Mälardalen University Licentiate Thesis 385

Timing and security in IoT systems: Characterization and edge-centric approaches

Sebastian Leclerc

Mälardalen University

Mälardalen University Press Licentiate Theses
No. 385

TIMING AND SECURITY IN IoT SYSTEMS

CHARACTERIZATION AND EDGE-CENTRIC APPROACHES

Sebastian Leclerc

2026

Department of Computer Science & Engineering

Copyright © Sebastian Leclerc, 2026
ISBN 978-91-7485-759-7
ISSN 1651-9256
Printed by E-Print AB, Stockholm, Sweden

"The infidelity of the educated world can be traced to a principle which, in my judgment, explains the failure of all argument and evidence: namely, contempt prior to examination."

– William Paley, *A View of the Evidences of Christianity* (1794)

Acknowledgement

So much has changed over the past few years, and the journey certainly has not been a straight one. No traditional graduation, returning alone to my hometown after many years away, lacking qualifications, only to now sit here writing my licentiate thesis. Thank God for the possibilities and perseverance.

None of this would have been possible without all the people I have met along the way. Everyone who has simply been there, helping, discussing, hugging, sharing coffee, crying, laughing, holding on, and letting go.

In no particular order, I would like to extend a special thank you to my main supervisor, Alessio, for guiding me over the past two years, always being available, helping clarify uncertainties, and improving my work. I would also like to thank Mohammad, who has likewise contributed through discussions and by helping shape and strengthen the work. I am deeply grateful to my family, especially my parents, who always support me in all my silly endeavors. To all the teachers and supervisors I have had over the years, especially Bjarne, Shariar, Sara, and others, thank you. To all old and new friends throughout these years, including the networking, later embedded, crew Andreas and Samuel, joined along the way by Kasra. And a big hug to Daniel, Amir, Kaspar, and all the other unnamed friends.

Sebastian Leclerc
Västerås, July, 2026

Abstract

The rapid growth of the Internet of Things (IoT) has led to increasingly complex and data-intensive systems, often with conflicting or interdependent application requirements. Many IoT applications have strict timing requirements, particularly for safety-critical services, e.g., in the industrial domain. However, due to the vastness of the IoT ecosystem, decisions about where, how, and what timing attributes to measure are often ad hoc and context-specific, fragmenting the community's understanding of what "timing" entails. Moreover, security and privacy requirements are often overlooked within IoT. Common approaches to uphold confidentiality, integrity, and availability are often not suitable for resource-constrained IoT devices, as they may introduce unacceptable overhead. Consequently, designing a secure and time-critical IoT system is a challenging task.

These challenges motivate the goal of this licentiate thesis, which is to *support timing and security in IoT systems through edge-centric approaches.* To achieve this, the research follows an Action Design Research-inspired process and combines systematic literature reviews, surveys, and experimental evaluations using real IoT devices. The findings include a consolidated view of timing definitions, a cross-domain synthesis of explicitly reported timing-related requirements, and a unified categorization of timing metrics in the literature. In addition, comparative experiments of edge-centric data-reduction techniques highlight accuracy and reduction trade-offs for different applications, while a study of lightweight security mechanisms demonstrates how protocol choices and configuration influence timing behavior.

Sammanfattning

Den snabba tillväxten av Internet of Things (IoT) har lett till alltmer komplexa och dataintensiva system, ofta med motstridiga eller ömsesidigt beroende applikationskrav. Många IoT-applikationer har strikta tidskrav, särskilt för säkerhetskritiska tjänster, till exempel inom industrin. På grund av IoT-ekosystemets omfattning är dock beslut om var, hur och vilka tidsattribut som ska mätas ofta ad hoc och beroende av specifika tillämpningar eller kontexter, vilket fragmenterar forskningsområdets förståelse av vad ”tid” innebär. Vidare förbises ofta krav på säkerhet och integritet inom IoT. Vanliga metoder för att upprätthålla sekretess, integritet och tillgänglighet är ofta inte lämpliga för resursbegränsade IoT-enheter, eftersom de kan medföra oacceptabel resurskostnad. Följaktligen är det en utmanande uppgift att designa ett säkert och tidskritiskt IoT-system.

Dessa utmaningar motiverar målet med denna licentiatavhandling, vilket är att *stödja tidsaspekter och säkerhet i IoT-system genom edge-centrerade tillvägagångssätt*. För att uppnå detta följer forskningen en process inspirerad av Action Design Research och kombinerar systematiska litteraturstudier, enkäter och experimentella studier med verkliga IoT-enheter. Resultaten inkluderar en samlad bild av tidsdefinitioner, en tvärdomänsyntes av explicit rapporterade tidsrelaterade krav samt en enhetlig kategorisering av tidsmått i litteraturen. Dessutom belyser experimentella jämförelser av edge-centrerade datareduktionstekniker avvägningar mellan noggrannhet och reduktion för olika tillämpningar, medan en studie av resurseffektiva säkerhetsmekanismer visar hur val av protokoll och konfiguration påverkar tidsaspekter.

List of Publications

Papers included in this thesis[1]

Paper A: Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei. *Characterizing Time-Critical Internet of Things.* Published in Elsevier Internet of Things journal, 2025.

Paper B: Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei. *Systematic Consolidation of IoT Communication Requirements Across Domains.* Under review for Elsevier Internet of Things journal, 2026.

Paper C: Sebastian Leclerc, Emma Hansen, Alessio Bucaioni, Mohammad Ashjaei. *Reducing IoT Data at the Edge: A Comparative Evaluation.* Published in IEEE International Conference on Industrial Technology (ICIT), 2026.

Paper D: Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei. *On the Latency of Secure MQTT over QUIC vs. TCP: An experimental evaluation.* Accepted for presentation at IEEE International Smart Cities Conference (ISC2), 2026.

[1]The included papers have been reformatted to comply with the thesis layout.

Publications not included in this thesis

Paper E: Sebastian Leclerc, Kasra Ekrad, Bjarne Johansson, Inés Alvarez Vadillo, Mohammad Ashjaei, and Saad Mubeen. *Towards Developing a Supervisory Agent for Adapting the QoS Network Configurations.* IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2024.

Paper F: Kasra Ekrad, **Sebastian Leclerc**, Bjarne Johansson, Inés Alvarez Vadillo, Saad Mubeen, Mohammad Ashjaei. *Real-Time Fault Diagnosis of Node and Link Failures for Industrial Controller Redundancy.* IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2024.

Contents

I Thesis **xv**

1 Introduction **1**

2 Foundations of timeliness and security in IoT systems **5**
2.1 Architectural impact on timeliness and security 6
2.2 Timeliness in software 7
2.3 Timeliness in communication 8
2.4 IoT security 9
2.5 Validating timing and security 10

3 Research plan **13**
3.1 Overall research goal 13
3.2 Research questions 14
3.3 Research contributions 16
3.4 Research process and methods 18
3.5 Included papers 21
3.5.1 Paper A 22
3.5.2 Paper B 23
3.5.3 Paper C 24
3.5.4 Paper D 25

4 Related work **27**
4.1 Timing definitions, requirements, and measurement 27
4.2 On leveraging the edge and data reduction 29
4.3 On IoT security 30

5 Summary and future work **33**
5.1 Future work 34

Bibliography **36**

II Included Papers **45**

6 Paper A **47**
6.1 Introduction . . . 40
6.2 Background . . . 41
6.2.1 Internet of Things Fundamentals . . . 42
6.2.2 Real-time Fundamentals and Timeliness . . . 43
6.3 Research Methodology . . . 44
6.3.1 Definition of Research Goal and Questions . . . 45
6.3.2 Screening Process . . . 45
6.3.3 Extraction Form and Data Extraction . . . 53
6.3.4 Data Analysis and Synthesis . . . 53
6.3.5 Survey . . . 54
6.3.6 Threats to validity . . . 54
6.4 Definitions of time-criticality and predictability (RQ1) . . . 55
6.4.1 Time-critical definition . . . 55
6.4.2 Predictability definition . . . 60
6.5 Timing metrics (RQ2) . . . 64
6.5.1 Application layer metrics . . . 70
6.5.2 Cloud layer metrics . . . 71
6.5.3 Edge/Fog layer metrics . . . 71
6.5.4 Network layer metrics . . . 72
6.5.5 Perception layer metrics . . . 73
6.6 Discussion . . . 74
6.6.1 Relevance of the research topics . . . 75
6.6.2 Practitioners Perspective . . . 76
6.6.3 Identifying, Capturing, and Presenting Timing Data 82
6.7 Related work . . . 84
6.8 Conclusion and Future Work . . . 86

7 Paper B **97**
7.1 Introduction . . . 100
7.1.1 Contributions . . . 101
7.1.2 Paper outlines . . . 102
7.2 Background . . . 102
7.3 Related work . . . 103
7.4 Research methodology . . . 104
7.4.1 Definition of research goal and questions . . . 105
7.4.2 Data extraction form . . . 105
7.4.3 Snowballing process . . . 106

7.4.4 Data analysis and synthesis . . . 110
7.4.5 Threats Towards Validity . . . 112
7.5 Internet of things requirements . . . 113
7.5.1 Requirement categories . . . 114
7.5.2 Requirements' domains and applications . . . 117
7.5.3 Consolidated requirements . . . 118
7.6 Discussion . . . 125
7.6.1 Scarcity and implicit nature of IoT requirements . 125
7.6.2 Interpretation and realism of reported requirements 126
7.6.3 Coverage, biases, and layered perspectives . . . 127
7.6.4 Implications for research and practice . . . 128
7.7 Conclusion and Future Work . . . 129
7.8 Data Availability . . . 130

8 Paper C **137**
8.1 Introduction . . . 141
8.2 Background and Related Work . . . 142
8.3 Framework for comparing edge data reduction techniques 144
8.3.1 Experimental Design and Methodology . . . 144
8.3.2 Implementation and Experimental Setup . . . 145
8.3.3 Overview and Visualization of Data Reduction Algorithms . . . 145
8.4 Results . . . 148
8.5 Discussion . . . 150
8.5.1 Choosing Data Reduction Methods: Accuracy vs. Efficiency . . . 152
8.5.2 Threats to Validity . . . 152
8.6 Conclusions . . . 153
8.7 Data Availability . . . 154

9 Paper D **157**
9.1 Introduction . . . 159
9.2 Background and Related work . . . 160
9.3 Experimental Setup and Methodology . . . 162
9.3.1 Hardware platforms and Network . . . 165
9.3.2 Operating System Configuration and CPU Isolation 165
9.3.3 Software Stack and MQTT Deployment . . . 166
9.3.4 Time Synchronization and Timestamping . . . 166
9.3.5 Security Configuration . . . 167
9.4 Experimental Design and Results . . . 167

9.5 Discussion 175
9.6 Conclusion and Future Work 177

Table of Abbreviations

Abbreviation	Meaning
ADR	Adaptive Data Rate
BLE	Bluetooth Low Energy
COAP	Constrained Application Protocol
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CPS	Cyber-Physical Systems
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DOS	Denial of Service
E2E	End-to-End
HTTPS	Hypertext Transfer Protocol Secure
IIOT	Industrial Internet of Things
IOT	Internet of Things
IRQ	Interrupt Request
ISR	Interrupt Service Routine
ISO	International Organization for Standardization
IT	Information Technology
KPI	Key Performance Indicator
LAN	Local Area Network
MAC	Medium Access Control
ML	Machine Learning
MLO	Multi-Link Operation
MQTT	Message Queuing Telemetry Transport
NIC	Network Interface Controller
OSI	Open Systems Interconnection
OS	Operating System
PDR	Packet Delivery Ratio
QOS	Quality of Service
RC	Research Challenge
RE	Requirements Engineering
RG	Research Goal
RQ	Research Question
RT	Real-Time
RTA	Response Time Analysis
RTOS	Real-Time Operating System
RTT	Round-Trip Time
SDN	Software-Defined Networking
SWAPC	Size, Weight, and Power Consumption
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
UDP	User Datagram Protocol
WCET	Worst-Case Execution Time
WCRT	Worst-Case Response Time
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network

Part I

Thesis

Chapter 1

Introduction

The Internet of Things (IoT) is a vast ecosystem, with origins in the 1980s idea of ubiquitous computing, meaning seamless computing in everyday objects [1]. IoT can be defined as a network of interconnected devices that sense, process, and exchange data from the physical and virtual world [2]. Today, IoT market forecasts point toward significant expansion, reaching an estimated 690 billion U.S. dollars by 2030[1]. IoT systems incorporate multiple technologies, producing value for individual users and corporations across many domains [1, 2]. System components commonly include resource-constrained devices, data-producing sensors, microcontrollers, and communication links that connect distributed devices with geographically closer edge nodes and with more distant cloud computing infrastructures [3, 4].

To organize this ecosystem, many different IoT reference architectures have been proposed [4], ranging from three-layer models to multilayer variants that include additional business concerns. The concept of edge computing emerged approximately three decades ago [5], originally aimed at alleviating infrastructure bottlenecks for content providers by offering distributed on-demand computing closer to users. Since then, edge computing has gained substantial research interest due to potential benefits such as scalability, improved Real-Time (RT) performance, and on-premise preprocessing close to the data source [4].

Given this architectural diversity and broad range of application domains, IoT systems exhibit highly varying requirements, particularly with respect to timing. However, two issues arise with the vastness of the IoT ecosystem, namely (i) there is no shared definition of what

[1] `https://www.statista.com/statistics/1369033/global-iot-market-size`

"*timing*" entails within the community, and (ii) the spectrum of requirements is highly heterogeneous. For instance, regarding the former, our work in [6] identified 24 distinct definitions of time-criticality in IoT. Regarding the latter, latency requirements vary significantly across domains and contexts as seen from our work in [7]. Even within a single system, different services may exhibit widely varying requirements. In the industrial domain, tight control loops are used in automation systems to process raw materials into finished products in factory lines. These may require upper-bound latencies between 0.0002 and 2 ms [8]. In contrast, a logging application without immediate operational consequences within the same system may tolerate latencies of up to 1000 ms [9].

Security and privacy are likewise important aspects of IoT systems. However, prior literature persistently highlights security gaps in IoT, for example, from the practitioners' perspective in study [10]. Security requirements are often described as qualitative system-level properties, rooted in the Confidentiality, Integrity, and Availability (CIA)-triad, and are typically enforced through standards and best practices, which vary across domains, organizations, and regulatory contexts. For instance, the aforementioned industrial automation system may emphasize security requirements such as infrastructure monitoring, and security by design [11]. These systems may need to comply with frameworks such as the International Electrotechnical Commission (IEC) 62443 series for industrial communication and system security, as well as International Organization for Standardization (ISO)/IEC 27000 for information security management [12]. In general, security solutions introduce a temporal overhead. Given the industrial example, the security solutions must be lightweight and predictable, as they are constrained by strict timing requirements.

Although both timing and security are well-recognized concerns, the combination of both qualities remains limited. Thus, timing and security were chosen as two critical qualities, as opposed to others, for the dependable operation of time-critical IoT.

These challenges motivate the goal of this licentiate thesis, which is to *support timing and security in IoT systems through edge-centric approaches.* To this end, the thesis first aims to establish an understanding of how timing is conceptualized in IoT, with a focus on selected aspects. It then explores a subset of technical mechanisms to support both timing and security requirements in IoT.

This thesis is organized into two main parts: the Thesis (Part I) and the Included Papers (Part II). Part I begins with this introduction in Sec-

tion 1. In Section 2, the foundational terminology and core concepts are provided. Section 3 details the research plan, including the overarching goal, identified challenges, intended contributions, the methodological approach guiding the work, and the peer-reviewed publications forming the basis of the thesis. Section 4 situates the thesis within the broader related work. Finally, Section 5 summarizes the work and presents some possible future research directions. Part II contains the papers included in this thesis.

Chapter 2

Foundations of timeliness and security in IoT systems

In this work, timeliness refers to the system's ability to perform its tasks (sensing, processing, communication, etc.) within specified application time requirements. Meanwhile, security generally denotes the preservation of CIA of application data as it moves across the system.

In general, timeliness in the IoT systems is influenced by several factors, including the architecture, hardware characteristics, software execution behavior, the capabilities of the Operating System (OS), communication mechanisms, the scale, and the environment and scenario in which the system operates [13–15]. Further complicating matters, security mechanisms can also affect timing behavior, as additional protection measures may introduce overhead or restrict available resources.

To describe some of the aspects that influence the timeliness and security of an IoT system, the high-level system model shown in Figure 2.1 is used. The system consists of temperature sensors in the perception layer that are wirelessly connected (dotted lines) to an intermediary edge layer. The edge layer, in turn, has wired connections (dashed lines) via the internet to a cloud layer, which interfaces with end users.

The level of control over the system components and its correlation with required security, latency, and resources is discussed further in the following background sections.

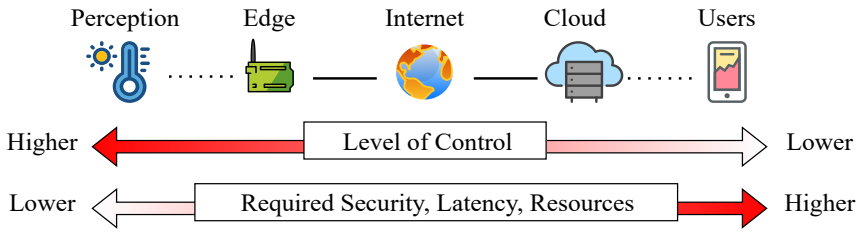

Figure 2.1: A high-level model of the components of the secure time-critical IoT system, generalizing the system level of control across different layers and its implications.

2.1 Architectural layers and their impact

In this work, the IoT system architecture is represented using four abstraction layers: the perception, edge, cloud, and intermediary network layers.

The perception layer consists of embedded devices, typically sensors and actuators, that form the foundation of many IoT applications [3, 4]. These systems rely on Size, Weight, Power, and Cost-Constrained (SWaP-C) hardware platforms that, for instance, collect environmental data and generate time-critical events based on physical stimuli. Above this, the edge layer (also referred to as a sink or gateway), typically located within the geographically closer Local Area Network (LAN) segment, may process the data locally or forward it to higher-level computing layers. This layer is generally less SWaP-C and consists of, for example, a Raspberry Pi during prototyping or production-grade hardware in operational deployments. Further upstream, the cloud layer provides processing and storage in remote data centers (e.g., servers from Amazon Web Services), accessible over the internet and offering virtually unlimited scalable resources. The intermediate links between each layer may consist of various wired and wireless communication solutions, each with its own benefits and considerations.

Due to the devices located in the perception and edge layers being fully under system control, and geographically close to where the data is produced, this generally implies fewer security requirements, lower latencies, albeit lower resources, as opposed to when communication crosses the less controllable internet threshold towards the cloud. Edge computing, therefore, has the potential of providing several benefits. For instance, the edge can be utilized to pre-process data, e.g., by intelli-

gently reducing the amount of data exchanged between the edge and the cloud [16]. This reduction can lower communication overhead and costs, while also limiting data exposure from a privacy perspective, particularly as system scale grows. However, it introduces trade-offs in terms of computational overhead, data fidelity, and system complexity, which must be carefully balanced depending on the application. While global data traffic has increased exponentially with digitalization and Artificial Intelligence (AI) services, impacting energy consumption, data reduction is one potential mechanism toward more sustainable IoT system design [17]. Another approach is to bolster security as seen in our work in [18], with further background discussed in Section 2.4. The edge generally enables faster response times compared to cloud-centric solutions, an advantage that is especially valuable for time-critical tasks, thereby providing greater control over timing and security behavior.

2.2 Timeliness in software

On the software side, the choice of OS is particularly important: lightweight Real-Time Operating Systems (RTOSs) for embedded systems, or a light Linux-based system patched with PREEMPT_RT, can provide stronger temporal guarantees through mechanisms such as preemption, temporal isolation, and priority inheritance [13, 19].

Many time-critical events originate from peripheral devices or sensor thresholds [13, 19]. For instance, a sensor's register containing temperature values is sampled at a certain resolution, and the resulting measurements can then be mapped to high-priority aperiodic or sporadic tasks that trigger Interrupt Requests (IRQs) when a predefined temperature threshold is met. These interrupts are handled by Interrupt Service Routine (ISR) routines and scheduled by the OS, which preempts the currently running task to ensure responsiveness under load. More advanced software techniques incorporate task offloading schemes (allocation, distribution) to other nodes in the system in order to meet the schedule. To accommodate SWaP-C and unreliable wireless conditions, lightweight protocols such as request-response-based Constrained Application Protocol (CoAP) or publish-subscribe-based Message Queuing Telemetry Transport (MQTT) are commonly used [4]. In the publisher-subscriber model, information is organized into topics carrying associated data. Publishers transmit messages to a broker, which then distributes them to all interested subscribers.

2.3 Timeliness in communication

The underlying communication within the system also affects both timeliness and security. Connecting embedded devices to the network, particularly the internet, can introduce delays due to interrupts generated by the Network Interface Controller (NIC) [20]. Nodes in the perception layer generally transmit measurements wirelessly to an edge node [4]. From there, the edge node's connection to the wider internet may rely on either wired or wireless technologies, after which traffic typically traverses multiple service providers' infrastructures before reaching the cloud.

Timing performance in these communication channels depends on aspects such as the system scale, clock synchronization, transport protocols, and the functions of Layers 1 and 2 (L1, L2) of the Open Systems Interconnection (OSI) model [14, 21]. L1 handles signal modulation, encoding, and transmission strength, whereas L2 manages Medium Access Control (MAC) mechanisms that regulate medium access. At the transport layer, classic protocols include the connection-oriented Transmission Control Protocol (TCP) and the connectionless User Datagram Protocol (UDP). In addition, a more recent alternative, QUIC [22], which is originally introduced by Google as "Quick UDP Internet Connections", was designed to improve Hypertext Transfer Protocol Secure (HTTPS) performance and natively integrates Transport Layer Security (TLS) 1.3 for secure communication. Moreover, delays might occur due to poor reliability levels, e.g., messages being lost, dropped, fragmented, or arriving out-of-order and having to be re-transmitted [14,21]. Communication reliability is commonly expressed in terms of an adequate Packet Delivery Ratio (PDR) over a time period.

The wireless link between nodes commonly relies on technologies such as Wi-Fi, Bluetooth Low Energy (BLE), or cellular communication [14,21]. For example, Wi-Fi employs contention-based Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) in L2 and lacks deterministic features such as Time Division Multiple Access (TDMA) and channel hopping, which limits its suitability for applications with strict timing requirements. Due to the broadcast nature of wireless communication, delays from multi-path fading, interference, and congestion are common, especially in dense deployments.

The wired links utilize L2 switching and L3 routing devices, forwarding frames and packets between networks. Beyond the LAN, the network can be seen as a black box, as system control of its operation is limited.

One possible approach is to utilize Software-Defined Networking (SDN) techniques to manage and schedule slices of the available network bandwidth depending on the application's latency sensitivity, specified by the traffic's Quality of Service (QoS) requirements [23].

2.4 IoT security

Security is not a static property; it is dynamic and evolves through the continuous discovery and exploitation of vulnerabilities. In this sense, to paraphrase Dijkstra [24], security testing can reveal the presence of vulnerabilities, but never prove their absence. A broad set of threats and security mechanisms applies to IoT systems [25]. Unfortunately, IoT often lacks fundamental security principles [10]. For example, a Shodan[1] query performed in May 2026 for "MQTT" shows that approximately 400 000 devices were reachable on port 1883, the default unencrypted MQTT port.

Many of the threats, especially in the communication layers, can be mitigated using fundamental measures such as encryption [15, 26, 27]. When properly configured, encryption can provide confidentiality, integrity, and authentication. It is typically achieved through a combination of symmetric cryptography (shared keys) and asymmetric cryptography (public-private key pairs), together with appropriate key-distribution schemes. However, encryption consumes considerable resources, including additional transmission overhead and increased Central Processing Unit (CPU) workload, which may introduce delays. In general, asymmetric cryptography is less computationally efficient. The challenges are amplified by the key-distribution requirements, i.e., the need to generate, distribute, rotate, and revoke keys. A compromised key without timely replacement leaves the system vulnerable.

A widely adopted encryption protocol implementing these mechanisms is TLS 1.3 [28]. During connection establishment, endpoints negotiate cryptographic parameters, typically using ephemeral key exchange mechanisms (e.g., ECDHE), which provide forward secrecy, meaning that compromised long-term keys cannot be used to decrypt past communication. Authentication is usually achieved through certificates (e.g., X.509), while data transmission relies on efficient symmetric encryption once the secure channel is established. Building on TLS, QUIC is a natively secured transport protocol designed to improve performance over

[1] `www.shodan.io` - A search engine that indexes connected devices on the internet.

UDP [22,29]. By integrating the TLS handshake and supporting features such as reduced connection establishment latency (e.g., session resumption) and elimination of head-of-line blocking, QUIC can reduce communication delays, which is particularly relevant for time-critical IoT applications and is explored in our work [18].

Numerous security requirements, standards, and domain-specific recommendations exist, often rooted in the CIA-triad. For instance, in the industrial sector, systems may follow frameworks such as the IEC 62443 series for industrial communication and system security, ISO/IEC 27000 for information security management, and ISO/IEC 15408 for security evaluation, as well as General Data Protection Regulation (GDPR) when personal or customer data is involved [11,12]. In the medical sector, regulatory frameworks define security standards and best practices, such as Health Insurance Portability and Accountability Act (HIPAA) (in the US) and the Network and Information Security 2 (NIS2) Directive (in EU), supported by European Union Agency for Cybersecurity (ENISA) guidelines. In the energy sector, IEC 62351 standards recommend the use of TLS with digital certificates [26].

Beyond encryption, additional security mechanisms exist across multiple system layers, including physical protection, device authentication, and intrusion detection. While important, these mechanisms are beyond the scope of this work.

2.5 Validating timing and security

Validating timing behavior can be performed using several different methods and from different perspectives, ranging from formal analysis techniques to experimental evaluation in simulations or on real hardware [19,30,31]. It should be noted, however, that validating timing through experimental testbeds is challenging, since many measurement techniques are intrusive and can distort the results (e.g., using *printf()* for logging). Classical Response-Time Analysis (RTA) attempts to verify whether a system can meet its timing constraints under worst-case conditions, accounting for metrics such as the Worst-Case Execution Time (WCET), which captures the maximum uninterrupted execution time of tasks, and Worst-Case Response-Time (WCRT), which additionally accounts for blocking, interruptions, and scheduling delays. Communication delays can be measured from many different perspectives, e.g., Tx/Rx, upstream/downstream, Round-Trip Time (RTT), etc.

Security requirements are inherently difficult to quantify, as they are typically expressed as qualitative system-level properties rather than concrete metrics, and may appear satisfied until new vulnerabilities are discovered in, e.g., the software stack [15, 26]. One practical approach is to guide security through established standards and best-practice frameworks (e.g., ISO 27001) and to assess compliance through systematic security evaluations. This may involve activities such as vulnerability assessment of hardware and software, penetration testing of network interfaces, attempts to inject malicious data, input/output validation, and the use of automated security testing tools.

Chapter 3

Research plan

This section presents the adopted research plan that includes the Research Goal (RG), Research Questions (RQs), Research Challenges (RCs), process, and research methods.

3.1 Overall research goal

The IoT market has experienced significant growth in recent years, leading to a surge in data generated by various IoT applications [4, 10]. Alongside the larger data volumes, factors such as ecosystem heterogeneity and increased connectivity introduce numerous challenges, including security and privacy risks, scalability, sustainability, as well as economic costs for data processing and storage.

Simultaneously, many IoT domains also impose strict timing constraints, particularly in time-critical or safety-critical applications. The notion of "timing" is described using a range of intertwined concepts across different, and sometimes overlapping, research communities and domains. Multiple timing-related concepts exist, with a wide range of definitions and interpretations. These observations suggest a diverse perspective on timing, depending on factors such as the application context, system layer, requirements, and research community. This makes it difficult to reason about and compare how to evaluate timing requirements in different IoT domains.

Timing and security are closely intertwined in IoT systems. Security mechanisms introduce computational and communication overhead that directly impacts timing behavior. Conversely, strict timing constraints limit which security techniques can be applied. As a result, timing and

security must be considered jointly. Despite the growing importance of both security and timing requirements, existing research addressing these aspects collectively remains limited. Current approaches are often created for a particular purpose and tailored to specific use cases, which can limit their generalizability, motivating the need for more application-agnostic approaches.

There are several challenges in ensuring that a message is securely delivered between nodes in an IoT system. This difficulty is amplified by the inherent complexity of the IoT landscape: messages traverse multiple architectural layers, software and network stacks, and operate over non-standardized solutions across geographically wide and dynamic environments. Meeting both sufficient security and timing requirements under such conditions is non-trivial. Consequently, the goal of this thesis is as follows:

RG: *To support timing and security in IoT systems through edge-centric approaches.*

That is, the thesis first seeks to establish a clear understanding of how timing is conceptualized in the IoT literature and practice, with a focus on timing definitions, application requirements, and measurement metrics. It then explores a subset of mechanisms, focusing on data reduction and transport-layer security, that can help meet these timing demands.

3.2 Research questions

The aforementioned goal was broken down into three specific RQs, each contributing to different and unique aspects of the goal.

RQ1: *How is timing characterized and determined in the IoT literature?*

Here, timing denotes the system's temporal behavior in the software- and network-stack, with properties such as latency, jitter, predictability, and deadlines. Characterization of a phenomenon can encompass many different aspects; to narrow the scope, the timing characterization used here focuses on the following three components: definitions, along with the requirements and measurements that determine them. The first component concerns how timing concepts are defined within IoT systems, aiming to provide a common understanding for the remainder of the the-

sis. The second component covers the IoT application requirements in different domains, with a focus on requirements that have a relationship with the aforementioned timing concepts. Finally, the third component addresses how the validation of said timing concepts is measured and which metrics are used.

The major challenge of this question stems from how wide the IoT domain is, where different application areas often use different interpretations of timing-related concepts. The lack of a unified view motivates the need for a common understanding toward standardization. In answering this question, the aim is also to uncover potential gaps and narrow interesting research directions.

RQ2: *How can we reduce large data volumes generated from IoT devices while meeting application requirements?*

As noted in the introduction and background sections, the rapid expansion of the IoT market has resulted in a vast number of interconnected devices generating continuous streams of data. This development prompts the question of whether such extensive data generation is always required from an end-to-end perspective. Transmitting and processing large volumes of data can lead to intolerable delays and performance degradation. System-wide examples of this include the flooding of buffers and queues, longer transmission times, increased fragmentation of big payloads, and additional processing delays. Moreover, relying heavily on cloud infrastructure to handle such data volumes is costly, difficult to scale, and ultimately unsustainable. Additionally, exposing large amounts of data over unsecured channels elevates privacy risks by making sensitive information more vulnerable. Further complicating matters, different applications have different timing and security requirements. For example, reducing the amount of data sent towards the cloud might not be as critical for latency or confidentiality when regulating the temperature in a smart home application as it is in an industrial setting. Likewise, the impact of reduction errors varies across applications. For example, cumulative deviations may affect resource management in a water flow monitoring application, while a safety-critical application may require stricter guarantees regarding maximum deviation.

RQ3: *How do secure communication mechanisms affect latency under different operating conditions in IoT systems?*

While reducing the data volumes that IoT systems generate addresses one aspect of the problem, the system still requires sufficient security mechanisms when communicating over unsecured channels. This is relevant for virtually all IoT users, as compromised devices can expose sensitive data, disrupt critical services, or be repurposed for large-scale attacks.

The major challenge with security integration in IoT stems from the inherently resource-constrained nature of many perception- and edge-layer devices. Wireless communication is also less reliable than its wired counterpart, and establishing a secure communication channel consumes resources such as time, CPU cycles, and memory.

Consequently, appropriate strategies involve either balancing the level of security against available resources or developing and evaluating more efficient mechanisms that allow systems to maintain their timing requirements in diverse operating conditions. Given the breadth of security in IoT, this challenge motivates focusing on the temporal impact of selected secure communication mechanisms under varying connection and traffic conditions.

3.3 Research contributions

Hereafter, a discussion on the three main RCs tackling the above RQs and RG. Table 3.1 provides the mapping of RC to RQ.

	RQ1	**RQ2**	**RQ3**
RC1	X		
RC2		X	
RC3			X

Table 3.1: Mapping of research contributions to questions.

RC1: *Systematic identification, classification, and consolidation of timing characteristics, metrics, and requirements across IoT domains and architectural layers.*

This contribution provides a structured, detailed, and survey-validated characterization of two timing concepts (time-criticality and predictability) in IoT systems that includes (i) descriptive characterizations of each concept, (ii) an orthogonal analysis between each characteristic and the layers of an IoT system, and (iii) a categorization of timing metrics across an IoT system. Moreover, the contribution provides a structured and detailed synthesis of IoT requirements across different domains that includes (i) the identification of recurring requirement categories enabling cross-domain comparison and (ii) representative common and tail requirement values enabling early-stage design decisions.

The value of this contribution is multi-fold; it (i) exposes the ad hoc and fragmented handling of timing in IoT systems and contributes to the future standardization of time measurement in IoT and (ii) reveals key gaps in reported requirements and standardization efforts on the current state of IoT requirements. This research contribution is published in Papers A and B.

RC2: *Selection and evaluation of edge-centric data reduction techniques.*

This contribution aims to highlight the benefits of using edge-centric data-reduction techniques for resource-constrained IoT systems. It includes (i) the selection and experimental evaluation of representative techniques from three common reduction categories, (ii) a systematic parameter sweep across multiple real-world scenarios with differing temporal characteristics, and (iii) a multi-metric assessment of each technique to determine its suitability for different application scenarios.

This contribution provides practical guidance for data-reduction techniques, highlighting the importance of context-aware algorithm selection and parameter tuning based on data variability, error tolerance, and application requirements. This research contribution is published in Paper C.

RC3: *Latency analysis of lightweight transport-layer security mechanisms in edge-cloud IoT architectures.*

This contribution investigates the latency impact of secure communication in edge-cloud IoT systems through experimental evaluation. It

includes (i) a comparative study of MQTT over TCP and QUIC under different operating conditions, (ii) an analysis of latency under connection establishment, concurrent connection scaling, and sustained data streaming, and (iii) an evaluation of how transport protocol choice, cryptographic configuration, and endpoint capabilities influence end-to-end performance.

The results provide insights into where the primary latency costs of security arise and highlight trade-offs between transport protocols, security configurations, and system behavior. These findings support the design of time-critical IoT applications by informing the selection of secure communication mechanisms under representative traffic conditions. This research contribution is presented in Paper D.

3.4 Research process and methods

This section describes the research process and methods used throughout the research. The process describes the overall workflow that shaped the work towards the goal. The methods are the specific research methods that formed the contributions. Action Design Research (ADR) is a commonly accepted research process that provides methodological guidance for solving complex problems by building and evaluating Information Technology (IT) artifacts. The ADR-inspired process is illustrated in Figure 3.1.

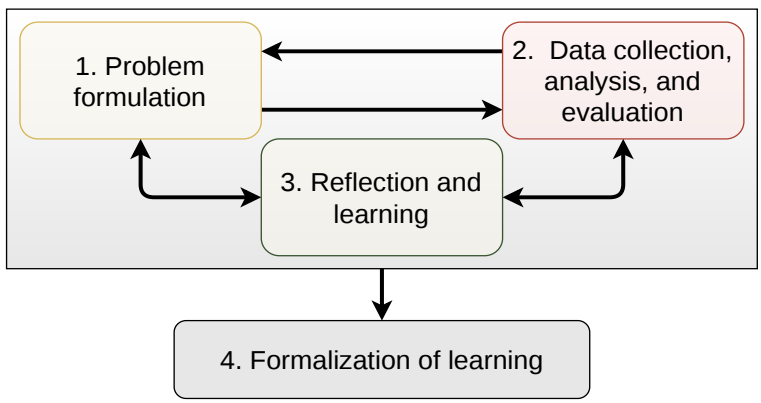

Figure 3.1: Adapted research process from [32].

In the initial problem step, a broad literature review was conducted to identify existing gaps and determine research needs relevant to both academia and industry. From this, the focus became on understanding timing in IoT systems and on exploring how edge-centric solutions could support time-critical applications. Based on the identified gaps, the RGs was formulated, and the corresponding RQs were derived. This step was iterative: the understanding of the problem evolved as the work progressed, and the research components were refined accordingly.

In the data collection, analysis, and evaluation step, work toward addressing each RC was carried out. Data was collected using Kitchenham's guidelines for secondary studies in software engineering [33] and through forward and backward snowballing as outlined by Wohlin [34]. For empirical input, a survey was conducted following the guidelines by Molléri et al. [35] and Kasunic's recommendations [36]. Automated searches in scientific databases were conducted, which have demonstrated strong support for systematic studies [37] using the PRISMA 2020 guidelines [38]. The study-selection process followed the approach by Ali and Petersen, emphasizing clear inclusion and exclusion criteria [39]. The analysis was carried out using a range of methods, as presented in the corresponding publications. For example, Grounded Theory served as a general framework for qualitative analysis, applying open coding as described in the foundational work [40]. Moreover, the recommendations on thematic synthesis described by Cruzes [41] were followed, combined with vertical content analysis [42] and horizontal narrative synthesis [43].

In the reflection and learning step, the outcomes of the previous stages were reviewed, refining the direction of the work where needed. For example, while working on [6] (Paper A) and considering the next steps, one idea was to evaluate different communication technologies suitable for time-critical IoT. However, during this stage, it was noted that some survey respondents emphasized the importance of grounding timing definitions in application-specific requirements. Inspired by this, and prior to assessing communication solutions, it became necessary to understand what these requirements actually are. This insight led to the formulation of [7] (Paper B), which investigates timing-related and domain-specific requirements to provide the foundation needed to reason about suitable communication mechanisms.

In the formalization of learning step, the insights generated throughout the research process were consolidated and transformed into peer-reviewed publications. This involved distilling the results, integrating feedback from supervisors, surveys and analyses, and

articulating the outcomes. For example, the method developed to compare data-reduction techniques across different application contexts was formalized in [16] (Paper C), where the step-by-step procedure was documented to enable replication, transparency, and reuse by other researchers.

3.5 Included papers

In this section, the papers that will be included in the licentiate thesis are presented. Table 3.2 illustrates the mapping between the papers and the RCs discussed in Section 3.3.

	RC1	**RC2**	**RC3**
Paper A	X		
Paper B	X		
Paper C		X	
Paper D			X

Table 3.2: Mapping of papers to contributions.

3.5.1 Paper A

Title: Characterizing Time-Critical Internet of Things

Authors: Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei

Abstract: The Internet of Things (IoT) is increasingly being adopted in diverse domains, many of which require strict timing constraints and predictable behavior. Despite the growing importance of timing characteristics in IoT applications, current approaches to address timing requirements are often fragmented, context-specific, and lack a unified understanding. Consequently, addressing timing aspects in IoT remains largely ad hoc and dependent on individual applications, making it challenging to generalize findings or systematically apply established solutions.

The goal of this study is to provide a comprehensive understanding of how timing is defined, characterized, and measured within the IoT community. We conducted this study through a systematic and structured mix methods research approach. First, we performed a systematic review of the literature, extracting and analyzing information from 38 primary studies, selected from a rigorous process involving 1176 studies. Second, to complement the literature findings, we conducted an expert survey involving 28 respondents from academia and industry, representing a variety of roles with specialized expertise in IoT systems and timing-related issues.

We identified two primary characterizations of timing within the IoT: time-criticality and predictability. Additionally, we collected and categorized 113 distinct timing metrics from literature into commonly found layers of an IoT system. The majority of the surveyed practitioners and researchers (75%) agree with our categorization and consider this research useful and relevant (71.5%). We believe that our study provides practitioners and researchers with insights into timing characteristics and metrics in IoT applications, towards the ultimate goal of standardization.

Personal contribution: I was the primary driver and main contributor, responsible for the literature search, screening, review, and data extraction, synthesis and analysis of the findings into the core contributions. I also contributed to the survey design questions, participant identification, and led the analysis of the survey results. Co-authors provided guidance throughout and reviewed the manuscript.

Status: Published in the Elsevier Internet of Things journal, 2025.

3.5.2 Paper B

Title: Systematic Consolidation of IoT Communication Requirements Across Domains

Authors: Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei

Abstract: The Internet of Things (IoT) increasingly powers cyber-physical (and human-in-the-loop) systems across many domains, where deployments must satisfy strict system and communication requirements, including timing, reliability, and data rate. Yet, although requirements are central to IoT engineering and evaluation, the literature often reports them in fragmented, implicit, and domain-specific ways, which prevents a consolidated view and limits reuse across application domains, technologies, and the cloud-to-thing continuum during early design stages.

The goal of this study is to provide a structured and consolidated overview of IoT requirements reported in the literature, with a particular focus on communication-related requirements. We conducted a systematic and fully recursive snowballing study. Starting from 5,571 candidate studies, we ultimately included 22 primary studies published within the past five years. From these studies, we extracted and synthesized reported IoT requirements across multiple application domains to enable cross-domain comparison.

Our analysis identifies seven recurring requirement categories (latency, data rate, reliability, scalability, data size, energy, and other), consolidates communication requirements by reporting representative common (90th percentile) and tail values, and compares current and futuristic IoT applications across domains. The results indicate a strong influence from cellular communication technologies and standardization efforts, and reveal a scarcity of reported requirements related to energy, security, and non-cellular communication solutions. This work provides researchers and practitioners with consolidated, application-grounded communication requirements that can guide early-stage IoT system design, technology selection, and evaluation across domains, while highlighting the need for more explicit reporting of energy, security, and non-cellular requirements to support trustworthy IoT ecosystems.

Personal contribution: I was the primary driver and led all major research activities. I conducted the snowballing-based literature search, screening, data extraction, and analysis, and synthesized the results into a consolidated set of IoT requirements. I also wrote the manuscript. Co-authors provided guidance and reviewed the paper.

Status: Submitted to the Elsevier Internet of Things journal, 2026.

3.5.3 Paper C

Title: Reducing IoT Data at the Edge: A Comparative Evaluation

Authors: Sebastian Leclerc, Emma Hansen, Alessio Bucaioni, Mohammad Ashjaei

Abstract: In resource-constrained Internet of Things environments, reducing data transmissions is essential for minimizing energy consumption, network load, and operational costs. Overly aggressive reduction may compromise accuracy, a critical factor in applications such as industrial control.

This paper aims to offer practical guidance for selecting suitable data reduction techniques by experimentally evaluating three promising methods from common reduction categories: Data Filtering, Data Aggregation, and Data Prediction. We perform a parameter sweep for each algorithm across three real-world temperature scenarios: stable, rising, and fluctuating. Each configuration is evaluated in terms of data reduction percentage and accuracy, using Total Accumulated Deviation, Mean Absolute Deviation, and Maximum Deviation.

Results show that Data Prediction generally achieves the highest accuracy across all scenarios, while Data Filtering tends to yield the greatest reduction at the expense of accuracy. However, all algorithms can be tuned to meet specific scenario demands or accuracy criteria, underscoring that no one-size-fits-all solution exists. We conclude that context-aware algorithm selection and parameter tuning are critical for effective Internet of Things data management.

Personal contribution: I guided the initial work that formed the basis of this contribution. I proposed the core research idea, which was further refined jointly by all co-authors. I subsequently extended the initial work by conducting additional literature review, performing further experiments, and carrying out a complete revision and extension of the manuscript.

Status: Published in IEEE International Conference on Industrial Technology (ICIT), 2026.

3.5.4 Paper D

Title: On the Latency of Secure MQTT over QUIC vs. TCP: An experimental evaluation

Authors: Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei

Abstract: Smart-city services increasingly depend on IoT communication infrastructures that must provide secure connectivity without compromising latency-sensitive operation. This creates an engineering trade-off: security mechanisms protect urban sensing and service platforms, but may also increase communication latency, especially when devices connect under load. QUIC addresses part of this trade-off by integrating secure connection establishment into the transport layer, making it a relevant candidate for MQTT-based smart-city deployments.

This study experimentally evaluates MQTT over QUIC and TCP under different security configurations, endpoint platforms, and traffic patterns. We separate three latency-relevant operating conditions: connection establishment, concurrent connection scaling, and sustained streaming in a multi-sensor, edge-to-cloud scenario motivated by smart-city applications.

The results indicate that, in the evaluated configurations, the main latency cost of security appears during connection establishment, whereas steady-state encrypted data transmission adds limited overhead across the tested traffic patterns. For single connections, QUIC reduces per-connection latency by up to around 3.2 times compared to TCP with TLS. Under concurrent connection bursts, QUIC shows linear tail-latency growth in the measurements, and under high streaming load, it degrades more gradually than the TCP-based alternatives. The findings identify practical trade-offs among transport protocol choice, cryptographic algorithm, key type, and size, and endpoint computing capability. These trade-offs inform the design of event-driven smart-city services in which devices must establish or maintain secure communication under constrained latency and hardware conditions.

Personal contribution: I was the primary driver of this work and led all major research activities. I set up the experimental testbed, collected and analyzed the data, and wrote the manuscript. All co-authors provided feedback throughout the process and reviewed the manuscript to improve its quality.

Status: Accepted for presentation at IEEE International Smart Cities Conference (ISC2), 2026.

Chapter 4

Related work

In this section, a summary of some of the related studies relevant to the thesis goal and contributions is provided.

4.1 Timing in IoT

Definitions: Although no prior study offers a formal or unified definition of "timing" in IoT, to the best of our knowledge, several studies implicitly characterize it through the aspects they analyze.

For example, study [44] discusses timing through notions such as predictability, determinism, and latency requirements across Industrial Internet of Things (IIoT) domains, including autonomous vehicles, worker safety, and augmented reality. In contrast, study [45] conceptualizes timing from the perspective of a specific use case, a life-critical support system which integrates large amounts of data from heterogeneous IoT nodes to support decision-making.

Study [46] frames timing from an architectural and analysis perspective, discussing single- and multi-core processors, memory interference, networked systems, and RTA.

Study [47] views timing through wireless communication Key Performance Indicators (KPIs) such as time synchronization, bounded latency, and reliability. They further discuss next-generation technology features, such as Multi-Link Operation (MLO), which enables the concurrent use of multiple wireless frequencies to improve end-to-end latency.

Some studies focus on mitigating disruptions that negatively affect timing behavior. For example, in a predictive maintenance context, some faults can be mitigated through redundancy, robust protocols, and ana-

lytics [48]. Other studies address external disturbances, for instance, by intelligently predicting the wireless interference from attackers [49].

Taken together, these studies illustrate that timing in IoT is understood through a diverse set of perspectives.

Requirements: IoT requirements have been studied from a variety of perspectives. Within Requirements Engineering (RE), requirements are commonly categorized into functional and non-functional requirements [50], where the latter describe system-wide attributes such as safety, security, and reliability. These are particularly challenging in IoT settings due to the vast ecosystem and the interaction of multiple requirement dimensions, such as timing. Study [51] investigates model-driven RE and highlights the difficulty of lifecycle requirement management, noting that requirements must remain traceable across large, evolving deployments operating under diverse conditions.

From an architectural viewpoint, study [52] outlines high-level requirements for edge-enabled IoT applications, such as low latency, bandwidth efficiency, and reliability. Some domain-specific studies report stringent requirements. For example, an industrial automation study [53] identifies strict constraints on delay, communication range, energy usage, update frequency, and security for Wireless Sensor Actuator Networks (WSANs).

From a communication perspective, study [31] explores timing requirements in highly time-critical applications such as closed-loop control and mobile robotics by comparing achievable latency bounds of different wireless technologies under high reliability targets.

In addition, several standardization bodies report requirement profiles for specific domains or technologies [54], further illustrating the diversity and fragmentation of IoT requirements.

Measurement: Finally, how timing is measured and evaluated in IoT systems varies widely across studies, with a broad range of methods used. For instance, study [31] argues that simple latency statistics (e.g., min., max., avg.) may conceal extreme values, complicating the assessment of strict timing requirements. Some formal and experimental approaches of evaluating the WCRT are discussed in study [55].

From a communication perspective, numerous metrics are used from different perspectives and technologies, e.g., end-to-end latency over BLE [56], or unidirectional latency over Wi-Fi 6 and 5G [31]. Study [57] shows that commonly used link-quality metrics (e.g.,

PDR, Received Signal Strength Indicator (RSSI)) often need to be combined to accurately characterize wireless performance, which in turn affects timing.

At the device level, study [13] evaluates timing behavior in RTOSs, analyzing delay and jitter from context switching. Task-level timing is also studied; for example, [58] examines scheduling delays for bursty tasks in edge-enabled IoT systems. Scheduling surveys such as [59, 60] highlight deadline adherence, delay distributions, and scheduling latency as central metrics.

Specialized measurement metrics have also been proposed, e.g., study [61], which estimates the number of connection events required for a successful transmission.

This diversity underscores the need for a unified characterization of how timing in IoT systems.

4.2 On leveraging the edge and data reduction

Architecture: Timeliness is shaped by system-level design choices, and edge computing is frequently proposed as an effective approach. For instance, study [52] argues that utilizing edge computing closer to where the data is being produced supports low-latency applications, noting that neither cloud- nor device-centric architectures can consistently meet the demands of time-critical IoT scenarios. This has been evaluated in several contexts, such as within IIoT and safety-critical scenarios [23].

The question of how to utilize the edge effectively has been explored from multiple angles. Interoperability is a concern in many IoT deployments, where heterogeneous devices must collaborate toward a common goal. Study [45] argues that semantic integration at the edge enables more timely and coherent decision-making in mission-critical scenarios. Other studies examine how to use the edge-cloud continuum efficiently through task allocation, distribution, and scheduling. For example, study [62] offloads edge tasks to idle end-nodes in a reverse-offloading manner, while study [23] leverages SDN to slice bandwidth according to the latency sensitivity of queued traffic in edge-cloud environments.

Overall, these studies highlight the central role of architectural design and edge-centric strategies in enabling timely IoT operations.

Data reduction: Another important use case for the edge is intelligent data reduction. Prior studies have explored data-reduction strategies from multiple angles. Surveys have identified between three [63]

and fourteen [64] different data reduction techniques. Processing data locally at the edge can improve privacy by limiting data exposure during transmission, although additional safeguards such as encryption remain necessary. Common reduction techniques rely on filtering, aggregation, prediction, or compression. Data filtering transmits values only when they exceed a threshold deviation; data aggregation combines multiple readings into summarized values; and prediction forecasts future values based on historical patterns using statistical or Machine Learning (ML) models. These techniques are inherently lossy, in contrast to compression methods, which may be lossy or lossless (with lossless variants requiring more computational resources).

Reduction strategies are often described as single-point or multi-point, depending on whether they operate at one system layer or across several [64]. Single-point approaches, typically applied at the edge, dominate the literature. More advanced strategies, such as study [65], propose a dual-point reduction-and-reconstruction scheme with concept drift detection, incorporating a reconstruction phase at the cloud layer using predictive or ML-based estimators. Study [66] proposes dynamically switching between reduction algorithms at run time based on data characteristics to reduce delays.

Taken together, these studies highlight the diversity of edge-based data reduction strategies and their role in balancing timeliness, resource efficiency, and data fidelity.

4.3 On IoT security

The security landscape of IoT systems is broad, and many Commercial Off-The-Shelf (COTS) devices still lack basic protections aligned with CIA principles. Study [25] surveys IoT threats, identifying 31 passive and active attacks across system layers, many of which can be mitigated through encryption, authentication, and integrity checks.

Security challenges arise from multiple perspectives. Architecturally, processing data at an on-premise edge layer can improve control and reduce exposure [52]. At the application level, study [67] demonstrates how unencrypted MQTT traffic enables session hijacking, emphasizing proper configuration and lightweight encryption.

From a cryptographic standpoint, traditional public-key methods may be unsuitable for highly constrained devices [68]. However, modern protocols such as TLS 1.3 with elliptic-curve cryptography reduce overhead, enabling broader feasibility. Still, trade-offs exist: study [69]

reports higher overhead for TLS on NB-IoT devices, while simpler symmetric approaches lack key security properties.

At the transport layer, QUIC [27] offers reduced latency and faster secure connection establishment, making it attractive for time-critical IoT. Subsequent evaluations [70] highlight its potential through various comparisons. More advanced architectures, such as [71], combine QUIC with session resumption mechanisms and SDN to reduce latency, but introduce additional security considerations related to key management and trust.

Taken together, these studies highlight that IoT security remains a continual cat-and-mouse game, requiring careful trade-offs between protection, performance, and usability.

Chapter 5

Summary and future work

This thesis explored timing and security in IoT systems, with a focus on characterization and edge-centric approaches. To answer RQ1, the research began with establishing a foundational characterization of "timing", grounded in definitions, requirements, and measurement from a broad set of studies and one survey.

This yielded two timing definitions in IoT, time-criticality and predictability, each supported by four characteristics. We further provide a dataset of IoT requirements spanning six domains and 19 applications, and consolidate communication requirements for three of the seven identified categories with estimated 90th-percentile bounds. In addition, we identify a rich data set of timing metrics in the literature, categorized across five IoT system layers and five metric types. Overall, this foundation reveals how fragmented reference models lead to a scattered understanding of timing and measurement practices in IoT, while also exposing gaps in requirements, including security, energy, and non-cellular deployments. In doing so, it provides a structured answer to RQ1 by characterizing how timing is defined, measured, and specified throughout the IoT literature.

From this foundation, we explored two promising edge-centric approaches. To address RQ2, the first approach involved a practical comparison of three data reduction techniques across three scenarios, using accuracy and reduction metrics, demonstrating that algorithm choice and context-aware tuning are essential to meet application requirements. To address RQ3, the second approach focused on a latency-centric evaluation of MQTT over TCP and QUIC across different platforms, configurations, and traffic patterns. The results show that the primary cost of security arises during connection establishment, where factors such

as transport protocol, authentication scheme, key size, and endpoint capability significantly influence latency, while steady-state encryption overhead is relatively stable. Overall, these findings highlight the inherent trade-offs between efficiency, accuracy, and secure communication, requiring context-aware design choices across the edge-cloud continuum.

5.1 Future work

Future research directions are manifold. The following discussion highlights several directions related to improving timing and security in IoT systems, organized into three main areas: modeling and AI, security, and networking.

Modeling and AI: A key direction is the development of models that explicitly capture both timing and security requirements. For example, extending the Asset Administration Shell (AAS) with such attributes could improve its use in time-critical IoT scenarios [72]. System requirements may also be formalized using ontologies, enabling structured representation and reasoning about constraints. In addition, formal methods offer complementary ways to model timing, uncertainty, concurrency, and adversarial behavior.

Translating complex and often vague standards into concrete requirements remains challenging [73]. Large Language Models (LLMs), particularly when combined with techniques such as Retrieval-Augmented Generation (RAG), offer a promising approach by grounding models in standard documents, helping developers derive consistent and context-specific requirements.

Security: Lightweight security mechanisms remain critical for resource-constrained devices. Remote attestation schemes can provide guarantees of device identity and integrity [74]. Improvements in certificate management and transport-layer mechanisms (e.g., mutual TLS (mTLS), QUIC optimizations) require careful evaluation of latency–security trade-offs. Beyond traditional encryption, techniques such as steganography, physical layer security, and covert communication offer alternative ways to protect or conceal communication [75].

Networking: Programmable approaches such as SDN, P4-enabled devices, and In-band Network Telemetry (INT) enable fine-grained control and observability [23, 76]. Adaptive communication strategies, including active probing, can improve responsiveness to changing conditions [77]. Further investigation of transport protocols such as QUIC may reduce latency, while link-layer scheduling presents opportunities to jointly optimize timing and security.

Bibliography

[1] M. Weyrich and C. Ebert, "Reference Architectures for the Internet of Things," *IEEE Softw.*, vol. 33, no. 1, pp. 112–116, Dec. 2016.

[2] ISO/IEC, "Internet of Things (IoT), Preliminary Report 2014," *iso.org*, 2014. [Online]. Available: https://www.iso.org/files/live/sites/isoorg/files/developing_standards/docs/en/internet_of_things_report-jtc1.pdf. (accessed 2026-02-24).

[3] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in *Proc. 2015 Workshop Mobile Big Data*, 2015, pp. 37–42.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. and Tut.*, vol. 17, no. 4, pp. 2347–2376, Jun. 2015.

[5] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally Distributed Content Delivery," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 50–58, Oct. 2002.

[6] S. Leclerc, A. Bucaioni, and M. Ashjaei, "Characterizing Time-Critical Internet of Things," *Internet Things*, vol. 34, p. 101721, Nov. 2025.

[7] ——, "Systematic Consolidation of IoT Communication Requirements Across Domains," , 2026.

[8] D. L. Gomez, G. A. Montoya, C. Lozano-Garzon, and Y. Donoso, "Strategies for Assuring Low Latency, Scalability and Interoperability in Edge Computing and TSN Networks for Critical IIoT Services," *IEEE Access*, vol. 11, pp. 42 546–42 577, 2023.

[9] A. Seferagić, J. Famaey, E. De Poorter, and J. Hoebeke, "Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 488, Jan. 2020.

[10] M. Fahmideh, A. Ahmad, A. Behnaz, J. Grundy, and W. Susilo, "Software Engineering for Internet of Things: The Practitioners' Perspective," *IEEE Trans. Softw. Eng.*, vol. 48, no. 8, pp. 2857–2878, Aug. 2022.

[11] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 4, pp. 2489–2520, Jul. 2020.

[12] Ehrlich, Marco and Trsek, Henning and Wisniewski, Lukasz and Jasperneite, Jürgen, "Survey of Security Standards for an Automated Industrie 4.0 Compatible Manufacturing," in *Proc. IEEE IECON*, 2019, pp. 2849–2854.

[13] I. Ungurean, "Timing Comparison of the Real-Time Operating Systems for Small Microcontrollers," *Symmetry*, vol. 12, no. 4, p. 592, Apr. 2020.

[14] W. Stallings and C. Beard, *Wireless Communication Networks and Systems, First Edition.* Hoboken, NJ, USA: Pearson, 2016.

[15] A. Pathan, H.-W. Lee, and C. S. Hong, "Security in Wireless Sensor Networks: Issues and Challenges," in *Proc. 8th Int. Conf. Adv. Commun. Technol.*, 2006, pp. 6 pp.–1048.

[16] S. Leclerc, E. Hansen, A. Bucaioni, and M. Ashjaei, "Reducing IoT Data at the Edge: A Comparative Evaluation," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, 2026, pp. 1–6.

[17] Ericsson, "ICT energy evolution: Telecom, data centers, and AI," *ericsson.com*, 2025. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/ict-energy-evolution-telecom-data-centers-and-ai. (accessed 2026-05-19).

[18] S. Leclerc, A. Bucaioni, and M. Ashjaei, "On the Latency of Secure MQTT over QUIC vs. TCP: An experimental evaluation," in , 2026.

[19] G. C. Buttazzo, *Hard Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications, Third Edition.* New York, NY, USA: Springer, 2011.

[20] I. Behnke, L. Pirl, L. Thamsen, R. Danicki, A. Polze, and O. Kao, "Interrupting Real-Time IoT Tasks: How Bad Can It Be to Connect Your Critical Embedded System to the Internet?" in *Proc. IEEE 39th Int. Perform. Comput. Commun. Conf.*, 2020, pp. 1–6.

[21] S. Bhandari and X. Wang, "Prioritized Clock Synchronization for Event Critical Applications in Wireless IoT Networks," *IEEE Sensors J.*, vol. 19, no. 16, pp. 7120–7128, Aug. 2019.

[22] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," Internet Requests for Comments, RFC Editor, RFC 9000, May. 2021. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9000.html

[23] F. Habeeb, K. Alwasel, A. Noor, D. N. Jha, D. AlQattan, Y. Li, G. S. Aujla, T. Szydlo, and R. Ranjan, "Dynamic Bandwidth Slicing for Time-Critical IoT Data Streams in the Edge-Cloud Continuum," *IEEE Trans. Ind. Inform.*, vol. 18, no. 11, pp. 8017–8026, Nov. 2022.

[24] E. W. Dijkstra, *Chapter I: Notes on Structured Programming.* GBR: Academic Press Ltd., 1972.

[25] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE Commun. Surv. and Tut.*, vol. 22, no. 1, pp. 616–644, Nov. 2020.

[26] C. Alcaraz and S. Zeadally, "Critical Infrastructure Protection: Requirements and Challenges for the 21st Century," *Int. J. Crit. Infrastruct. Prot.*, vol. 8, pp. 53–66, Jan. 2015.

[27] A. Langley et al., "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Proc. ACM SIGCOMM Conf.*, 2017, pp. 183–196.

[28] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," Internet Requests for Comments, RFC Editor, RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8446.html

[29] A. Langley et al., "The QUIC Transport Protocol: Design and Internet-Scale Deployment," in *Proc. of the Conf. of the ACM Special Interest Group on Data Commun.*, 2017, pp. 183–196.

[30] E. Shahri, P. Pedreiras, and L. Almeida, "End-to-End Response Time Analysis for RT-MQTT: Trajectory Approach versus Holistic Approach," in *Proc. 19th Int. Conf. Factory Commun. Syst.*, 2023, pp. 1–8.

[31] K. Bhimavarapu, Z. Pang, O. Dobrijevic, and P. Wiatr, "Unobtrusive, Accurate, and Live Measurements of Network Latency and Reliability for Time-Critical Internet of Things," *IEEE Internet Things Mag.*, vol. 5, no. 3, pp. 38–43, Sep. 2022.

[32] S. Cronholm and H. Göbel, "Action design research: integration of method support," *Int. J. Manag. Proj. Bus.*, vol. 15, no. 8, pp. 19–47, Apr. 2022.

[33] B. Kitchenham and P. Brereton, "A systematic review of systematic review process research in software engineering," *Inf. and Softw. Technol.*, vol. 55, no. 12, pp. 2049–2075, Dec. 2013.

[34] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. 18th Int. Conf. Eval. and Assessment Softw. Eng.*, 2014, pp. 1–10.

[35] J. S. Molléri, K. Petersen, and E. Mendes, "Survey Guidelines in Software Engineering: An Annotated Review," in *Proc. 10th ACM/IEEE Int. Symp. Empirical Softw. Eng. and Meas.*, 2016, pp. 1–6.

[36] *Designing an effective survey*, 1st ed., Carnegie Mellon Softw. Eng. Inst., Pittsburgh, PA, USA, 2005.

[37] P. Brereton, A. B. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. and Softw.*, vol. 80, no. 4, pp. 571–583, Apr. 2007.

[38] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, M. Brennan *et al.*, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *BMJ*, vol. 372, p. n71, Mar. 2021.

[39] N. B. Ali and K. Petersen, "Evaluating strategies for study selection in systematic literature studies," in *Proc. 8th ACM/IEEE Int. Symp. Empirical Softw. Eng. and Meas.*, 2014, pp. 1–4.

[40] K. Charmaz and L. L. Belgrave, "Grounded Theory," in *The Blackwell Encyclopedia of Sociology*. Hoboken, NJ, USA: Wiley, 2007.

[41] D. S. Cruzes and T. Dyba, "Recommended Steps for Thematic Synthesis in Software Engineering," in *Int. Symp. Empirical Softw. Eng. and Meas.*, 2011, pp. 275–284.

[42] R. Franzosi, *Quantitative Narrative Analysis*. Newbury Park, CA, USA: Sage, 2010.

[43] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, and J. Popay, "Testing Methodological Guidance on the Conduct of Narrative Synthesis in Systematic Reviews: Effectiveness of Interventions to Promote Smoke Alarm Ownership and Function," *Evaluation*, vol. 15, no. 1, pp. 49–73, Jan. 2009.

[44] I. Behnke and H. Austad, "Real-Time Performance of Industrial IoT Communication Technologies: A Review," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7399–7410, Mar. 2024.

[45] A. Soularidis, K. I. Kotis, and G. A. Vouros, "Real-Time Semantic Data Integration and Reasoning in Life-and Time-Critical Decision Support Systems," *Electronics*, vol. 13, no. 3, p. 526, Jan. 2024.

[46] T. Mitra, J. Teich, and L. Thiele, "Time-Critical Systems Design: A Survey," *IEEE Des. and Test*, vol. 35, no. 2, pp. 8–26, Apr. 2018.

[47] J. Perez-Ramirez, O. Seijo, and I. Val, "Time-Critical IoT Applications Enabled by Wi-Fi 6 and Beyond," *IEEE Internet Things Mag.*, vol. 5, no. 3, pp. 44–49, Sep. 2022.

[48] K. S. Kiangala and Z. Wang, "An Effective Communication Prototype for Time-Critical IIoT Manufacturing Factories Using Zero-Loss Redundancy Protocols, Time-Sensitive Networking, and Edge-Computing in an Industry 4.0 Environment," *Processes*, vol. 9, no. 11, p. 2084, Nov. 2021.

[49] H. A. Bany Salameh, M. H. Khadr, M. Al-Quraan, M. Ayyash, H. Elgala, and S. Almajali, "Jamming-Aware Simultaneous Multi-Channel Decisions for Opportunistic Access in Delay-Critical IoT-

Based Sensor Networks," *IEEE Sensors J.*, vol. 22, no. 3, pp. 2889–2898, Feb. 2022.

[50] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," in *Proc. Conf. Future Softw. Eng.*, 2000, pp. 35–46.

[51] M. R. Tabassum and S. Mustafiz, "Advances in Requirements Engineering for IoT and CPS: A Survey on Model-Driven Research and Practices," in *Proc. IEEE/ACM 7th Int. Workshop Softw. Eng. Res. Pract. IoT*, 2025, pp. 40–47.

[52] C. C. Byers, "Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, Aug. 2017.

[53] J. Åkerberg, M. Gidlund, and M. Björkman, "Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation," in *Proc. 9th IEEE Int. Conf. Ind. Informatics*, 2011, pp. 410–415.

[54] A. Kunz, L. Nkenyereye, and J. Song, "5G Evolution of Cellular IoT for V2X," in *Proc. IEEE Conf. Standards Commun. Netw.*, 2018, pp. 1–6.

[55] M. Becker, D. Dasari, and D. Casini, "On the QNX IPC: Assessing Predictability for Local and Distributed Real-Time Systems," in *Proc. IEEE 29th Real-Time and Embedded Technol. and Appl. Symp.*, 2023, pp. 289–302.

[56] Y. Li, J. Lv, B. Li, and W. Dong, "RT-BLE: Real-time Multi-Connection Scheduling for Bluetooth Low Energy," in *Proc. IEEE Conf. Comput. Commun.*, 2023, pp. 1–10.

[57] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos, "Assessing link quality in IEEE 802.11 Wireless Networks: Which is the right metric?" in *Proc. 19th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2008, pp. 1–6.

[58] C. Zhang, F. Shen, J. Jin, and Y. Yang, "Fog Network Task Scheduling for IoT Applications," in *Proc. 2nd Workshop Fog Comput. and IoT*, 2020, pp. 10:1–10:9.

[59] S. Kharb and A. Singhrova, "A Survey on Network Formation and Scheduling Algorithms for Time Slotted Channel Hopping in Industrial Networks," *J. Netw. and Comput. Appl.*, vol. 126, pp. 59–87, Jan. 2019.

[60] S. Abolhassani Khajeh, M. Saberikamarposhti, and A. M. Rahmani, "Real-Time Scheduling in IoT Applications: A Systematic Review," *Sensors*, vol. 23, no. 1, p. 232, Dec. 2022.

[61] M. Spörk, C. A. Boano, and K. Römer, "Improving the Timeliness of Bluetooth Low Energy in Dynamic RF Environments," *ACM Trans. Internet Things*, vol. 1, no. 2, pp. 1–32, Apr. 2020.

[62] Y. Kim, C. Song, H. Han, H. Jung, and S. Kang, "Collaborative Task Scheduling for IoT-Assisted Edge Computing," *IEEE Access*, vol. 8, pp. 216 593–216 606, Dec. 2020.

[63] D. Kreković, P. Krivić, I. Podnar Žarko, M. Kušek, and D. Le-Phuoc, "Reducing communication overhead in the IoT-edge-cloud continuum: A survey on protocols and data reduction strategies," *Internet of Things*, vol. 31, p. 101553, May. 2025.

[64] L. Pioli, C. F. Dorneles, D. D.J de Macedo, and M. A. R. Dantas, "An overview of data reduction solutions at the edge of IoT systems: a systematic mapping of the literature," *Computing*, vol. 104, pp. 1867–1889, Mar. 2022.

[65] H. Zhang, J. Na, and B. Zhang, "Autonomous Internet of Things (IoT) Data Reduction Based on Adaptive Threshold," *Sensors*, vol. 23, p. 9427, Nov. 2023.

[66] A. Papageorgiou, B. Cheng, and E. Kovacs, "Real-time data reduction at the network edge of Internet-of-Things systems," in *Proc. 11th Int. Conf. Netw. Serv. Manag.*, 2015, pp. 284–291.

[67] T. K. Boppana and P. Bagade, "Security Risks in MQTT-based Industrial IoT Applications," in *Proc. IEEE Int. Conf. Omni-layer Intell. Syst.*, 2022, pp. 1–5.

[68] A. Aly, F. Khomh, and Y.-G. Guéhéneuc, "Enforcing Security in Internet of Things Frameworks: A Systematic Literature Review," *Internet Things*, vol. 6, p. 100050, Jun. 2019.

[69] M. Has, M. Kušek, and I. P. Žarko, "Evaluating Secure Variants of the MQTT Protocol on Resource-Constrained Devices for Precision Agriculture," in *Proc. IEEE Int. Conf. Telecommun. (ConTEL)*, 2023, pp. 1–7.

[70] L. Almeida, R. P. Pinto, and B. M. C. Silva, "Implementation and Performance Analysis of QUIC Protocol in IoT Monitoring Networks," in *Proc. IEEE Int. Mediterranean Conf. Commun. Netw.*, 2025, pp. 1–6.

[71] G. Pettorru and M. Martalà, "A Persistent and Secure Publish-Subscriber Architecture for Low-Latency IoT Communications," *IEEE Trans. Netw. Serv. Manag.*, vol. 23, pp. 729–739, Nov. 2026.

[72] X. Ye and S. H. Hong, "Toward Industry 4.0 Components: Insights Into and Implementation of Asset Administration Shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, Mar. 2019.

[73] Q. Rouland, S. Gjorcheski, and J. Jaskolka, "Eliciting a Security Architecture Requirements Baseline from Standards and Regulations," in *Proc. IEEE Int. Requirements Engineering Conf. Workshops (REW)*, 2023, pp. 224–229.

[74] J. Cao, T. Zhu, R. Ma, Z. Guo, Y. Zhang, and H. Li, "A Software-Based Remote Attestation Scheme for Internet of Things Devices," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1422–1434, Feb. 2023.

[75] X. Chen, J. An, Z. Xiong, C. Xing, N. Zhao, F. R. Yu, and A. Nallanathan, "Covert Communications: A Comprehensive Survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1173–1198, Apr. 2023.

[76] R. Ben Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, "PINT: Probabilistic In-band Network Telemetry," in *Proc. ACM SIGCOMM*, 2020, pp. 662–680.

[77] M. Spörk, M. Schuß, C. A. Boano, and K. Römer, "Ensuring End-to-End Dependability Requirements in Cloud-based Bluetooth Low Energy Applications," in *Proc. Int. Conf. Embedded Wireless Syst. and Netw.*, 2021, pp. 55–66.