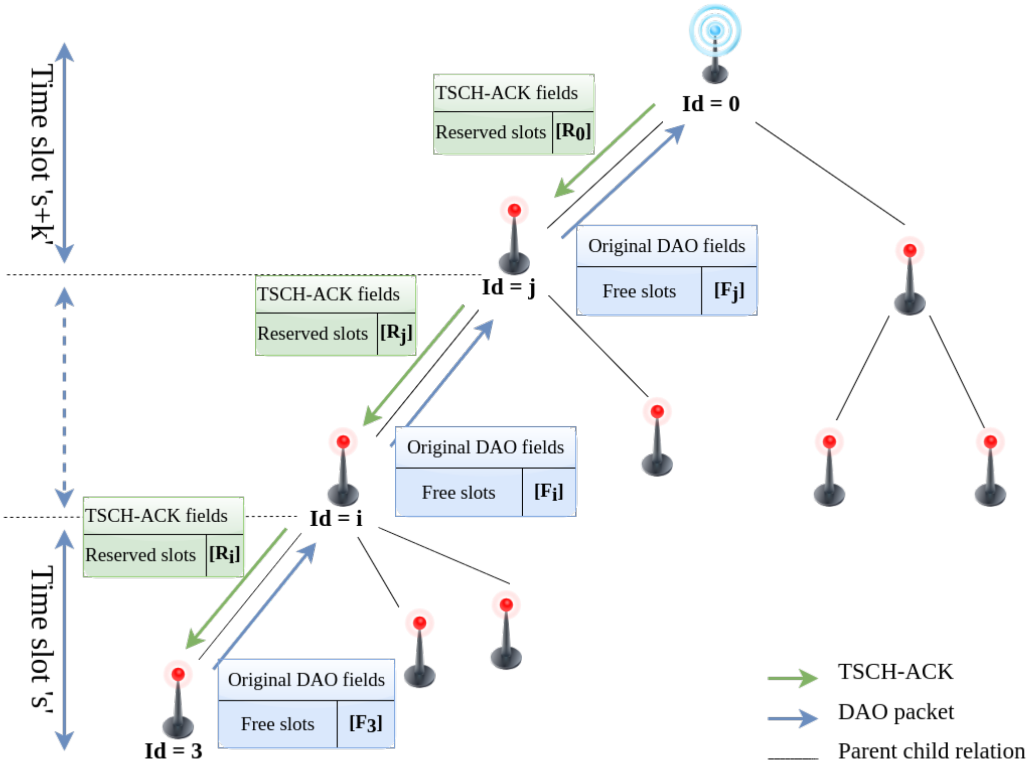


FIGURE 3.3: Illustration of the DAO packet transmission and TSCH ACK utilization.

3.7 Adaptive Self-Sufficient Cell Reservation

In a typical 6TiSCH network, all DAO packets are destined towards the DAG root, thus navigating in a multi-hop fashion. In our proposal, essential information about cell reservation, which consists of the available slots and proposed slot lists, is embedded within these DAO packets. When a parent node receives a DAO packet, as it is the principal conduit to the DAG root, it reserves new cells for its new child where the number of cells should not exceed the maximum number of cells to allocate at once. Then, it loads the list of its free slots (instead of the old one that belongs to the child) and the number of slots to be selected (slots to be allocated to its new child) into the DAO packet. This packet is then seamlessly forwarded, with each intermediate node towards the DAG root executing the same procedure. As shown in Fig. 3.3, upon reception of a DAO packet from node 3, node i undertakes cell selection for reservation and embeds the list of chosen slots into the TSCH ACK packet. It then forwards the enhanced packet to its parent.

To reduce latency at every hop, we propose that the receiver of a DAO packet selects the nearest time slot to the actual one used for the reception. One way to do it is to rely on the concept of the Low-Latency SF (LLSF) protocol [40] where the slots are selected according to the gap to the left of the actual slot. However, we propose here to select the nearest slot to the actual one in any direction. Accordingly, if the node is joining the DODAG, the actual slot will be in the list of proposed slots, thus maximizing its chance of selection. If the node has already joined the DODAG, then the actual slot is the slot that is already allocated to its parent. Consequently, both energy and latency tied to time slot reservations can be reduced.

3.8 Conclusions

In this chapter, we presented a novel approach that optimizes the interaction between SF and RPL to improve the performance of 6TiSCH networks. By integrating several proposed changes into the exchanged control messages and operations of SF and RPL, our solution enhances key performance metrics, including node joining time, packet traffic (both transmitted and received), end-to-end latency, jitter, and node lifetime, all while maintaining energy consumption comparable to that of the MSF benchmark. This work offers valuable insights into enhancing 6TiSCH operations, with implications for the development of IoT networks for 5G and beyond.

Chapter 4

Strategic Approaches to Control Packet Optimization

4.1 Introduction

In this chapter we propose enhancements to the 6TiSCH protocol, focusing on optimizing RPL and 6P operations. We introduce an adaptive cell reservation mechanism for parent switching and node joining, which improves efficiency and reduces energy consumption. Additionally, we address packet size constraints by using binary representations for available time slots and implementing autonomous channel selection, thereby minimizing communication overhead and improving overall network performance in low-power, lossy environments.

4.2 Adaptive Self-Sufficient Cell Reservation

In conventional 6TiSCH operations, when a node switches its parent, it sends a DAO (Destination Advertisement Object) packet to the DAG-root. This packet traverses multiple hops through intermediate nodes before reaching the root, which in turn issues a DAO-ACK packet. The DAO-ACK follows the same multi-hop path back to the originating node, completing the parent switching process. Additionally, the cell reservation process is triggered, requiring a 6P transaction packet to request the allocation of time slots from the new parent [37].

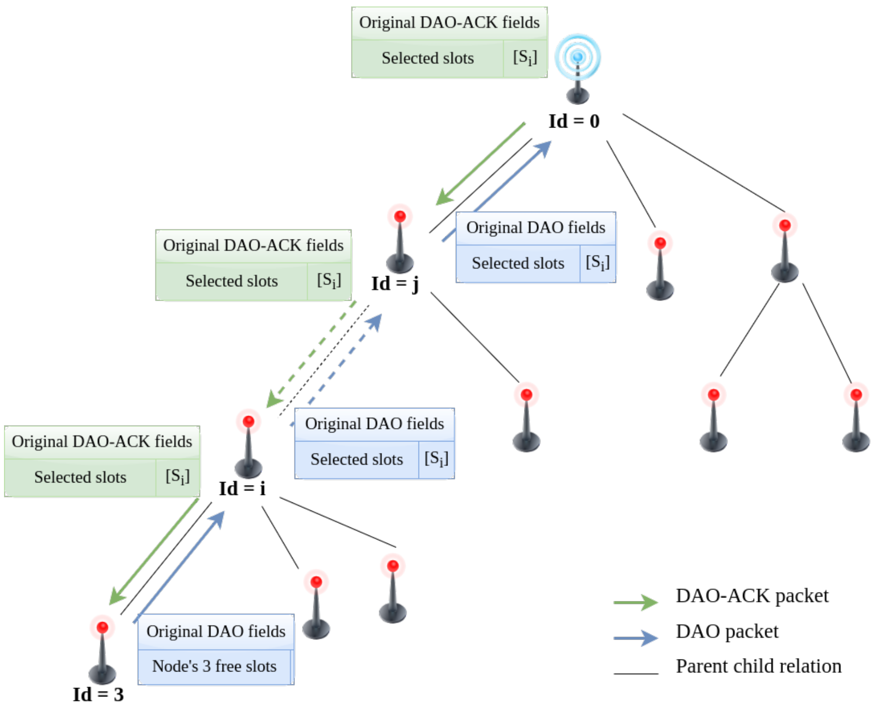


FIGURE 4.1: DAO and DAO-ACK packets exchange.

To improve this procedure, we propose an enhancement where the node initiating the DAO transmission includes a list of its available time slots and the number of slots it intends to reserve for the transmission. The parent node, being the first hop towards the DAG-root, locks the necessary slots based on this list and its own available slots. It then removes the original list of free slots from the DAO packet and replaces it with its own selected slots for the next hop, as illustrated in Fig. 4.1. Specifically, when node i receives a DAO packet from node 3, it selects the required time slots for reservation and embeds the list of chosen slots into the TSCH ACK packet. It then forwards the enhanced packet to its parent. The detailed procedure for handling the reception and processing of DAO packets is outlined in Algorithm 2.

Upon receiving the DAO packet, the DAG-root embeds the list of selected slots by the parent node into the DAO-ACK packet and forwards it to the DAO source, as shown in Fig. 4.1. The parent, before forwarding the DAO-ACK to its child (the DAO sender), reserves the allocated cells. Finally, upon receiving the DAO-ACK, the original DAO initiator node proceeds to reserve the time slots as designated by its parent. The specific actions for receiving the DAO-ACK packet are described in Algorithm 3.

Algorithm 2: Proposed Algorithm for DAO Reception.

Data: DAO packet:
DAO_sender: DAO packet source.
parent: Parent of "*DAO_sender*".
nb_Slot: Number of slots to allocate.
free_slots : "*DAO_sender*" free slots.
Result: Updated DAO packet
locked_slots: Selected slots.

```
if parent = my_adresse then
    my_free_slots ← Receiver's free slots.
    shared_slots ← my_free_slots ∩ free_slots
    selected_slots ← Nearest "nb_Slot" slots to the first slot in shared_slots.
    for slot_ID in selected_slots do
        Lock slot_ID in locked_slots.
    Delete free_slots from the packet.
    if "I am DAG-root" then
        Create DAO_ACK packet.
        Append selected_slots to DAO_ACK packet.
        Send DAO_ACK packet.
    else
        Append selected_slots to DAO packet.
        Forward DAO packet.
else
    if "I am DAG-root" then
        Create DAO_ACK packet.
        Append selected_slots to DAO_ACK packet.
        Send DAO_ACK packet.
    else
        Forward DAO packet.
```

To ensure the efficient reservation of time slots, the priority of DAO and DAO-ACK packets is set equal to that of 6P packets. This synchronization of slot reservation improves resource allocation, optimizing network communication by minimizing delays and reducing energy consumption.

4.2.1 Latency minimization

To minimize latency at each hop, we propose that the receiver of the DAO packet selects the nearest available slot from the list of the free slots of the joining node. This mechanism is inspired by the Low-Latency SF (LLSF) protocol, where slots are selected based on their proximity to the first free slot [40]. This approach ensures reduced latency and energy consumption during the parent switching operation, which enhances the efficiency of the overall network.

Algorithm 3: Proposed Algorithm for DAO-ACK Reception.

Data: DAO_ACK packet:
selected_slots: Parent's selected slots.
parent: Parent of DAO packet source.
child: DAO packet source.

```
if parent = my_adresse then
    channel_ID ← Reception_channel(my_adresse).
    for slot_ID in selected_slots do
        Unlock slot_ID.
        Allocate the cell(channel_ID, slot_ID, Rx).
        Forward the packet.
else
    if child = my_adresse then
        channel_ID ← Reception_channel(parent).
        for slot_ID in selected_slots do
            Allocate the cell (channel_ID, slot_ID, Tx).
    else
        Forward the packet.
```

4.3 Minimizing Overhead in RPL and 6P Packets

In low-power, lossy networks like those operating with the 6TiSCH architecture, the size of the packets and the energy consumption associated with packet transmission are crucial constraints. To address this, we propose an approach that minimizes the overhead of RPL and 6P packets by efficiently utilizing the available space and reducing the amount of data that needs to be transmitted. Specifically, we introduce a binary representation for the list of available time slots, which allows us to achieve significant reductions in the size of the data being transmitted while maintaining the necessary information.

4.3.1 Binary Representation of Available Time Slots

To reduce the transmission overhead, we use a binary representation for the available time slots in the slot frame. For example, in a slot frame with 100 time slots, we represent the state of each slot using a 100-bit vector. Each bit, indexed from $i = 1$ to $i = 100$, corresponds to the state of the i^{th} time slot: a value of "0" indicates the slot is available, and "1" indicates the slot is occupied. This compact representation significantly reduces the size of the data included in the RPL and 6P packets.

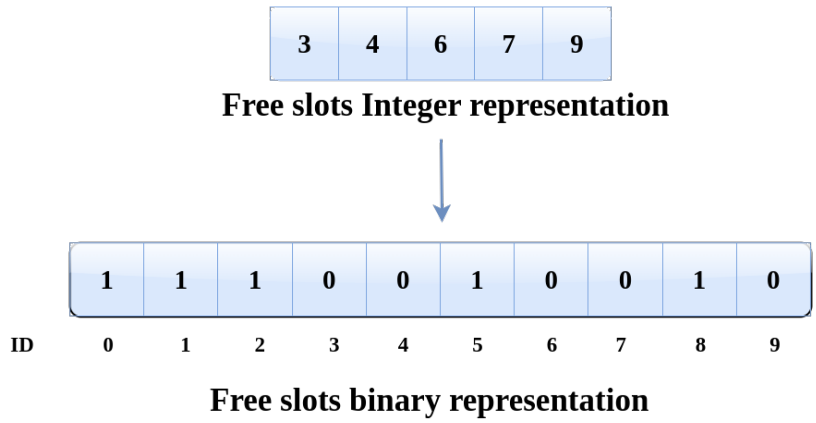


FIGURE 4.2: Integer vs. binary representation of available time slots.

4.3.2 Autonomous Channel Selection

To further optimize resource usage, we propose an autonomous receiver-based mechanism for channel selection. In this mechanism, the transmission channel is determined based on the receiver's MAC address, obviating the need for nodes to exchange channel information. The time slot alone is sufficient to establish communication. To ensure channel diversity and maintain network resilience, we adopt the channel selection method used in the MSF protocol, where the transmission/reception channel ($channel$_ID) is computed based on the Absolute Slot Number (ASN) and the channel offset. The channel is determined by the equation:

$$channel\_\mathrm{ID} = (ASN + cell\_\mathrm{channel})\,\%nb\_\mathrm{channels}$$

where nb_{channels} is the total number of channels, and % denotes the modulo operation. This method ensures efficient and reliable communication while minimizing the overhead in packet transmission.

4.4 Conclusion

In this chapter, we introduced a novel approach to enhance the performance of mesh 6TiSCH networks by optimizing the interaction between the RPL and 6P protocols. Specifically, we proposed several improvements in the exchanged control packets, including adaptive cell reservation and reduction of overhead. Through simulations, we demonstrated that our approach outperforms the 6TiSCH baseline, achieving better results in key performance metrics such as the number of control/data packets exchanged, packet delivery ratio, network joining time, power consumption, and node lifetime.

Chapter 5

Simulation & results

5.1 Introduction

In this chapter, we present the results of the simulations conducted to evaluate the performance of our contribution. The simulations aim to assess the effectiveness of various protocol parameters, scheduling functions, and traffic management strategies, as well as their impact on network life time and energy consumption. Using the 6TiSCH Simulator, which is specifically designed for 6TiSCH-based networks, we model different network topologies and traffic patterns to simulate real-world scenarios. Some of the results and figures presented in this chapter have been previously published in [46] and [47]. The content has been adapted and expanded here to provide a more comprehensive analysis in the context of this thesis.

The chapter begins by introducing the 6TiSCH simulator then outlining the simulation setup, including the selection of performance metrics, and the simulation parameters. We then present the results of several key experiments, followed by an analysis of these results in relation to the research objectives. Finally, the chapter concludes with a discussion of the implications of these findings, highlighting how they contribute to a deeper understanding of the 6TiSCH protocol's performance and its potential for real-world deployment.

5.2 6TiSCH Simulator

To thoroughly evaluate the efficiency and robustness of the proposed scheme, we utilized the 6TiSCH simulator [3]. It is an open-source simulation tool that models and evaluates the performance of Time-Slotted Channel Hopping (TSCH) in low-power wireless networks, especially for Industrial Internet of Things (IIoT) applications. This simulator offers a realistic and flexible environment for analyzing key network performance metrics, such as packet transmission success rate, queue dynamics, congestion behavior, energy consumption, and parent switching efficiency [41]. Moreover, it has been widely adopted in prior research to validate scheduling and routing mechanisms based on the 6TiSCH architecture [41–44].

The 6TiSCH simulator adopts an energy model that derives current-consumption values for each slot type (Tx, Rx, Idle, Sleep) from experimentally measured hardware platforms such as the CC2538 and CC1200, and integrates them over time to calculate cumulative charge or energy consumption [45]. This modeling approach provides a realistic estimation of node energy behavior in TSCH networks under diverse and dynamic operating conditions.

Simulator	**Learning Curve**	**Scalability**	**6TiSCH Implementation**	**Standard Compliant**
ns-3	High	Medium	None	N/A
OMNet++	High	Medium	None	N/A
TOSSIM	Medium	High	None	N/A
Cooja (emulator)	High	Low	Yes (Partial)	Partially
OpenSim (emulator)	High	Low	Yes	Yes (byte-accurate)
6TiSCH Simulator	Low	High	Yes	Yes (behavioral)

TABLE 5.1: Comparison between the 6TiSCH Simulator and the different network simulator alternatives (source [3]).

As highlighted in Table 1, the 6TiSCH Simulator is particularly well-suited for our study due to several key factors. Firstly, it provides a specific implementation of the 6TiSCH protocol, fully aligned with the requirements set forth in the 6TiSCH RFCs and drafts. This is a significant advantage over other simulators like Cooja, which does not support the Minimal Schedule Function (MSF). Secondly, the 6TiSCH Simulator offers excellent scalability, enabling simulations with a large number of nodes, up to several hundred, making it ideal for our large-scale network scenarios. Unlike general-purpose simulators such as NS-3 or OMNet++, which are designed to support a broad array of protocols and models, the 6TiSCH Simulator is a highly specialized tool tailored for rapid prototyping and testing of 6TiSCH-specific scenarios, making it the most appropriate choice for this research.

Although the 6Tisch simulator offers detailed insights into TSCH-based networks, it lacks the ability to simulate complex mobility patterns, which may limit the accuracy of performance under highly dynamic conditions.

5.3 Performance Evaluation of Packet Balancing Method

5.3.1 Simulation Configuration

In order to evaluate the effectiveness and resilience of the proposed scheme, we utilized the 6TiSCH simulator [3]. The generated plots encompass outlier results, ensuring a comprehensive representation of the observed performance across various scenarios. For completeness, the generated plots include outlier values to accurately represent the entire spectrum of observed simulation results. The simulations were conducted over two distinct durations: short-term of 30 minutes and long-term of 270 minutes (4.5 hours), with each configuration being repeated 100 times across the following network setups:

- *Conventional 6TiSCH Configuration:* This configuration utilizes the conventional Minimal Schedule Function (MSF) for scheduling and the OF0 Objective Function (OF) for the RPL routing protocol [18, 38], denoted as "MSF" in the results.
- *Proposed Solution-based Configuration:* This setup applies our proposed solution, which includes a modified MSF along with the OF0 function. This configuration is labeled as "PB" in the results.

The network configuration consists of $N_1 = 50$ (and $N_2 = 100$) nodes placed randomly within a $1\,\text{km}^2$ area. The x and y coordinates of the nodes are randomly generated. The Pister-hack connectivity model [3] is employed, ensuring that each node maintains at least three links to its neighbors. During the simulation, the observed routing tree depths varied between 2 and 3 hops, with a median depth of 2 hops for the $N_1 = 50$ node network and between 2 and 3 hops for the $N_2 = 100$ node network. We assumed that data transmission frequencies ranged from $\{5, 15, 45, 60\}$ seconds.

Table 5.4 presents the simulation parameters, while Table 5.3 lists the specific settings for the proposed solution.

The choice of the “TSCH queue empty places threshold for cell reservation” is based on the following reasoning: each time a packet is enqueued, the node evaluates the remaining capacity in the TSCH queue. When the available space reaches the predefined threshold, the node triggers a cell reservation request using the 6P protocol. A specified number of cells (as shown in Table 5.3) can be requested during this process. Further requests are constrained by the parameter ”Cycle interval before next cell request”. This threshold is set to 2 empty places in the queue, allowing space for the 6P transaction packet and for reserving capacity for future packets. Such a configuration prevents queue overflow and improves the network’s ability to handle data packets, particularly in high-traffic scenarios.

Table 5.2: Simulation Parameters

Parameter	**Value**	
Number of nodes	50, 100	
Number of simulation runs	100	
Simulation duration (min)	30, 270	
Network area (km^2)	1	
Service packet period (sec)	5, 15, 45, 60	
Slotframe length (in number of slots)	100	
Time slot duration (sec)	0.01	
Secure Joining required	True	
TSCH maximum payload length (bytes)	120	
TSCH queue size (number of packets)	10	
Number of radio channels	16	
	MSF	PB
DIO packet size (bytes)	76	120
DAO packet size (bytes)	20	75

To determine the optimal number of proposed slots per DIO, we conducted experiments with various configurations. Figures 5.1a and 5.1b illustrate the impact of varying the number of proposed slots on the joining time and current consumption (indicating energy consumption). These tests were carried out over a 30-minute observation period, using a service packet period of 5 seconds, and for network sizes of $N_1 = 50$ and $N_2 = 100$ nodes. As seen in Fig. 5.1a, increasing the number of proposed slots reduces the joining time for any number of nodes. This is expected, as more proposed slots provide greater opportunities for a new node to

Table 5.3: Parameters of the Proposed Solution

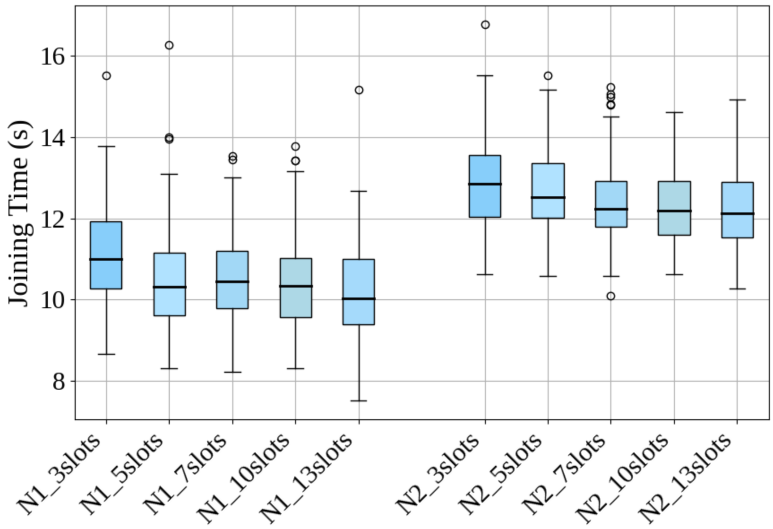
Parameter	**Value**
Number of proposed slots per DIO	7
DIO cells duration (in number of slotframes)	10
Number of permanent DIO slots	1
Slot selection ratio within DIO slots	3
Maximum number of cells to reserve at once	5
Initial phase duration (min)	45
TSCH queue empty places threshold for cell reservation	2
Cycle interval before next cell request	2
Number of cells reserved upon reservation process	1

quickly join the DODAG. However, since the number of proposed slots cannot be excessively high due to packet size constraints, we find that a value of 7 proposed slots yields performance comparable to 13 slots, especially in large networks (N_2). Similarly, in Fig. 5.1b, a value of 7 proposed slots minimizes power consumption, as higher values, such as 13, may result in inefficient idle listening. Hence, we recommend setting the number of proposed slots per DIO to 7 for optimal performance in terms of joining time and power consumption.

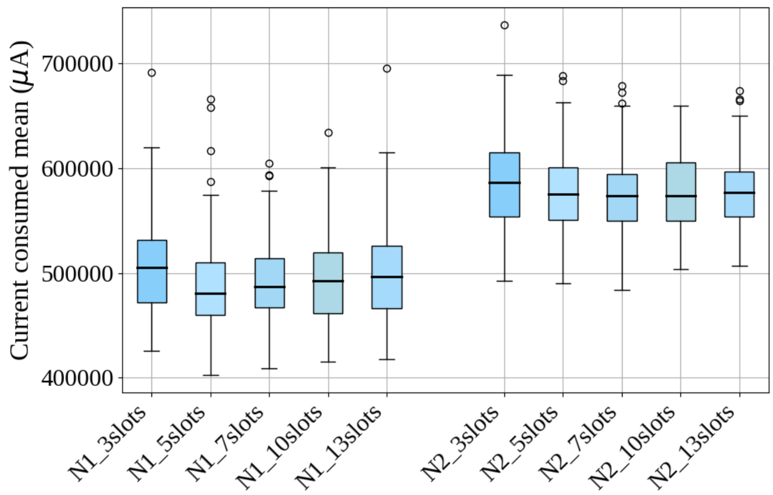
5.3.2 Short-Term Results

The initial phase of a network's operation, known as 'network formation', is critical for establishing the network's parameters and topology. This phase sets the foundation for network functionality, but it also entails considerable energy expenditure. To assess the effectiveness of our solution in accelerating the network formation process and minimizing power consumption [39], we conducted simulations and present the results in Figs. 5.2 and 5.3. These figures were generated under the same conditions as the previous simulations, while varying the service packet period. Additionally, we compare the performance of our proposed method, denoted as PB, with that of the benchmark, MSF.

As depicted in Fig. 5.2, the PB solution consistently achieves lower joining times for any number of nodes (N) and service packet period. For instance, with $N =$ 50 and a service packet period of 15 seconds, PB achieves an average joining time of 11 seconds, while MSF requires approximately 12 seconds under the same



(A) Joining time (in sec)



(B) Consumed current (in μA)

FIGURE 5.1: Joining time and consumed current vs. number of proposed slots per DIO.

conditions. This performance difference is attributed to the way the PB scheme operates: a node within PB uses predefined slots from the received DIO to define the transmission cells for sending the DAO, whereas MSF involves sending a 6P request to its preferred parent and waiting for a response to reserve cells. In PB, the DAO packet serves a dual purpose: establishing the DODAG structure with RPL and participating in the SF.

As the service packet period increases, we observe a slight degradation in the joining time. This behavior is expected, as nodes with a shorter service packet period transmit EB packets more frequently, thus increasing the probability of synchronization with other nodes and enabling faster DODAG joining. Conversely,

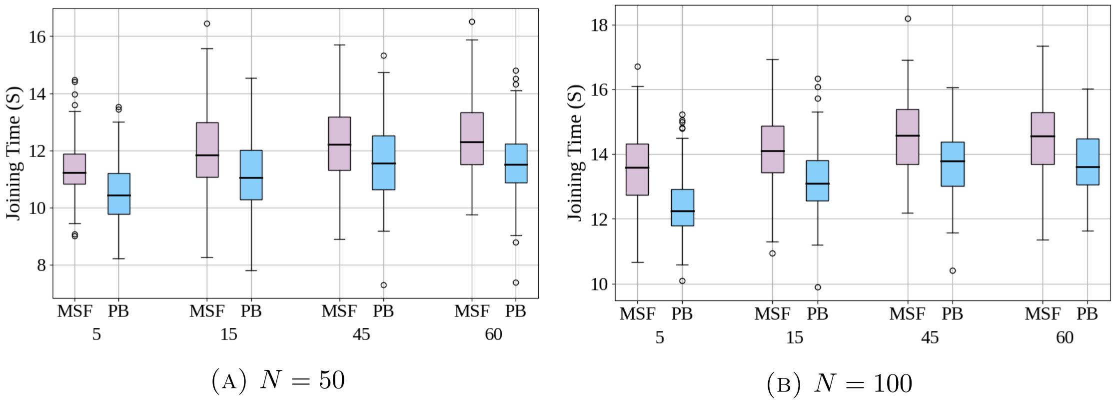


FIGURE 5.2: Joining time (in sec) vs. length of service packet period (short term).

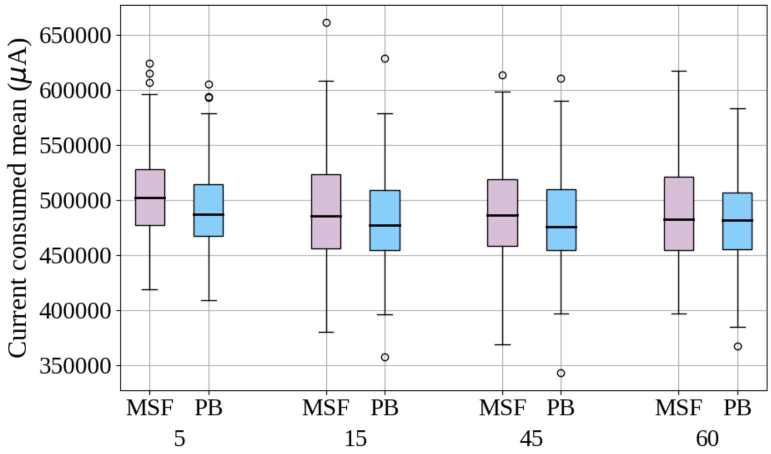
a longer service packet period reduces the likelihood of synchronization, leading to a longer joining time.

The network formation phase in 6TiSCH networks often incurs significant energy consumption before the transmission of useful data. Fig. 5.3 presents the current consumption during network formation for different scenarios, as described in Fig. 5.2. The results clearly show that PB outperforms MSF in terms of energy consumption for all values of N and service packet period. As N increases, the performance gap between PB and MSF becomes more pronounced, further demonstrating the robustness of our solution in dense network environments.

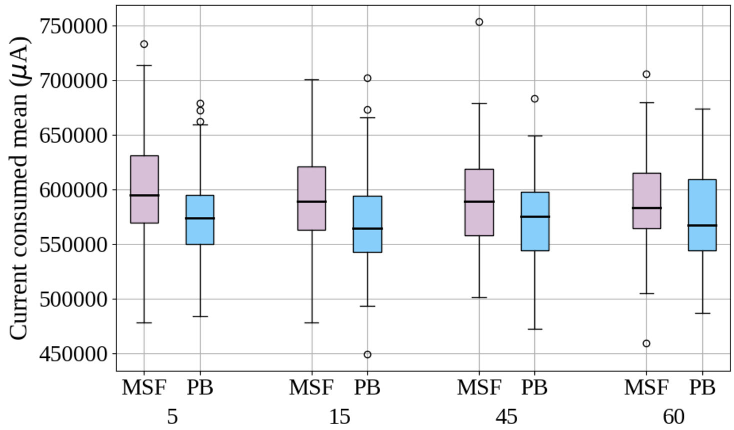
Finally, it is worth noting that the "service packet period" has a minimal effect on the overall consumed current. This indicates that, regardless of the service packet period, the PB scheme remains efficient in energy consumption.

5.3.3 Long-term Performance Evaluation

In this section, we assess the performance of PB and MSF for long-term operation in terms of key metrics such as the number of transmitted/received packets, end-to-end (E2E) latency, network jitter, energy consumption, and node lifetime. The evaluation focuses on the system's efficiency and stability over time.



(A) $N = 50$



(B) $N = 100$

FIGURE 5.3: Consumed current (in microampere) vs. length of service packet period (in sec) (short term).

5.3.3.1 Transmission and Reception of Packets

Fig.5.4 illustrates the number of transmitted (Tx) and received (Rx) packets for varying service packet periods and different number of nodes in the network (50, 100) N. For any N and service packet period, our approach outperforms the MSF benchmark. Specifically, networks using the PB scheme achieve more frequent transmissions, and nodes successfully receive a greater number of packets. As the service packet period increases (i.e., when a longer service is running), the number of exchanged packets decreases. This result is expected, as a longer service period reduces the frequency of packet transmission during the 4.5-hour observation period. Moreover, when $N = 100$, the number of exchanged packets almost doubles compared to the scenario where $N = 50$. The availability of more nodes within the DODAG allows for the transmission and reception of more packets, within the constraints of the 6TiSCH available channels and time slots.

5.3.3.2 End-to-End Latency

Fig. 5.5 presents the network's E2E latency for different service packet periods and N. E2E latency is defined as the time elapsed from the moment a source node transmits a service/application packet until the packet is successfully received and acknowledged by the destination node. Our method, PB, consistently achieves lower E2E latency compared to MSF under all network conditions. As the service packet period increases, the E2E latency follows a logarithmic trend and eventually saturates at higher values. For example, for $N = 100$, the average E2E latency stabilizes around 2.1 seconds for service packet periods of 45 and 60 seconds.

5.3.3.3 Network Jitter

In Fig. 5.6, we display the median and average jitter for transmitted packets, for varying N and service packet periods over the long-term observation period. Jitter refers to the variation in packet arrival times within the network, which is typically caused by factors such as congestion, route changes, or other network inconsistencies. The median represents the typical experience of a node, excluding outliers, while the average jitter reflects the overall network performance over time. According to the results, MSF exhibits a high initial peak in jitter (both median and average), primarily during the network formation phase, which can negatively impact real-time services. In contrast, PB consistently maintains lower jitter levels, indicating a more stable packet delivery time. The median jitter of MSF shows significant variability, making it unsuitable for real-time applications that require consistent data delivery. In comparison, PB demonstrates more predictable jitter performance. As N increases from 50 to 100, jitter performance remains stable and does not degrade, particularly for the PB scheme.

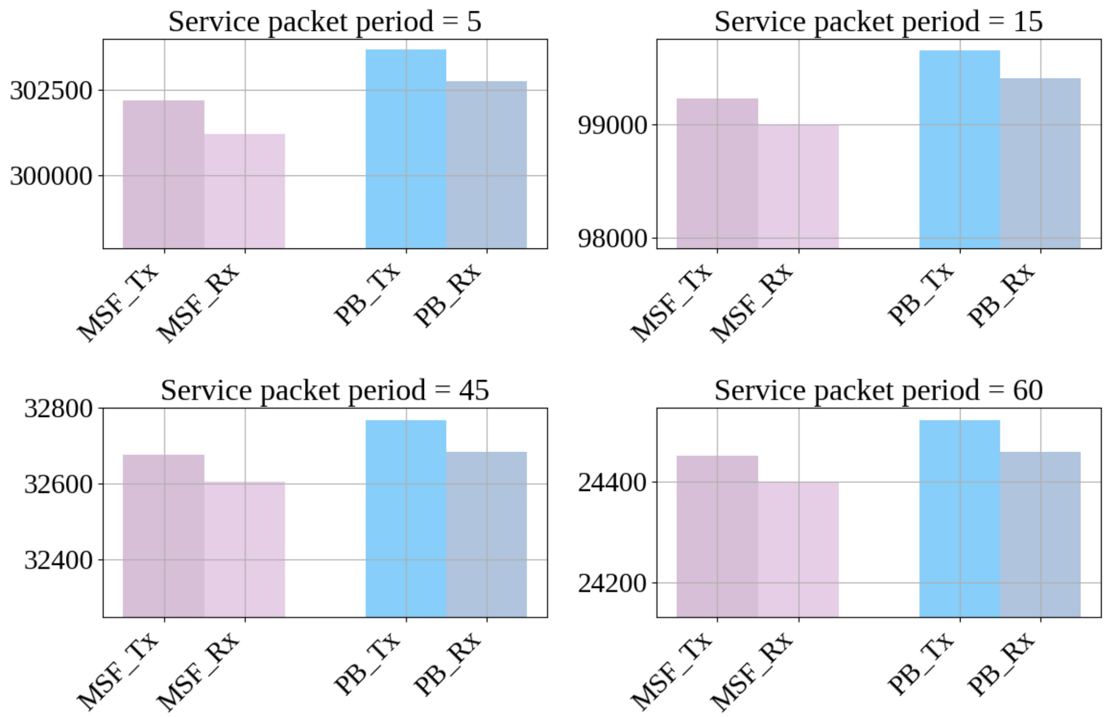
5.3.3.4 Energy Consumption

Fig. 5.7 presents the Cumulative Distribution Function (CDF) of charge consumption (in Coulomb) for the PB (blue) and MSF (purple) schemes, for various

N values and service packet periods. The energy consumption profiles of PB and MSF are nearly identical across all scenarios, indicating that both methods achieve similar energy efficiency. This suggests that our solution conserves energy while significantly improving other performance metrics, such as packet joining time, E2E latency, and jitter. Notably, for a given N, the charge consumption does not change significantly with increasing service packet periods. Additionally, the CDFs for $N = 50$ and $N = 100$ at fixed service packet periods are very similar, implying that N has minimal impact on the consumed charge per node.

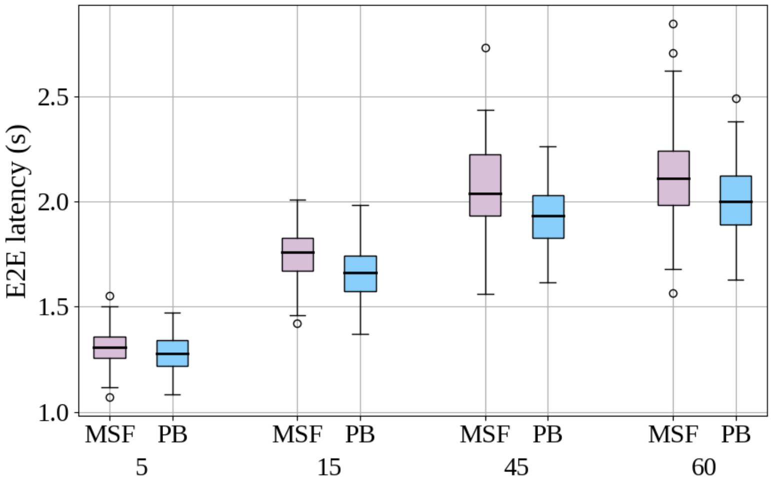
Finally, in Fig. 5.8, we examine the CDF of node lifetime for both PB (blue) and MSF (purple) solutions, considering different values of N and service packet periods. For all scenarios, nodes in the PB-based network have slightly shorter lifetimes compared to those in the MSF-based network. For instance, when $N = 50$ and the service packet period is set to 45 seconds, approximately 67% of nodes using PB will have a lifetime of less than 5 years, compared to 58% for MSF. As N increases, the gap in node lifetime between PB and MSF decreases, which suggests that our method becomes more energy-efficient in denser networks.

(A) $N = 50$

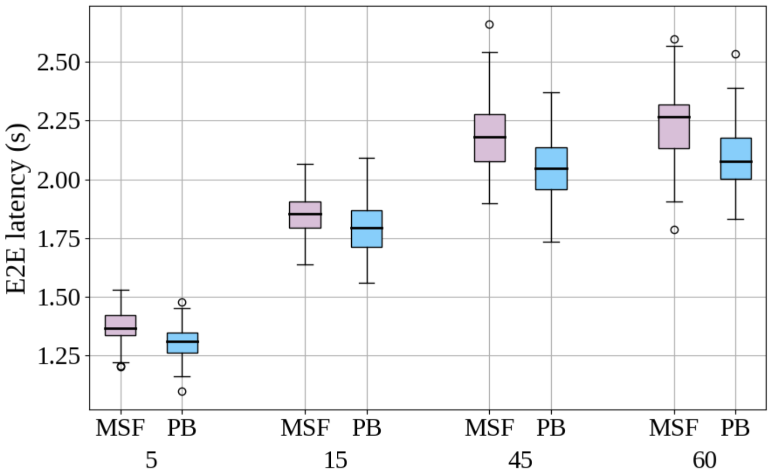


(B) $N = 100$

FIGURE 5.4: Number of transmitted (Tx) and received (Rx) packets (different service packet periods, short term).

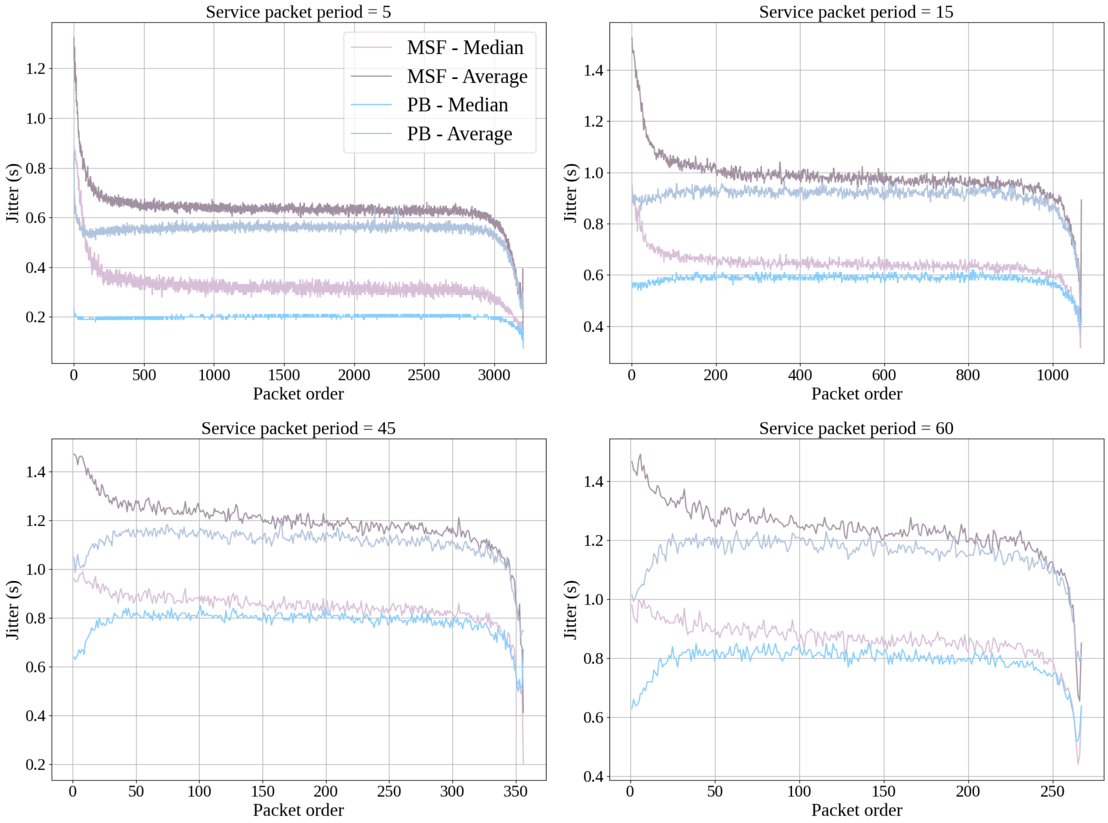


(A) $N = 50$

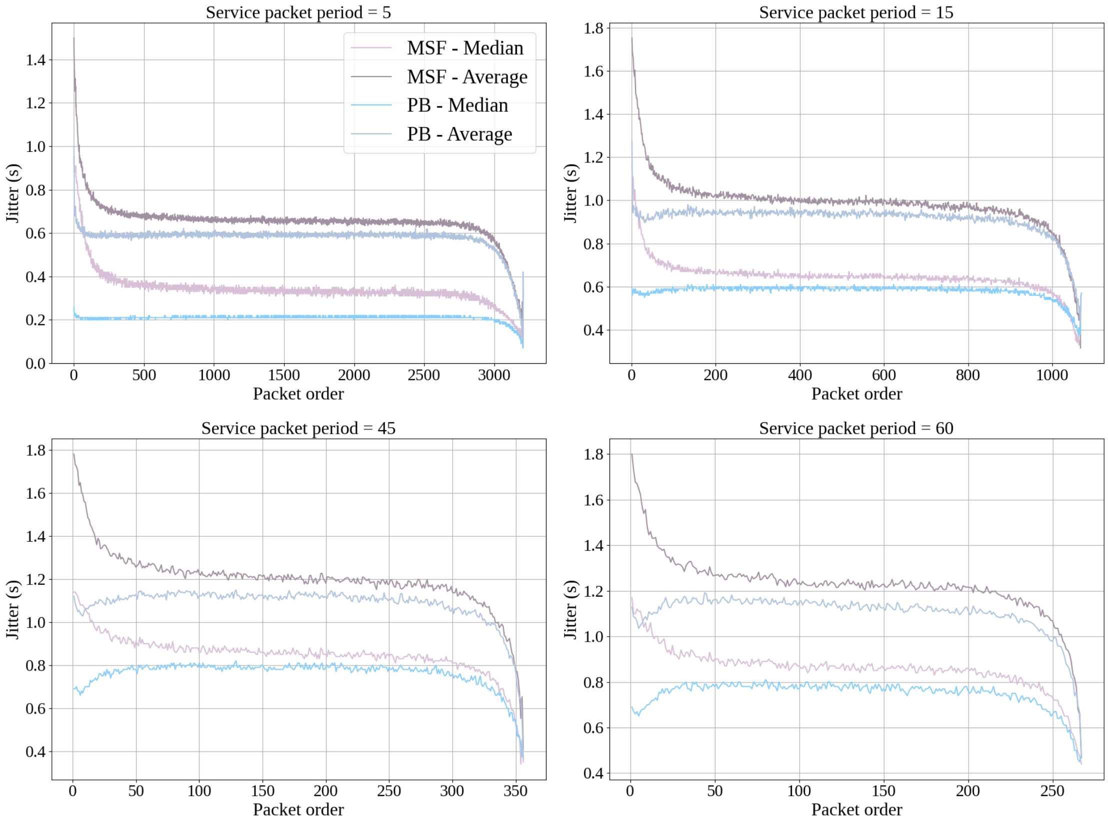


(B) $N = 100$

FIGURE 5.5: E2E latency (in sec) vs. service packet period (long term).

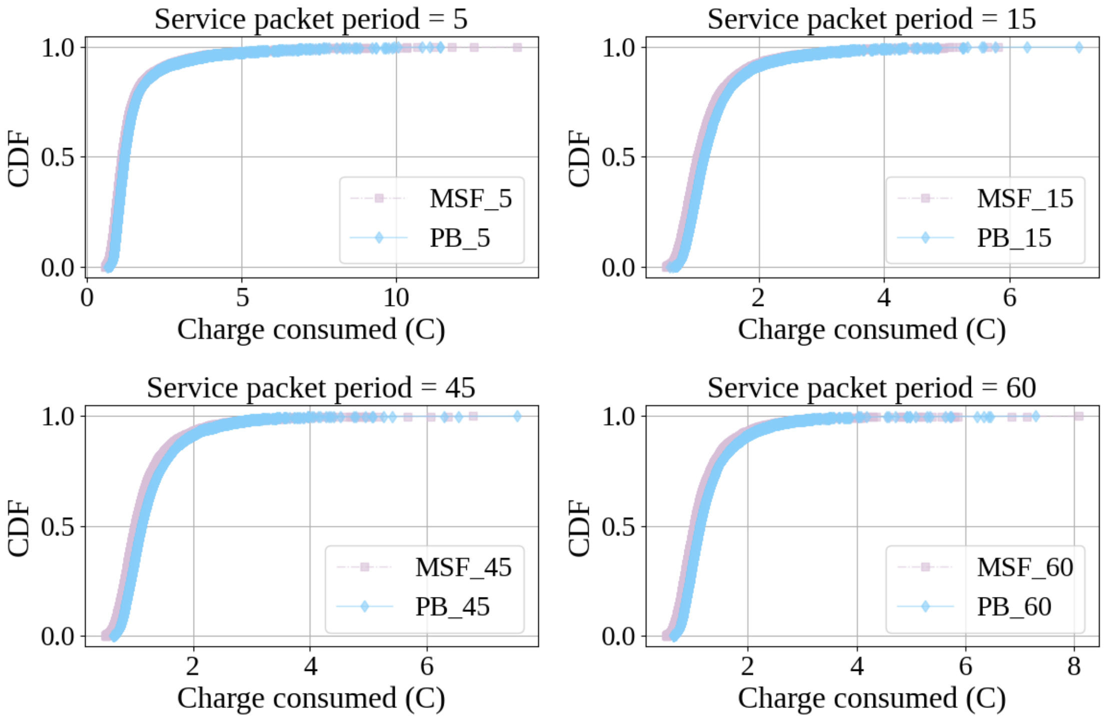


(A) $N = 50$

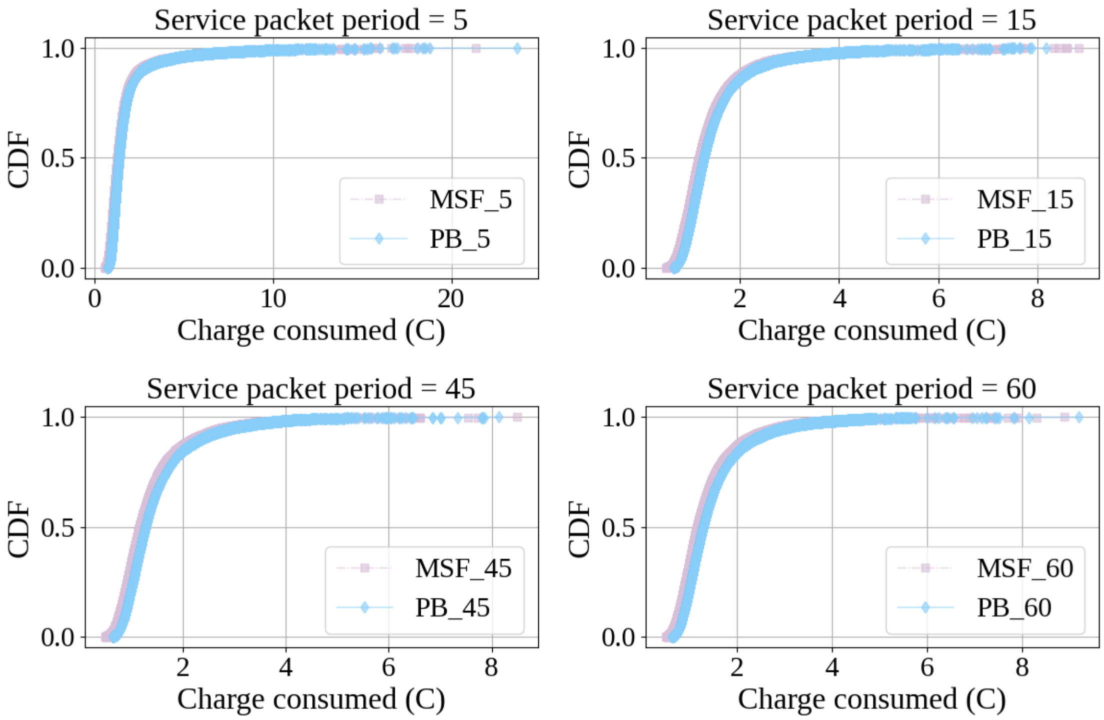


(B) $N = 100$

FIGURE 5.6: Median and average jitter of transmitted packets (different service packet periods, long term).

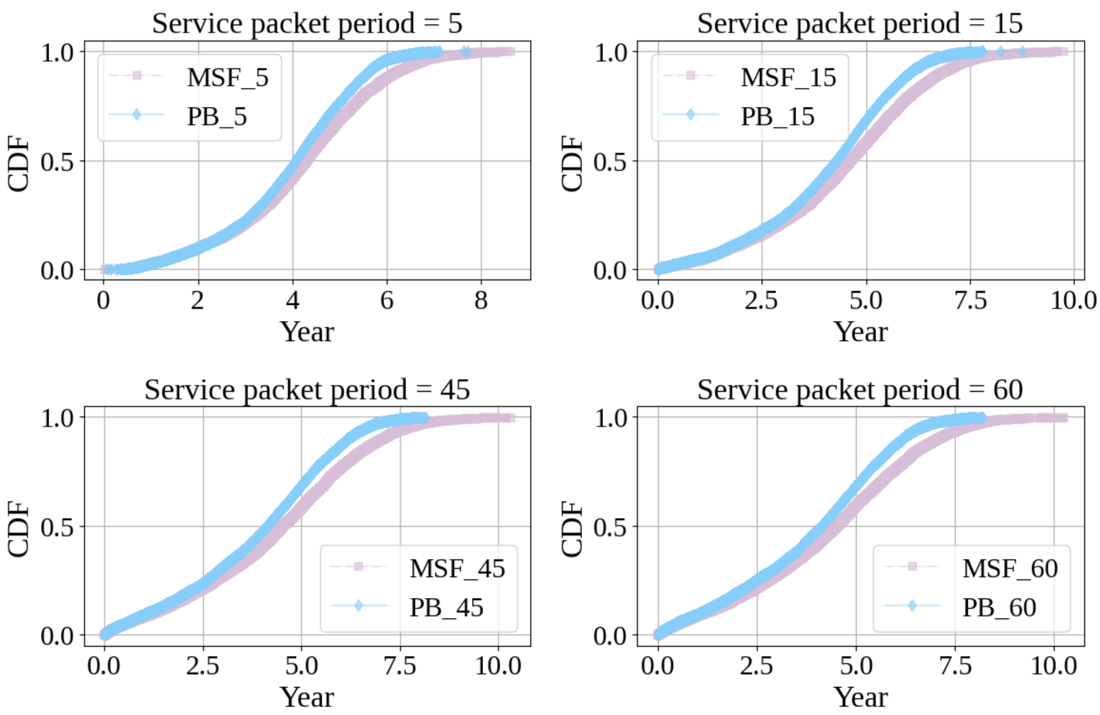


(A) $N = 50$

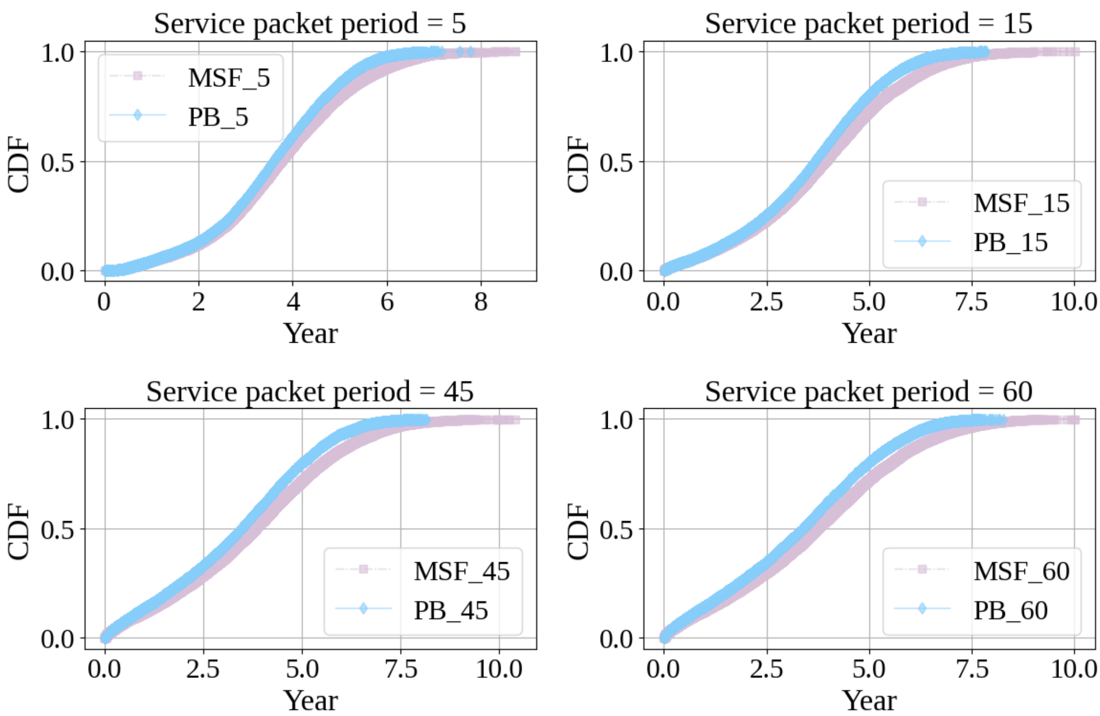


(B) $N = 100$

FIGURE 5.7: CDF of consumed charge.



(A) $N = 50$



(B) $N = 100$

FIGURE 5.8: CDF of nodes' lifetime.

5.4 Performance Evaluation of Packet Optimization method

In this section, we evaluate the performance of our proposed control packet optimization (CR-DAO) method within the 6TiSCH network. We present the results for various key performance indicators (KPIs) such as packet exchange, packet delivery ratio (PDR), latency, joining time, current consumption, and node lifetime, comparing CR-DAO with the standard 6TiSCH protocol.

5.4.1 Simulation parameters

To provide a comprehensive view, the generated plots include outliers, capturing the entire spectrum of results. The simulations were conducted over a duration of 7.5 hours (450 minutes) and were repeated 100 times for two distinct network configurations.

1. The state-of-the-art 6TiSCH, called "Standard 6TiSCH" or "STD-6TiSCH",
2. Our method, named "Cell Reservation DAO" or "CR-DAO".

The remaining of the simulation parameters are detailed in Table. 5.4.

TABLE 5.4: Simulation Parameters

Parameter	**Value**
Number of Mesh Nodes	50
Number of Simulation Runs	100
Simulation Duration	450 minutes
Application Packet Periods	5, 15, 54, 60 seconds
Slot Frame Length	50 slots
Time Slot Duration	0.01 seconds

5.4.2 Control Packet Exchange

As shown in Fig. 5.9, the number of 6P control packets exchanged in the 6TiSCH network is significantly affected by the application period when our CR-DAO

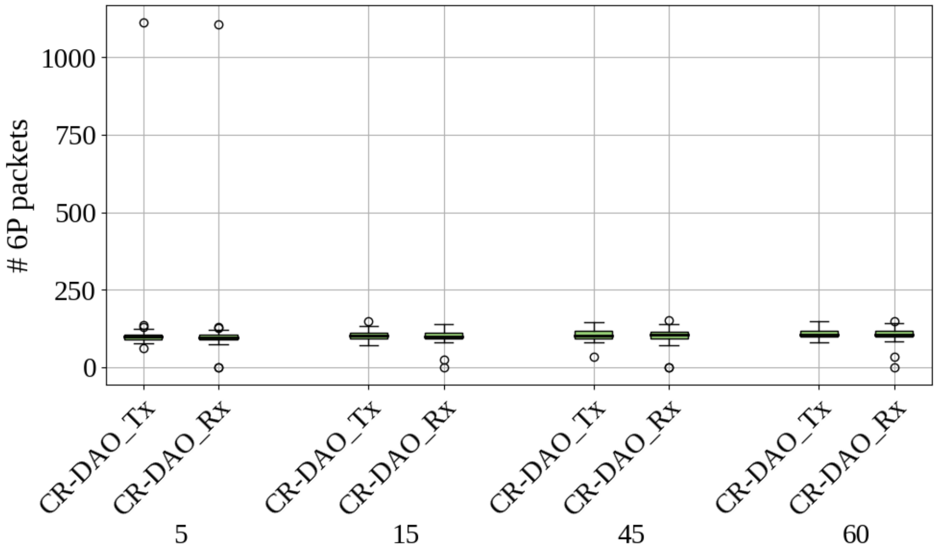


FIGURE 5.9: Total number of Tx/Rx 6P packets (different periods).

method is deployed. For different application periods, the average number of exchanged 6P packets hovers around 100, with a noticeable decrease in the number of outliers as the application period increases. This trend is due to the fact that larger application periods result in fewer control packet exchanges over time. Fewer control packets lead to reduced collisions and less frequent channel allocation/reallocation events, improving network stability and reducing overhead.

5.4.3 Data Packet Transmission and Reception

In Fig. 5.10, we present the number of application data packets transmitted and received for both CR-DAO and the baseline STD-6TiSCH, across various application periods. Our CR-DAO method consistently transmits and receives more packets than the STD-6TiSCH protocol. Moreover, the close alignment between the number of transmitted and received packets underscores the effectiveness of CR-DAO in ensuring reliable communication.

As the application period increases, the number of exchanged packets decreases. This is expected because a larger application period results in a longer time between packet transmissions, thereby reducing the frequency of data exchanges over the observation period.

Further examination of the End-to-End Packet Delivery Ratio (PDR), shown in Fig. 5.11, reveals that CR-DAO achieves nearly 100% PDR for all application

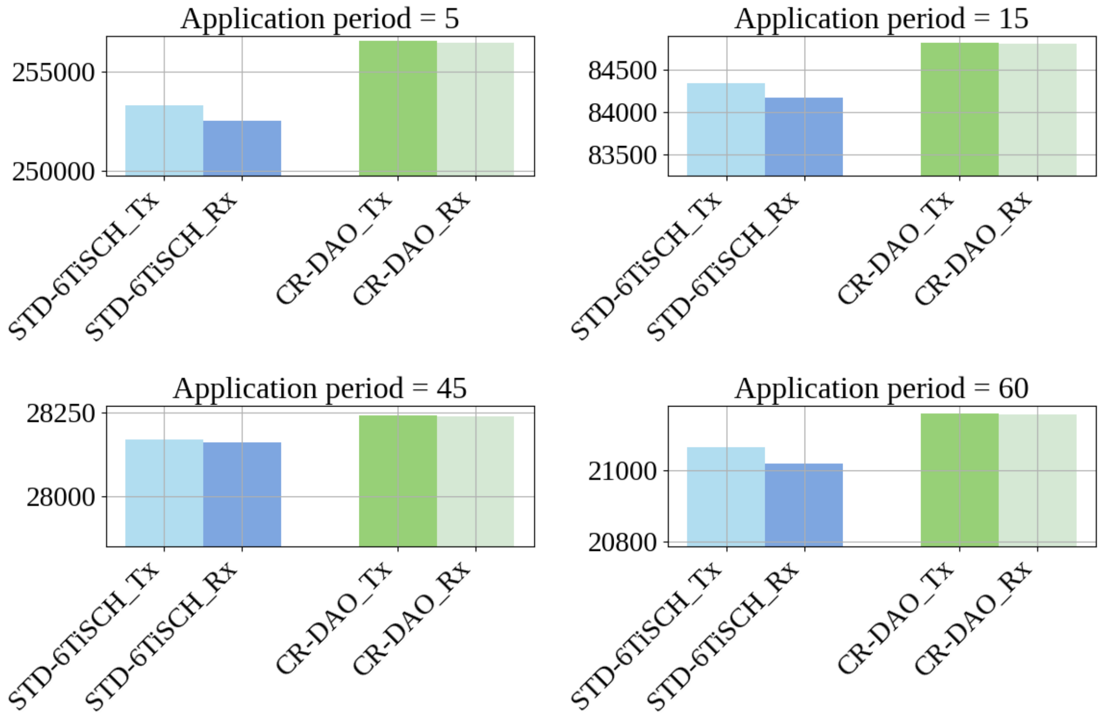


FIGURE 5.10: Total number of Tx/Rx data packets (different periods).

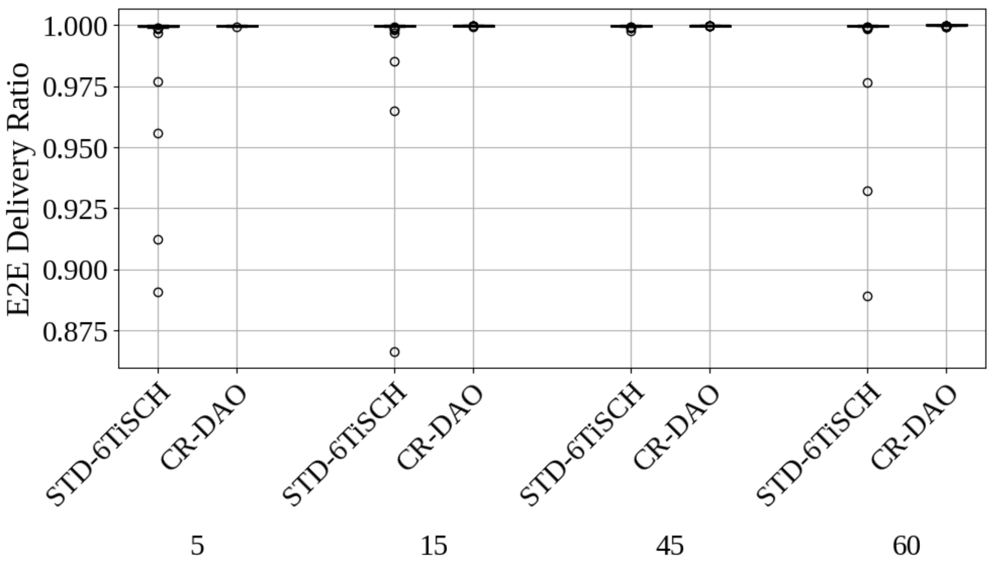


FIGURE 5.11: End-to-End (E2E) packet delivery ratio (different application periods).

periods, outperforming the STD-6TiSCH method, which has a lower PDR, reaching as low as 87%. This demonstrates the superior reliability of CR-DAO, which ensures high delivery success even in challenging network conditions.

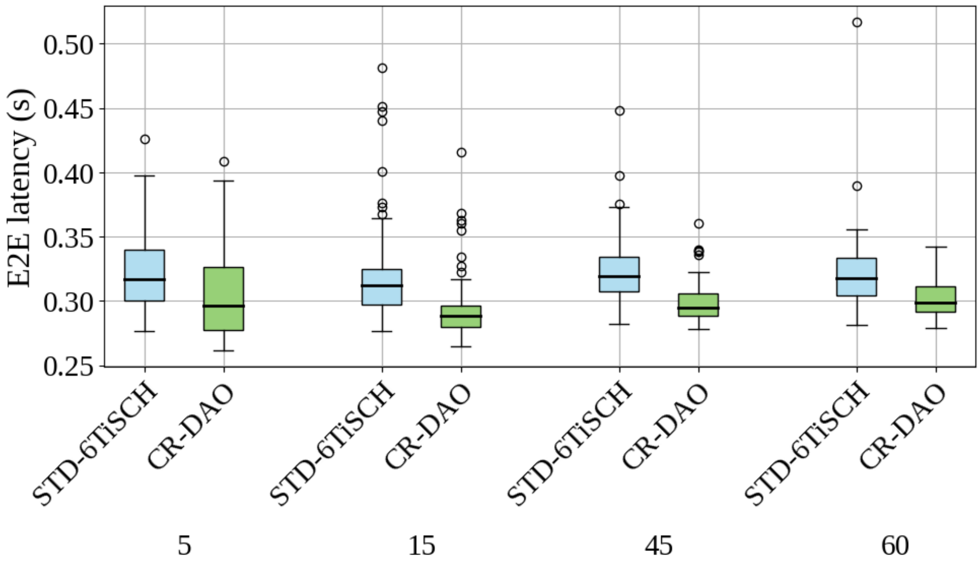


FIGURE 5.12: End-to-End (E2E) delivery latency (different application periods).

5.4.4 Latency Performance

Latency, defined as the time interval between the transmission of an application data packet and its successful reception at the DAG-root, is a crucial metric for evaluating network performance. As shown in Fig. 5.12, CR-DAO consistently achieves lower average latency than the standard STD-6TiSCH method for all application period values. This indicates that our approach provides faster packet delivery, enhancing the network's responsiveness.

Additionally, the latency remains stable across different application periods, highlighting that varying the period does not significantly affect the efficiency of our method in terms of delay. This stability is particularly beneficial for applications requiring real-time data transmission.

5.4.5 Network Joining Time

Fig. 5.13 illustrates the network joining time, which is the duration required for a node to successfully join the network after its activation. Our CR-DAO method outperforms STD-6TiSCH in terms of faster joining times, regardless of the application period. A shorter joining time enables nodes to begin contributing to the network more quickly, thereby improving network performance and efficiency.

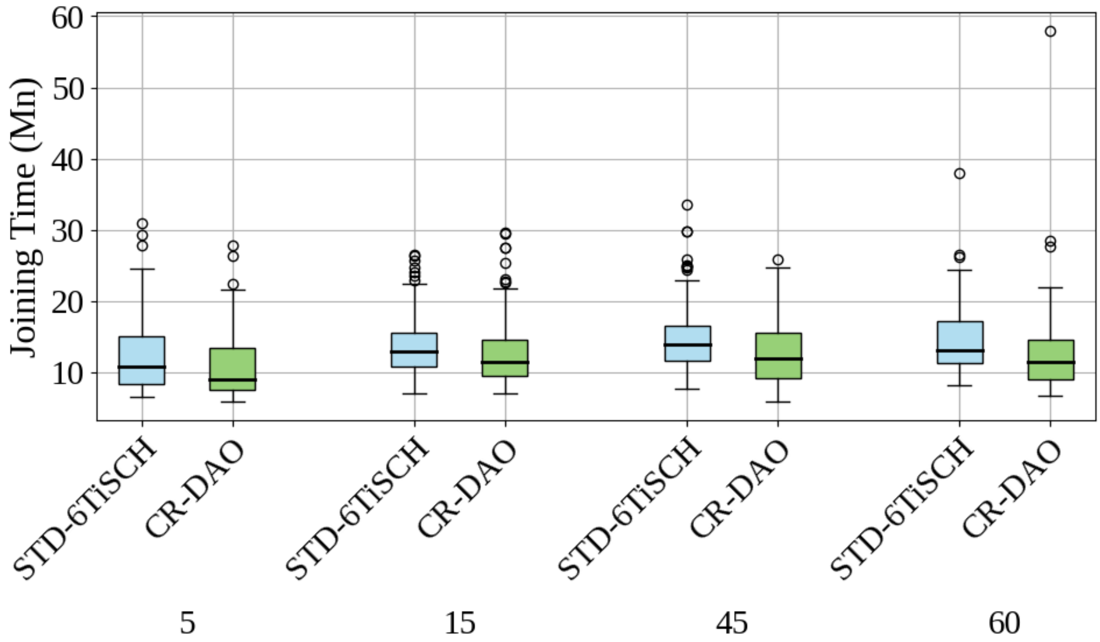


Figure 5.13: Network joining time (different application periods).

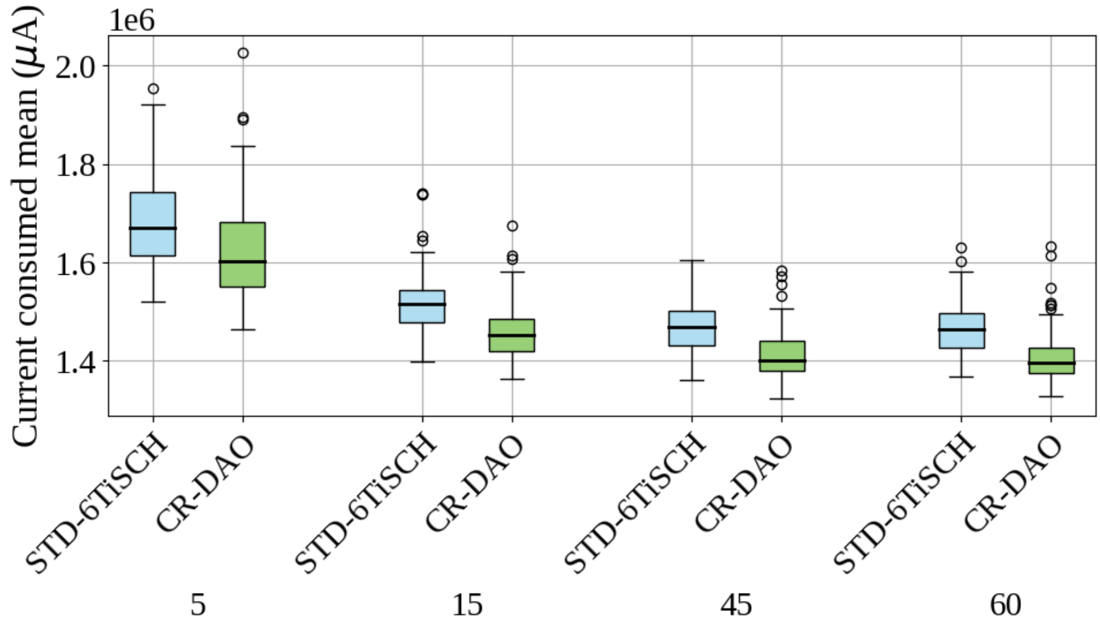


Figure 5.14: Current consumption (different application periods).

Furthermore, the average joining time remains stable across different application periods, suggesting that the joining process is not significantly influenced by changes in the period, reinforcing the robustness of our approach.

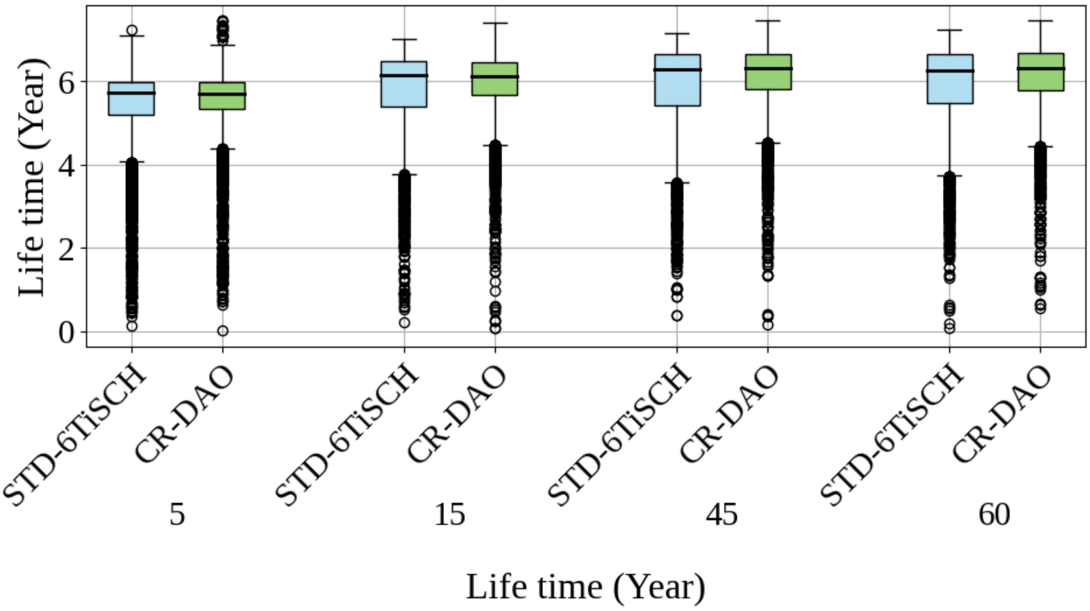


FIGURE 5.15: Node lifetime (different application periods).

5.4.6 Current Consumption

Fig. 5.14 presents the average current consumption for the studied methods, with CR-DAO consistently outperforming STD-6TiSCH. This improvement can be attributed to the reduced number of control packets required by CR-DAO, as control packets are smaller in size compared to those in the STD-6TiSCH method. As the application period increases, current consumption decreases due to fewer packet transmissions, resulting in lower energy usage.

5.4.7 Node Lifetime

Finally, in Fig. 5.15, we compare the node lifetime across the STD-6TiSCH and CR-DAO networks for different application periods. On average, the node lifetime is similar for both methods. However, a closer examination of the outliers reveals that CR-DAO can extend the longevity of nodes in certain scenarios, highlighting its potential to improve the operational lifespan of nodes when compared to the baseline method.

5.5 Conclusion

The results presented in this section clearly demonstrate the superiority of our CR-DAO method over the standard STD-6TiSCH protocol in various key performance areas, including packet exchange efficiency, latency, reliability, joining time, current consumption, and node lifetime. Our approach not only maintains the performance of the standard method but also improves it in critical areas, such as latency and packet delivery ratio, while simultaneously reducing energy consumption and increasing node lifetime.

General Conclusion

The growing demands of the Internet of Things (IoT) have emphasized the need for scalable, energy-efficient, and reliable communication systems. This thesis has explored these challenges within the framework of 6TiSCH networks, focusing on reducing energy consumption and improving reliability by optimizing control packet exchanges and enhancing cross-layer interactions.

This research proposed two key contributions to address the limitations of existing 6TiSCH networks by reduction in Control Packet Overhead, by minimizing the number of control packets exchanged during the interactions between the 6P protocol and the scheduling function, the first contribution significantly reduced the energy consumption of 6TiSCH networks. This optimization ensured that resource allocation processes were efficient without compromising network performance.

Through theoretical analysis and extensive simulations, both contributions demonstrated their effectiveness in reducing energy consumption and enhancing the reliability of 6TiSCH networks. The proposed solutions addressed the challenges of scalability, energy efficiency, and reliability, making them highly relevant to the development of next-generation IoT networks.

The findings of this thesis contribute to advancing the state-of-the-art in 6TiSCH networks by tackling critical issues that have hindered their deployment at scale. By optimizing resource management and control mechanisms, the proposed solutions enable IoT systems to operate more efficiently and sustainably. These improvements are particularly significant for applications requiring deterministic and reliable communication, such as industrial automation, smart cities, and healthcare monitoring.

Moreover, the research highlights the importance of cross-layer optimization in achieving high-performance IoT networks. By demonstrating the benefits of integrating scheduling functions with routing protocols, this work provides a framework for future innovations that could extend beyond 6TiSCH to other IoT and wireless communication standards.

While this thesis has addressed key challenges in 6TiSCH networks, it also opens avenues for future research. Potential directions include:

- Extending the proposed solutions to heterogeneous network environments involving diverse devices and communication standards.
- Investigating the scalability of the contributions in ultra-dense IoT networks with higher traffic demands.
- Exploring real-world implementations and experimental validations of the proposed solutions to assess their performance in practical settings.

In conclusion, this thesis has contributed to advancing the design and optimization of 6TiSCH networks by addressing the intertwined challenges of energy efficiency and reliability. The proposed solutions lay the groundwork for scalable and sustainable IoT systems, supporting the vision of a connected future where billions of devices communicate seamlessly and efficiently.

Bibliography

[1] Abdeldjalil Tabouche, Badis Djamaa, and Mustapha Reda Senouci. Traffic-Aware Reliable Scheduling in TSCH Networks for Industry 4.0: A Systematic Mapping Review. *IEEE Communications Surveys & Tutorials*, pages 1–1, 2023. ISSN 1553-877X, 2373-745X. doi: 10.1109/COMST.2023.3302157. URL https://ieeexplore.ieee.org/document/10208136/.

[2] Seema Kharb and Anita Singhrova. A survey on network formation and scheduling algorithms for time slotted channel hopping in industrial networks. *J. Netw. & Computer Appl.*, 126:59–87, 2019. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2018.11.004.

[3] Esteban Municio, Gilles Daneels, Miomir Vucinic, Serge Latré, Jef Famaey, Yousuke Tanaka, Klaus Brun, Kim Muraoka, Xavier Vilajosana, and Thiemo Watteyne. Simulating 6tisch networks. *Transactions on Emerging Telecommunications Technologies*, 30(5):e3494, 2019. doi: 10.1002/ett.3494. URL https://doi.org/10.1002/ett.3494.

[4] Nikumani Choudhury, Rakesh Matam, Mithun Mukherjee, and Jaime Lloret. A performance-to-cost analysis of ieee 802.15.4 mac with 802.15.4e mac modes. *IEEE Access*, 8:41936–41950, 2020. doi: 10.1109/ACCESS.2020.2976654.

[5] Tengfei Chang, Mališa Vučinić, Xavier Vilajosana, Simon Duquennoy, and Diego Roberto Dujovne. 6TiSCH minimal scheduling function (MSF). RFC 9033, May 2021. URL https://www.rfc-editor.org/info/rfc9033.

[6] Qin Wang, Xavier Vilajosana, and Thomas Watteyne. 6TiSCH Operation Sublayer (6top) Protocol (6P). RFC 8480, November 2018. URL https://www.rfc-editor.org/info/rfc8480.

[7] Suk Kyu Lee, Mungyu Bae, and Hwangnam Kim. Future of iot networks: A survey. *Applied Sciences*, 7(10), 2017. ISSN 2076-3417. doi: 10.3390/app7101072. URL https://www.mdpi.com/2076-3417/7/10/1072.

[8] Jeonggil Ko, Andreas Terzis, Stephen Dawson-Haggerty, David E. Culler, Jonathan W. Hui, and Philip Levis. Connecting low-power and lossy networks to the internet. *IEEE Communications Magazine*, 49(4):96–101, 2011. doi: 10.1109/MCOM.2011.5741163.

[9] Md Turab Hossain, Jesmin Binti, and Rayhan Uddin. A review paper on ipv4 and ipv6: A comprehensive survey. *American Journal of Computer Science and Technology*, 07:170–175, 10 2024. doi: 10.11648/j.ajcst.20240704.14.

[10] Nuno Galego, Rui Pascoal, and Pedro Brandão. *IPv6 in IoT*, pages 89–94. 01 2024. ISBN 978-3-031-44130-1. doi: 10.1007/978-3-031-44131-8_9.

[11] Kingsley Igulu, Friday Onuodu, and Thipendra P Singh. *IPV6: Strengths and Limitations*, pages 147–172. 03 2024. ISBN 978-981-97-0051-6. doi: 10.1007/978-981-97-0052-3_8.

[12] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. IEEE 802.15.4e: A survey. *Computer Communications*, 88:1–24, 2016. ISSN 01403664. doi: 10.1016/j.comcom.2016.05.004.

[13] Rajeev Piyare, George Oikonomou, and Atis Elsts. Tsch for long range low data rate applications. *IEEE Access*, PP:1–1, 12 2020. doi: 10.1109/ACCESS.2020.3046769.

[14] Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, 2016. doi: 10.1109/IEEESTD.2016.7460875.

[15] T. Watteyne, V. Handziski, X. Vilajosana, S. Duquennoy, O. Hahm, E. Baccelli, and A. Wolisz. Industrial wireless ip-based cyber x2013;physical systems. *Proceedings of the IEEE*, 104(5):1025–1038, May 2016. ISSN 0018-9219. doi: 10.1109/JPROC.2015.2509186.

[16] Pascal Thubert and Michael Richardson. Routing for RPL (Routing Protocol for Low-Power and Lossy Networks) Leaves. RFC 9010, April 2021. URL https://www.rfc-editor.org/info/rfc9010.

[17] Xavier Vilajosana, Kris Pister, and Thomas Watteyne. Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration. RFC 8180, May 2017. URL https://www.rfc-editor.org/info/rfc8180.

[18] Tengfei Chang, Mališa Vučinić, Xavier Vilajosana, Simon Duquennoy, and Diego Roberto Dujovne. 6TiSCH Minimal Scheduling Function (MSF). Internet-Draft draft-ietf-6tisch-msf-10, Internet Engineering Task Force, 2019. URL https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-msf-10. Work in Progress.

[19] Mina Rady, Quentin Lampin, Dominique Barthel, and Thomas Watteyne. 6DYN: 6TiSCH with heterogeneous slot durations. *Sensors*, 21(5):1611, Feb. 2021. ISSN 1424-8220. doi: 10.3390/s21051611.

[20] Glenn Daneels, Dries Van Leemput, Carmen Delgado, Eli De Poorter, Steven Latré, and Jeroen Famaey. Parent and PHY selection in slot bonding IEEE 802.15.4e TSCH networks. *Sensors*, 21:5150, Jul. 2021. doi: 10.3390/s21155150.

[21] Xenofon Fafoutis, Atis Elsts, George C. Oikonomou, Robert J. Piechocki, and Ian Craddock. Adaptive static scheduling in IEEE 802.15.4 TSCH networks. In *Proc. IEEE World Forum IoT (WF-IoT)*, pages 263–268, 2018. doi: 10.1109/WF-IoT.2018.8355114.

[22] Yasuyuki Tanaka, Pascale Minet, Mališa Vučinić, Xavier Vilajosana, and Thomas Watteyne. Ysf: A 6tisch scheduling function minimizing latency of data gathering in iiot. *IEEE Internet of Things Journal*, 9(11):8607–8615, 2022. doi: 10.1109/JIOT.2021.3118017.

[23] Glenn Daneels, Bart Spinnewyn, Steven Latré, and Jeroen Famaey. ReSF: recurrent low-latency scheduling in IEEE 802.15.4e TSCH networks. *Ad Hoc Netw.*, 69:100–114, 2018. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2017.11.002.

[24] Min-Jae Kim and Sang-Hwa Chung. Efficient route management method for mobile nodes in 6tisch network. *Sensors*, 21:3074, 04 2021. doi: 10.3390/s21093074.

[25] Seungbeom Jeong, Jeongyeup Paek, Hyung-Sin Kim, and Saewoong Bahk. TESLA: Traffic-aware elastic slotframe adjustment in TSCH networks. *IEEE Access*, 7:130468–130483, Sept. 2019. doi: 10.1109/ACCESS.2019.2940457.

[26] Burak Aydın, Sedat Görmüş, Hakan Aydın, and Sercan Külcü. A new routing objective function for IETF 6TiSCH protocol. In *Proc. Sig. Process. Commun. Appl. Conf. (SIU)*, pages 1–4, 2022. doi: 10.1109/SIU55565.2022.9864711.

[27] Tayfun Bekar, Sedat Görmüş, Burak Aydın, and Hakan Aydın. Q-learning algorithm inspired objective function optimization for IETF 6TiSCH networks. In *Proc. Int. Conf. Smart Appl. Commun. Network. (SmartNets)*, pages 1–6, 2023. doi: 10.1109/SmartNets58706.2023.10216124.

[28] Alakesh Kalita, Abhishek Hazra, and Mohan Gurusamy. Efficient schemes for improved performance in 6tisch networks. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, 2023. doi: 10.1109/INFOCOMWKSHPS57453.2023.10225947.

[29] Lucas Aimaretto and Diego Dujovne. Enhancing end-to-end determinism and reliability in 6tisch networks with disjoint leaf-based mpls-like tunnels. *Internet of Things*, 24:100988, 2023. ISSN 2542-6605. doi: https://doi.org/10.1016/j.iot.2023.100988. URL https://www.sciencedirect.com/science/article/pii/S2542660523003116.

[30] Dzaky Fawwaz and Sang-Hwa Chung. Intelligent parent change to improve 6tisch network transmission using multi-agent q-learning. *IEEE Access*, PP:1–1, 01 2024. doi: 10.1109/ACCESS.2024.3433589.

[31] Karnish Tapadar, Manas Khatua, and Venkatesh Tamarapalli. Imsf: Improved minimal scheduling function for link scheduling in 6tisch networks. In *Proc. Int. Conf. Dist. Comput. and Network.*, pages 124–127, 01 2022. doi: 10.1145/3491003.3491027.

[32] Dario Fanucchi, Barbara Staehle, and Rudi Knorr. Network formation for industrial IoT: Evaluation, limits and recommendations. In *Proc. IEEE Int. Conf. Emerg. Technol. Factory Automation (ETFA)*, volume 1, pages 227–234, 2018. doi: 10.1109/ETFA.2018.8502509.

[33] Alakesh Kalita and Manas Khatua. Opportunistic transmission of control packets for faster formation of 6TiSCH network. *ACM Trans. IoT*, 2(1):1–29, February 2021. ISSN 2691-1914, 2577-6207.

[34] Mohamed Mohamadi, Badis Djamaa, Mustapha Reda Senouci, and Abdelhamid Mellouk. Fan: Fast and active network formation in ieee 802.15.4

tsch networks. *Journal of Network and Computer Applications*, 183-184: 103026, 2021. ISSN 1084-8045. doi: https://doi.org/10.1016/j.jnca.2021.103026. URL https://www.sciencedirect.com/science/article/pii/S1084804521000527.

[35] Jeongbae Park and Sang-Hwa Chung. Enhanced network bootstrapping technique using multi-channel in 6tisch minimal cells. *IEEE Access*, PP:1–1, 01 2025. doi: 10.1109/ACCESS.2025.3539276.

[36] Thomas Watteyne, Maria Rita Palattella, and Luigi Alfredo Grieco. Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. RFC 7554, May 2015. URL https://www.rfc-editor.org/info/rfc7554.

[37] Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, March 2012. URL https://www.rfc-editor.org/info/rfc6550.

[38] Pascal Thubert. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). RFC 6552, March 2012. URL https://www.rfc-editor.org/info/rfc6552.

[39] Xavier Vilajosana, Qin Wang, Fabien Chraim, Thomas Watteyne, Tengfei Chang, and Kristofer Pister. A realistic energy consumption model for tsch networks. *Sensors Journal, IEEE*, 14:482–489, 02 2014. doi: 10.1109/JSEN.2013.2285411.

[40] Tengfei Chang, Thomas Watteyne, Qin Wang, and Xavier Vilajosana. LLSF: Low latency scheduling function for 6TiSCH networks. In *Proc. Int. Conf. Dist. Comput. Sensor Syst. (DCOSS)*, pages 93–95, 2016. doi: 10.1109/DCOSS.2016.10.

[41] Xavier Vilajosana, Thomas Watteyne, Tengfei Chang, Mališa Vučinić, Simon Duquennoy, and Pascal Thubert. IETF 6TiSCH: A tutorial. *IEEE Commun. Surveys Tuts.*, 22(1):595–615, Jan. 2020. ISSN 1553-877X. doi: 10.1109/COMST.2019.2939407.

[42] Fransiskus Xaverius Kevin Koesnadi and Sang-Hwa Chung. Improved cell allocation strategies using k-means clustering in congested 6tisch environments. *Sensors*, 24(17), 2024. ISSN 1424-8220. doi: 10.3390/s24175608. URL https://www.mdpi.com/1424-8220/24/17/5608.

[43] Karnish N.A. Tapadar, Manas Khatua, and Venkatesh Tamarapalli. Traffic rate agnostic end-to-end delay optimization using receiver-based adaptive link scheduling in 6tisch networks. *Ad Hoc Networks*, 155:103397, 2024. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2024.103397. URL https://www.sciencedirect.com/science/article/pii/S1570870524000088.

[44] Min-Jae Kim and Sang-Hwa Chung. Efficient route management method for mobile nodes in 6tisch network. *Sensors*, 21(9), 2021. ISSN 1424-8220. doi: 10.3390/s21093074. URL https://www.mdpi.com/1424-8220/21/9/3074.

[45] Xavier Vilajosana, Qin Wang, Fabien Chraim, Thomas Watteyne, Tengfei Chang, and Kristofer S. J. Pister. A realistic energy consumption model for tsch networks. *IEEE Sensors Journal*, 14(2):482–489, 2014. doi: 10.1109/JSEN.2013.2285411.

[46] Ahlam Hannachi, Wael Jaafar, Salim Bitam, and Nabil Ouazene. A novel energy-efficient cross-layer design for scheduling and routing in 6tisch networks. *Computer Communications*, 239:108173, 2025. ISSN 0140-3664. doi: https://doi.org/10.1016/j.comcom.2025.108173. URL https://www.sciencedirect.com/science/article/pii/S0140366425001306.

[47] Ahlam Hannachi, Wael Jaafar, and Salim Bitam. On the optimization of control packet management in mesh 6tisch networks. In *2024 IEEE 10th World Forum on Internet of Things (WF-IoT)*, pages 601–606, 2024. doi: 10.1109/WF-IoT62078.2024.10811247.