Distributed Scheduling Algorithms for LoRa-based Wide Area Cyber-Physical Systems

Laksh Bhatia
Imperial College London



Imperial College London

Distributed Scheduling Algorithms for LoRa-based Wide Area Cyber-Physical Systems

Laksh Bhatia

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of
Imperial College London.

Department of Computing
Imperial College London

July 2022

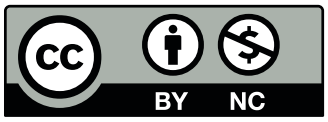
Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.



This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License.

http://creativecommons.org/licenses/by-nc/4.0/

© 2022 Laksh Bhatia

Statement of Originality

This is to certify to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes. I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

London, July 2022 L. J. B.

Acknowledgements

Deciding to pursue a PhD, actually pursuing it and finishing it is a task that a single person cannot accomplish alone. It requires a team of people, and a firm determination, to achieve this feat. For me, my group of people was exciting and inspiring. I would first like to thank my advisor Julie for all the support, discussions, and the freedom to explore ideas.

Next, I would like to thank my colleagues Mike, Po-Yu, Ivana, Cong, Mohammad, Youssef, Kaihan, Anqi, Michael, Pamela and Milan. I had countless discussions of life, the universe and everything, which helped me dig deeper into my ideas and become a better researcher. I would also like to thank David for being my internal advisor and for the thought-provoking discussions.

Finally, I would like to thank M for all of her help during this journey. I would not have made it through this process if it wasn't for her immense patience, belief and encouragement. And lastly, I would like to thank my mother, father and sister, Kritika, for their constant support and motivation. That's all folks!

L. J. B.

Abstract

Low Power Wide Area Networks (LPWAN) are a class of wireless communication protocols that work over long distances, consume low power and support low datarates. LPWANs have been designed for monitoring applications, with sparse communication from nodes to servers and sparser from servers to nodes. Inspite of their initial design, LPWANs have the potential to target applications with higher and stricter requirements like those of Cyber-Physical Systems (CPS). Due to their long-range capabilities, LPWANs can specifically target CPS applications distributed over a wide-area, which is referred to as Wide-Area CPS (WA-CPS). Augmenting WA-CPSs with wireless communication would allow for more flexible, low-cost and easily maintainable deployment. However, wireless communications come with problems like reduced reliability and unpredictable latencies, making them harder to use for CPSs.

With this intention, this thesis explores the use of LPWANs, specifically LoRa, to meet the communication and control requirements of WA-CPSs. The thesis focuses on using LoRa due to its high resilience to noise, several communication parameters to choose from and a freely modifiable communication stack and servers making it ideal for research and deployment. However, LoRaWAN suffers from low reliability due to its ALOHA channel access method. The thesis posits that "Distributed algorithms would increase the protocol's reliability allowing it to meet the requirements of WA-CPSs". Three different application scenarios are explored in this thesis that leverage unexplored aspects of LoRa to meet their requirements. The application scenarios are delay-tolerant vehicular networks, multi-stakeholder WA-CPS deployments and water distribution networks. The systems use novel algorithms to facilitate communication between the nodes and gateways to ensure a highly reliable system. The results outperform state-of-art techniques to prove that LoRa is currently under-utilised and can be used for CPS applications.

Contents

Copyright **2**

Statement of Originality **3**

Acknowledgements **4**

Abstract **5**

List of Figures **10**

List of Tables **13**

1 Introduction **14**

1.1 Low Power Wide Area Networks 15

1.2 Cyber-Physical Systems 16

1.3 Wireless communication and CPSs 18

1.4 Thesis Aim 19

1.5 Publications 21

1.5.1 Background Publications 21

1.5.2 Publications directly related to work of this thesis 23

1.6 Thesis's applications 24

1.7 Challenges and Contributions 25

1.8 Thesis Outline 29

2 Background to LoRa, LoRaWAN and Related Works **31**

2.1 LPWANs 32

2.2 Hardware . . . 32
2.2.1 Nodes . . . 33
2.2.2 Gateway . . . 34
2.3 LoRa Physical layer . . . 36
2.4 LoRaWAN . . . 39
2.4.1 Device Classes . . . 39
2.4.2 Reliability . . . 40
2.4.3 Architecture . . . 42
2.5 Related Work . . . 44
2.5.1 LoRa Physical Layer . . . 44
2.5.2 Increasing Physical Layer Decoding Rates . . . 45
2.5.3 Data-link Layer . . . 46
2.5.4 Network coordination . . . 48
2.5.5 Miscellaneous . . . 50
2.6 Summary . . . 53

3 Opportunistic Data Forwarding in Delay-tolerant Vehicular Networks 54
3.1 Problem Introduction . . . 55
3.2 Problem Definition . . . 56
3.2.1 System Model . . . 56
3.2.2 CA-ETX and challenges when applying in MLoRa-SG . . . 57
3.3 Opportunistic Data Forwarding with RCA-ETX . . . 58
3.3.1 Operation Overview of Data Forwarding with RCA-ETX . . . 58
3.3.2 Calculating RCA-ETX$_{x,\mathcal{G}}(t)$ . . . 60
3.3.3 Calculating RCA-ETX$_{x,y}(t)$ . . . 61
3.4 Real-time Opportunistic Backpressure Collection (ROBC) . . . 62
3.4.1 Backpressure routing . . . 62
3.4.2 Queue Dynamics . . . 63
3.4.3 ROBC Algorithm . . . 63
3.5 Customised Device Classes for Practical Operations . . . 65
3.6 Evaluation based on London Bus Network . . . 65
3.6.1 Experimental Setup . . . 66
3.6.2 Simulation Results . . . 69
3.7 Discussion, Limitations and Future Work . . . 75

3.8 Related Work . . . 76
3.9 Summary . . . 78

4 Increasing Reliability by Leveraging Overlapping Networks 80
4.1 Problem Introduction . . . 81
4.2 LoED dataset and Preliminaries . . . 83
4.2.1 LoED Dataset . . . 83
4.2.2 Exploiting Gateways from other Networks . . . 87
4.3 IRONWAN Overview . . . 89
4.3.1 IRONWAN Architecture . . . 90
4.3.2 Operation Overview . . . 91
4.3.3 Objective and Challenges . . . 93
4.4 RMIP Design . . . 95
4.4.1 Problem Statement . . . 95
4.4.2 Estimating $\Delta t'$ for missing messages . . . 96
4.4.3 Inter-Arrival Time Prediction . . . 97
4.4.4 Reference-Anchor Prediction . . . 98
4.4.5 Change Detection in Inter-Arrival Time . . . 98
4.5 Interference Prediction . . . 100
4.5.1 InterPred Overview . . . 100
4.5.2 State, Action, Policy, and Reward . . . 101
4.5.3 Agent Training based on Pseudo Actions . . . 103
4.5.4 InterPred Validation . . . 104
4.6 Implementation and Evaluation . . . 107
4.6.1 Testbed Evaluation . . . 108
4.6.2 Large-scale Simulation Evaluation . . . 108
4.7 Discussion, Limitations and Future Work . . . 117
4.8 Related Work . . . 118
4.9 Summary . . . 119

5 Control Communication Co-Design for slow-loop Critical Infrastructures 121
5.1 Problem Introduction . . . 122
5.2 System Overview . . . 124

5.2.1 System Architecture . . . 124
5.2.2 Co-design Constraints . . . 128
5.3 A LoRa data-link protocol for Control . . . 129
5.3.1 *Ctrl-MAC* Operating Principle . . . 129
5.3.2 Ctrl-MAC Implementation . . . 132
5.3.3 *Ctrl-MAC* Delay Analysis . . . 133
5.3.4 *Ctrl-MAC* and LoRaWAN . . . 136
5.4 Control Model for a WDN . . . 140
5.4.1 System Model & Controller Design . . . 140
5.4.2 Updating the Control Input . . . 141
5.4.3 Stability Analysis . . . 142
5.4.4 System Operation . . . 143
5.5 Implementation and Evaluation . . . 145
5.5.1 Evaluation via Co-Simulation . . . 146
5.5.2 Real World Evaluation . . . 151
5.5.3 Evaluation Conclusion . . . 154
5.6 Discussion, Limitations and Future Work . . . 154
5.7 Related Work . . . 156
5.8 Summary . . . 159

6 Conclusion and Future Work **161**
6.1 Discussion . . . 163
6.1.1 Metrics . . . 163
6.1.2 Cellular and LoRa . . . 164
6.2 Future Developments . . . 165

Bibliography **167**

A Appendix **191**
A.1 CPS Industries . . . 191
A.2 Proof of Corollary 1 (By Anqi Fu (co-author in [41])) . . . 194
A.3 Attributions . . . 196

List of Figures

1.1 Wireless communication technologies with datarate and range 15
1.2 Contributions of this thesis 26

2.1 LoRa node example 34
2.2 LoRa gateway architecture [60] 35
2.3 LoRa symbol examples 37
2.4 Time on air for different spreading factors with $CR = 1$ and $BW = 125$ [2] 38
2.5 Spectrum limits in the 868.0-870.0 MHz plan[63] 39
2.6 LoRaWAN Device Classes [2] 41
2.7 LoRaWAN architecture [2] 42

3.1 Compared to the network at $t = 0$, link $(x, y) \in \mathcal{L}$ regains communication (i.e. $c(t)_{x,y} > 0$), whereas device y loses its link with sinks $\mathcal{G}$ (i.e. $c(t)_{x,\mathcal{G}} = 0$) at $t = 1$. 57
3.2 An operational overview of our solution with LoRaWAN Class-C sensors, which are able to consistently overhear data packets from nearby neighbours). Device x decides not to send its packets via y when receiving its RCA-ETX sent at time t_0 and t_1, whereas x sends its packet to y at time t_2. 59
3.3 Node-to-gateway PST $\mu_{x,\mathcal{G}}(t)$ is estimated at every time t (Derived from [105]) 61

3.4 Modified Class-C is proposed to provide a reception feature required by node-to-node communications. Tx denotes the time slots reserved for device up-links. Rx1 denotes time slots for gateway acknowledgement via the same channel as Tx. Rx2 denotes the time slots for gateway down-links. . . . 65
3.5 This map shows the simulation area where bus routes are the lines with colors. . . . 66
3.6 London bus network's (a) number of active buses and (b) distribution of buses' active duration[113] over 24 hours. . . . 67
3.7 Average end-to-end delay with errors. . . . 70
3.8 Total network throughput . . . 71
3.9 Throughput in environments where node-to-node communication range is 500m. . . . 72
3.10 Throughput in environments where node-to-node communication range is 1000m. . . . 73
3.11 Average Number of hops for the entire network . . . 74
3.12 Average number of messages sent per node . . . 74

4.1 IRONWAN creates a wireless overlay network to link overlapping networks (networks 1 and 2) to exchange uplink and downlink transmissions. IRONWAN replays failed messages for uplink collisions(red colour) and can use the link for acknowledgements . . . 82
4.2 (a) Distribution of spreading factor at each gateway (b) Distribution of message types at each gateway (c) Distribution of frequencies at each gateway . . . 88
4.3 Effect of partitioning in a LoRaWAN network . . . 89
4.4 Interaction between LoRa radio, packet forwarder and IRONWAN's modules (shown in colour) . . . 90
4.5 (a) All received messages are replicated by manager and forwarded to all modules, (b) IRONWAN's G2G communication for a gateway to request a lost uplink message from a neighbour gateway. (c) IRONWAN's G2G communication for a gateway with no remaining downlink duty-cycle to request that a neighbour gateway send a downlink message. . . . 92

4.6 Examples of: (a) uplink messages sent and received by a node and the gateway, respectively, without retries. (b) uplink messages sent by a node with retries. (c) uplink messages received at the gateway with retries. . . . 96
4.7 The precision and recall of the change detection in RMIP given given different window size n (5-15) and error threshold e (0.5s-2.0s) 99
4.8 InterPred overview 101
4.9 InterPred validation results 106
4.10 Testbed evaluation 109
4.11 Evaluation of LoRaWAN and IRONWAN with increasing load and number of gateways 111
4.12 Average number of retransmissions 112
4.13 Overhead for IRONWAN compared to LoRaWAN 112
4.14 Varying maximum number of retransmissions 114
4.15 Comparison of IRONWAN, FLIP and ideal centralised system 116

5.1 The water distribution network. 123
5.2 System architecture. 125
5.3 (a) Channel allocation and frame structure in request and data channels (b) The RRM structure for $k = 3$, $M_D = 3$ which implies that $size(C_2)$=$\log_2 M_D$, $l = 16$. 130
5.4 *Ctrl-MAC* (a) Channel disciplines comparison, (b) Request slots comparison. 133
5.5 The probability of achieving $t_{req} \leq x$ for M/M/1 queue. 135
5.6 (a) System response to constant customer demand, (b) Distribution of inter-arrival times of the events. 144
5.7 Daily customer demand (24 hours) 148
5.8 System response to a fault in the tank 1. 149
5.9 (a) Hardware-in-the-Loop experimental setup, (b) Hardware-in-the-Park experimental setup with up-stream delay and SNR values as seen at the gateway (the gateway is marked by the star, individual nodes locations by red circles). 153

List of Tables

2.1 LPWAN comparisons [55, 58] 33

4.1 Gateway data and location overview. 85

5.1 E2E PDR (%) of *Ctrl-MAC*/LoRaWAN/LoRaWAN++ under different data loads and data distributions. 138
5.2 E2E Delays (s) of *Ctrl-MAC*/LoRaWAN/LoRaWAN++ under different data loads and distributions 138
5.3 UL Reliability (%) of *Ctrl-MAC*/LoRaWAN/LoRaWAN++ for 10 and 200 nodes 139
5.4 The evaluation of the C^3 system for constant customer demand in comparison with an idealistic wired control and LoRaWAN++ for control. 147
5.5 Evaluation of the C^3 system for dynamic customer demand compared to idealistic wired control and LoRaWAN++ for control. 148
5.6 The evaluation of the C^3 system for constant customer demand with fault in comparison with an idealistic wired control and LoRaWAN for control. 150
5.7 The evaluation the C^3 system for 10 DMAs (32 nodes) in comparison with LoRaWAN++ for control. 151
5.8 Round Trip Times and Overshoot for Hardware-in-the-loop experiment. 153
5.9 Wireless CPS categories with hardware implementations 158

A.1 Vehicular networks wireless requirements[21] 192
A.2 Critical infrastructure wireless requirements[190] 194

1

Introduction

Wireless communication is the transfer of data using electromagnetic waves without wires. A significant amount of research has made this transfer reliable, work over all distances and secure. The past few decades have seen the development of several wireless communication technologies. Commonly used technologies (shown in Figure 1.1) are short-range technologies like Near Field Communication (NFC), Wireless Personal Area Networks (WPAN) like Bluetooth, IEEE 802.15.4, Wireless Local Area Networks (WLAN), like WiFi, WiMax, and long-range technologies like Cellular (2G, 3G, 4G, 5G), Low Power Wide Area Networks (LPWAN). These communication technologies offer different communication ranges, datarate, latency, and consume different amounts of energy, making them suited to different applications.

This thesis focuses on LPWANs, specifically LoRa, that are a relatively new paradigm in wireless communication systems. LPWANs are designed for long-range, low datarate and low-powered communication. They are designed as one-hop networks and generally accommodate relatively small packets of a few kilobytes. The initial use-case for all LPWANs was remote monitoring like smart sensors, which require sparse communication from nodes to servers and sparser communication from servers to nodes. These applications also had low-reliability requirements. Despite their initial design, LPWANs have the potential to be used in applications with stricter requirements than monitoring applications. We ask whether LPWAN's reliability operating in the unlicensed space can be improved and what new applications can they target. A class of applications with harsher requirements than monitoring are Cyber-Physical Systems (CPS). This thesis explores *the use of LPWANs for CPSs distributed over wide-areas.*

We begin the chapter by learning about LPWANs, CPSs and wireless communication in CPSs. We then discuss the aim of the thesis, publications, thesis's applications and

contributions.

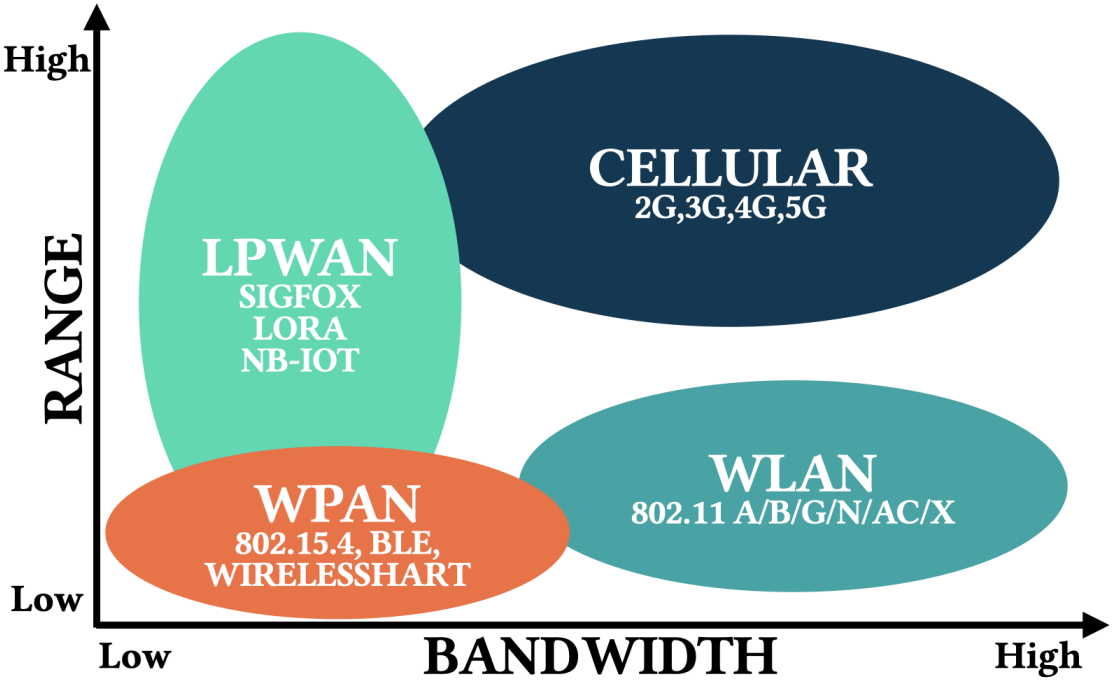


Figure 1.1: Wireless communication technologies with datarate and range

1.1 Low Power Wide Area Networks

Several LPWANs have been designed in the last decade[1]. Some of the prominent LPWAN technologies widely used in research and industry are LoRa/LoRaWAN[2], Sigfox[3] and NB-IoT[4]. LoRa is a physical layer patented by Semtech Corporation. LoRaWAN is the medium access protocol based on the Aloha channel access mechanism and runs on top of LoRa. NB-IoT is a technology standardized by the 3GPP and is a simplified version of the LTE standard designed for power-constrained devices. Sigfox is a carrier-controlled proprietary communication stack. It uses GFSK and DBPSK for its physical layer. LoRa and Sigfox both work in the unlicensed spectrum and NB-IoT works on the licensed spectrum. However, Sigfox is operator based and requires a company to manage the network making it difficult for ad-hoc deployments.

While LoRa's physical layer is proprietary, its communication stack is open, and that of Sigfox and NB-IoT are not. Because of this, Sigfox and NB-IoT do not provide the

flexibility to design protocols on their physical layers. LoRa, on the other hand, provides this flexibility by only patenting the physical layer and allowing the community to develop the protocols around it. LoRa is a chirp spread spectrum modulation making it relatively more resilient to noise and other environmental factors. LoRa, unlike most LPWANs, uses the same modulation for uplink and downlink communication, allowing for bi-directional communication amongst the nodes and gateways. LoRa's deployment models are also flexible that has led to its rapid adoption and usage in several application scenarios. LoRa's medium access protocol is LoRaWAN which is built on the ALOHA channel access scheme and standardised by the LoRa alliance. The ALOHA channel access scheme makes LoRaWAN unsuitable for most CPSs as it cannot provide the reliability or the guarantees needed for CPSs. For the above reasons, this thesis uses LoRa as the physical layer, and modifies and redesigns the systems on top of it to be suited for CPSs. A detailed description of LPWANs, LoRa and LoRaWAN can be found in Chapter 2. Next we discuss Cyber-Physical Systems.

1.2 Cyber-Physical Systems

A CPS is the merger of physical systems with the computing domain connected by a communications core[5]. CPSs are feedback systems that are networked and distributed, adaptive to changes in the physical and cyber systems, and intelligent[1]. A CPS consists of a physical layer that includes components like pipes, cables, robots, vehicles etc. and the cyber layer that consists of the digital computer system which takes data from the physical-layer and reasons and makes decisions to meet an application's requirements. In CPSs the *communication networks* play a central role as they connect the cyber and the physical layers which enables the control of physical systems' components. CPSs combine the research of several different areas. Some of the prominent ones are control systems, embedded systems, communication networks (wired and wireless), real-time systems and computing combined with modelling of physical systems and the requirements derived from the models.

A CPS is different from traditional computing systems because of the coupling between the cyber and physical layers. Physical processes are part of the feedback loop in CPSs[6]. Actions taken in the cyber layer affect the physical processes, and

[1] cyberphysicalsystems.org

changes in the physical process affect what resources are needed from the cyber layer. Physical systems connected to a cyber system is not a new phenomenon, e.g. internet-connected environmental sensors, meters, valves, assets etc., have all existed for a while under the IoT (IoT) umbrella or in control systems e.g. Supervisory control and data acquisition (SCADA) systems. However, these are generally one-to-one mappings where reading from one sensor affects one actuator or are sensor-only systems i.e. monitoring systems. CPS's envision a more complex approach where sensors, actuators and systems themselves are inter-connected and communicate with each other. In such a system, multiple sensors and actuators interact and information from one system affects another. E.g. a self-driving car is a system in itself with its own sensors and actuators, that communicates to other self-driving cars leading to a complex inter-connected systems-of-systems.

CPS is a vast domain with applications in several industries. The popular industries are robotics, smart buildings, transportation, precision agriculture, smart infrastructure, power grids, medical devices, critical infrastructure, aerospace[7, 8, 9, 10]. Because of the numerous industries and applications, all CPS applications have different requirements. The complex interactions required to meet these requirements make designing a CPS system non-trivial. An understanding of the application domain and all the physical and cyber components and how they need to interact is needed to design an appropriate solution. A core to all CPSs is the ability of the physical layer to exchange data with the cyber-layer to make control decisions that ensures system stability. This makes communication central to any CPS, and any failure to the communication system would lead to a failure of the application.

Most currently deployed CPSs like robots use wired communication systems[11, 12], which makes it difficult for them to be deployed in a city-scale area, be mobile or in hard-to-reach places and the deployment and management costs non-trivial. These are all essential features for applications in smart-city CPSs. A smart-city consists of several CPSs with 100s of different components, that can run on batteries, and are deployed over a city's geographical area. Designing these CPSs using wireless communications will allow for a low-cost, low-powered and easily maintainable system. In this thesis, we refer to CPSs distributed over a large geographical area as Wide-Area CPSs (WA-CPS). Some of the applications in smart-city CPS's are building automation, transportation networks and critical infrastructures. In the following sections, we describe the various

wireless systems being used in CPSs, the challenges of using LPWANs for WA-CPSs, the thesis's aim, research questions and contributions.

1.3 Wireless communication and CPSs

A tremendous amount of research is going on in enabling wireless communication in CPSs like aerospace[13], critical infrastructure[14], vehicular networks[15], precision agriculture[16, 17], factory automation, industrial automation, building automation[18].

Park et al.[18] perform a detailed analysis of wireless network and control systems co-design with motivating applications in the factory, industrial and building automation. The authors discuss the various control schemes like time-triggered, event-triggered and self-triggered controls and their co-design with various wireless protocols. Their survey focuses on WPAN and WLAN scale networks like 802.15.4, 802.11, and WirelessHART. The survey discusses the varied requirements for wireless-CPS applications and how different solutions need to be designed to meet these requirements. Park et al.[19] further discuss applications of WPAN and WLAN networks in wireless avionics within an aeroplane which would significantly reduce the weights, and in turn, save cost by 12%[20]. To guarantee deterministic performance, the authors propose that Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) and choosing the appropriate slot lengths to guarantee a stable system[19].

Another wireless CPS area actively being researched is vehicular networks[21]. The area of vehicular networks is growing under the Vehicular to everything (V2X) umbrella, which includes vehicle to vehicle (V2V), vehicle to infrastructure (V2I), vehicle to pedestrian (V2P), vehicle to device (V2D) and vehicle to grid (V2G). Cellular and WLAN networks are used to design V2X systems. However, it is dominated by the Cellular community, which has extended 4G and 5G networks to meet the requirements of vehicular networks[15]. WLAN systems like Dedicated short-range communications (DSRC) are also designed for vehicular networks[22]. These technologies target applications like cooperative awareness, basic safety, and decentralised environmental notification. These applications require high reliability, which the current communication system can meet. However, applications like self-driving have stringent closed-loop requirements which are currently being developed within Cellular and WLAN networks[15].

Most wireless protocols developed over the past few decades do not meet the

requirements for closed-loop CPSs. Only in the last few years are protocols emerging for wireless CPS applications. These protocols offer high reliability and availability by having complete control over the wireless environment, which is only feasible for cellular technologies or in highly controlled environments. Hence, the most widely deployed wireless CPSs can be found in factory automation systems like robotic systems. These robotic applications are deployed in controlled environments and generally require short-range communication. This makes WPAN and WLAN most suited for robotic applications. To design wireless protocols to meet the factory automation's requirements, there has been an effort to create standards like WirelessHART[23] and ISA 100.11a[24]. However, there has been a slow adoption of these technologies by industry. This stems from the lack of benefits of wireless communication in already existing deployments, a lousy perception of wireless communication and unaligned research and real-world goals[25]. To this end, wireless researchers are working towards designing benchmarking solutions[26, 27] that can be useful to compare wireless protocols and increase the confidence of the industry in wireless solutions.

This section discussed how different communication technologies are used for different applications. The applications and the wireless system need to be considered together to ensure system stability by designing appropriate communication protocols. With this intention, this thesis extends LPWAN's to meet the requirements of WA-CPSs.

1.4 Thesis Aim

This thesis posits that current LPWANs, particularly LoRa and LoRaWAN, can be great enablers for WA-CPSs. However, they fail to meet the requirements in their current form as they do not provide the required reliability level, nor can they guarantee end-to-end delays. One of the reasons for this shortcoming is the medium access protocol and the reliability mechanisms designed within the protocol. It is not surprising as LoRaWAN is designed for periodic or sporadic transfer from nodes to servers with a minimal reliability requirement. Clearly, these design choices make it harder for LoRaWAN to be used for CPS applications. Augmenting WA-CPSs with wireless communication would allow for remote management, control, low-cost and easily maintainable deployments. It would also allow for devices in inaccessible locations, or mobile to have the above features at low-power. With this intention, the aim of the thesis is to defend the

following:

Low datarate, long-range communication systems like LoRa can be used for wide-area CPSs by using distributed algorithms to appropriately exploit the communication links between the cyber and physical layers.

In low-datarate systems, the limited communicationn bandwidth needs to be exploited in an efficient manner to meet the requirements of a CPS. Along with their low-datarate, LoRa has a long communication range, and ad-hoc deployment model, which leads to a non-uniform deployment with overlapping communication zones. The thesis posits, that distributed algorithms are naturally suited for such a communication system instead of centralised ones. The algorithms need to account for LoRa's capabilities while making decisions about the data that needs to be exchanged in the network and how to appropriately schedule it. The scheduling schemes will increase the protocol's reliability in various scenarios allowing it to meet a CPS's requirements. Scheduling the communication links is a commonly used technique to ensure high reliability. Link scheduling for CPS applications is also not a new phenomenon. Most industrial wireless protocols operating in the 2.4GHz wireless spectrum like WirelessHART[23], Time Synchronised Channel Hopping[28], ISA 100.11a[24] all use scheduling schemes to guarantee stable CPS operations. All of these protocols use a synchronous communication protocol where nodes have a fixed transmission and reception schedule. These communication protocols can guarantee the desired reliability levels required for CPS applications because of their pre-determined communication schedules. This approach cannot always work in LPWANs as they are distributed over large areas and have relatively sparse communication bandwidth, so tools like time synchronisation that use a part of the limited bandwidth for synchronisation cannot be used in all cases. Communication protocols used in 2.4GHz work well in short to medium range scenarios with a relatively low number of nodes. In long-range scenarios with 1000s of nodes, naive link scheduling of allocating communication slots or expecting all nodes to transmit synchronously becomes difficult.

This is where distributed scheduling comes in play. Distributed scheduling relies on information exchange between the nodes or gateways to find optimal schedules locally. Additionally, autonomous scheduling algorithms allow the distributed scheduling systems to decide its transmission schedules by itself without additional communication

overhead[31]. There has been much research for distributed and autonomous scheduling algorithms in TSCH[31, 32] and WirelessHART[30] that run in the 2.4GHz wireless spectrum. Distributed scheduling is naturally suited for LPWANs as the gateways are distributed over a wide-area and handle 1000s of nodes each. Centralised scheduling would take hours, if not days, to find a globally optimal schedule[33]. The centralised algorithms would also not take into account the lossy, dynamic nature of the links. Distributed and autonomous scheduling would allow us to schedule links faster, taking into consideration local conditions.

With this intention, the work in this thesis looks at how distributed algorithms can be used to leverage LoRa' unique and under-utilized elements to allow LoRa to be used for application it was not originally designed for, specifically WA-CPSs. The thesis first identifies applications that current LoRa(WAN) cannot target. These applications' requirements are then used to design a system using LoRa as the physical layer. The system includes the design-time communication protocol and the run-time algorithms to make optimal usage of the communication protocol. The algorithms decide what data to transmit, how much to transmit, whom to transmit it to and when to transmit depending on the specific solution. This work would allow WA-CPSs to become efficient, reliable and provide guarantees for system stability that current state-of-art solutions cannot. The research outlined in this thesis should pave the way for LPWANs to be used in CPSs.

1.5 Publications

This section highlights the publications that I have been part of since I started my PhD. The first section lists the background publications, which helped me gauge the CPS space and help answer the first research question, *what WA-CPSs can LoRa be used for?*

1.5.1 Background Publications

While I was digging into the problems of WA-CPSs and LoRaWAN, I was part of several different projects. These projects helped me understand the broad space of CPSs and wireless communication. It helped me better understand that it is difficult to have *one fits all* solution as the application requirements and tools available can be quite diverse. The below list of publications use different wireless communication technologies like WiFi,

WirelessHART and 802.15.4, for varied applications like control of a water distribution network, a two-phase oil and gravity separator, data collection from unmanned aerial vehicles and from a rotating drum of a mine.

O1. H. Wang, G. Zhou, L. Bhatia, Z. Zhu, W. Li, and J. A. McCann. "Energy-Neutral and QoS-Aware Protocol in Wireless Sensor Networks for Health Monitoring of Hoisting Systems". In: *IEEE Transactions on Industrial Informatics* 16.8 (2020), pp. 5543–5553. DOI: 10.1109/TII.2020.2969218

O2. I. Tomić, M. J. Breza, G. Jackson, L. Bhatia, and J. A. McCann. "Design and Evaluation of Jamming Resilient Cyber-Physical Systems". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 687–694. DOI: 10.1109/Cybermatics_2018.2018.00138

O3. Laksh Bhatia, David Boyle, and Julie A. McCann. "Poster Abstract: Aerial Interactions with Wireless Sensors". In: *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. SenSys '18. Shenzhen, China: Association for Computing Machinery, 2018, pp. 373–374. ISBN: 9781450359528. DOI: 10.1145/3274783.3275189

O4. Michael Breza, Laksh Bhatia, Ivana Tomic, Anqi Fu, Waqas Ikram, Valentinos Kongezos, and Julie A McCann. "The Separator, a Two-Phase Oil and Water Gravity CPS Separator Testbed". In: *arXiv preprint arXiv:2002.00945* (2020)

O5. Mohammad Heggo, Laksh Bhatia, and Julie A McCann. "RFTacho: An OAM approach for non-intrusive RF monitoring of rotating machines". In: *Proceedings of the 21th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2022)*. IPSN '22. New York, NY, USA: Association for Computing Machinery, 2022

O6. Mohammad Heggo, Laksh Bhatia, and Julie A. McCann. "Poster Abstract: Cognisense: A Contactless Rotation Speed Measurement System". In: *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. SenSys '21. Coimbra, Portugal: Association for Computing Machinery, 2021, pp. 347–348.

ISBN: 9781450390972. DOI: 10.1145/3485730.3492879. URL: https://doi.org/10.1145/3485730.3492879

O7. Laksh Bhatia, Boon Wong, Yathreb Bouazizi, Michael Breza, and Julie A McCann. "SNIAPE: Sensor Network and IoT Application Performance Evaluation Benchmark". In: *(under review)* (2021)

The results of these publications outlined, how all of these applications are short-range, have relatively high datarates and so using LoRa might not be the best choice for them. With this information, we will discuss what applications are best-suited for LoRa in Section 1.6.

1.5.2 Publications directly related to work of this thesis

The below list contains the publications directly related for the work in this thesis:

U1. Laksh Bhatia, Ivana Tomić, Anqi Fu, Michael Breza, and Julie A. Mccann. "Control Communication Co-Design for Wide Area Cyber-Physical Systems". In: *ACM Transactions on Cyber-Physical Systems* 5.2 (Jan. 2021). ISSN: 2378-962X. DOI: 10.1145/3418528

U2. P. -Y. Chen, L. Bhatia, R. Kolcun, D. Boyle, and J. A. McCann. "Contact-Aware Opportunistic Data Forwarding in Disconnected LoRaWAN Mobile Networks". In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. 2020, pp. 574–583. DOI: 10.1109/ICDCS47774.2020.00110

U3. Laksh Bhatia, Po-Yu Chen, Michael Breza, Cong Zhao, and Julie A. McCann. "IRONWAN: Increasing Reliability of Overlapping Networks in LoRaWAN". in: *IEEE Internet of Things Journal* (2021), pp. 1–1. DOI: 10.1109/JIOT.2021.3125842

U4. Ivana Tomić, Laksh Bhatia, Michael J Breza, and Julie A McCann. "The Limits of LoRaWAN in Event-Triggered Wireless Networked Control Systems". In: *2018 UKACC 12th International Conference on Control (CONTROL)* (2018), pp. 101–106

U5. Laksh Bhatia, Michael Breza, Ramona Marfievici, and Julie A. McCann. "LoED: The LoRaWAN at the Edge Dataset: Dataset". In: *Proceedings of the Third Workshop on Data: Acquisition To Analysis*. DATA '20. Virtual Event, Japan: Asso-

ciation for Computing Machinery, 2020, pp. 7–8. ISBN: 9781450381369. DOI: 10.1145/3419016.3431491

U6. Laksh Bhatia, Ivana Tomić, and Julie A. McCann. "Poster Abstract: LPWA-MAC: A Low Power Wide Area Network MAC Protocol for Cyber-Physical Systems". In: *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. SenSys '18. Shenzhen, China: Association for Computing Machinery, 2018, pp. 361–362. ISBN: 9781450359528. URL: https://doi.org/10.1145/3274783.3275183

1.6 Thesis's applications

WA-CPSs are CPSs distributed over a wide geographical area. Current short-range communication technologies like WiFi, BLE, 802.15.4 are ill-suited to meet wide-area requirements without a massive cost and management overhead. As these communication systems can only communicate for 10s of meters, 100s of nodes communicating using multiple hops are needed to cover kilometre distances, which significantly increases deployment complexity in terms of management, having the right energy source and finding appropriate deployment locations. Control and management of these networks is also difficult because of the lossy channels and multi-hop topologies. There is a push to use Software Defined Networking (SDN) approaches to facilitate remote management, control and reconfiguration of these networks[47]. However, SDN approaches currently have only been tested on 10s of nodes questioning whether they would scale to 1000s of nodes needed to cover the large geographical areas. Traditional cellular networks from 2-5G are expensive to deploy, manage, and generally not low-powered. The update cycle of cellular technologies is prolonged, which also makes it harder to adapt to the new use-cases constantly emerging in this space. For this reason, this work explores whether LPWANs, specifically LoRa can meet the requirements of the chosen applications.

This thesis focuses on WA-CPSs like critical infrastructures and vehicular networks. There are several applications that fall under these two industries and a review of them can be found in Appendix A.1. The review shows that applications have different requirements in terms of datarate, latency, communication range and reliability. The conclusion of the review is not all of these requirements can be met by LPWANs. So, the applications that we focus on are ones with slow update intervals and delay tolerance.

Two applications from vehicular networks and critical infrastructures are chosen for Chapters 3 and 5. These applications have low sampling rates, are delay-tolerant and are distributed over a wide area. Chapter 3 deals with delay-tolerant vehicular networks , where data needs to be delivered with low-latencies and high-throughput. However, no disastrous results would happen if messages are lost. Chapter 5 deals with the control of slow-loop feedback loops like water distribution networks which require bounded delays, high reliability and failed messages could lead to tank overflows. The two applications show two aspects of delay-tolerant WA-CPSs, where one application requires high throughput, and the other requires high-reliability and bounded latencies. Another type of application that this thesis focuses in Chapter 4 are multiple overlapping networks. With the rising number of CPSs in a smart-city environment, it is expected that 100s of applications will be running simultaneously in each other's vicinity. These applications have varied requirements, deployed by different stakeholder and all of those need to be met by the communication network. Unlike applications from Chapter 3 and 5 where we look at a particular application, Chapter 4 improves reliabilities without any knowledge of the applications.

Next, we will discuss the specific contributions of this thesis, what kind of scheduling schemes are leveraged and how they are used to facilitate the various applications.

1.7 Challenges and Contributions

This thesis's contributions are around identifying unused resources given an application scenario and then designing distributed scheduling schemes to increase overall reliability and meet application requirements. The main contributions of this thesis, also shown in Figure 1.2, are described below.

Chapter 3: Opportunistic data forwarding in delay-tolerant vehicular networks
Vehicular networks or transportation networks are applications designed to improve a user's experience while commuting. Vehicular networks includes elements like self-driving cars, buses, trains, pedestrians and traffic signals. There are several sub-applications under vehicular networks and they all suffer from different challenges. This chapter focuses on delay-tolerant applications like near real-time information of vehicles and roads, weather reports, data and control of micro-mobility vehicles where reliable delivery information will improve the user's overall experience and also the

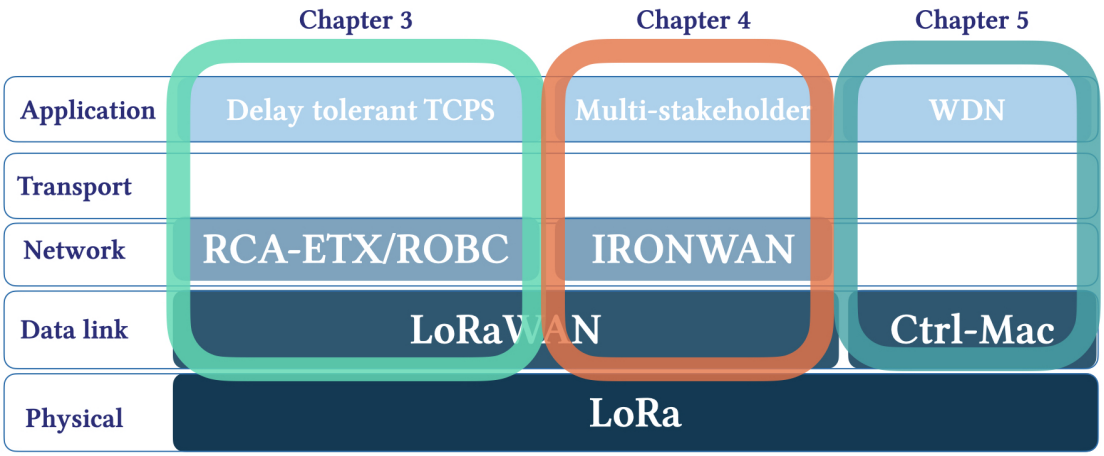


Figure 1.2: Contributions of this thesis

traffic flow on the roads. This application is delay-tolerant and requires high throughput, so the cyber-layer has enough information to make decisions for the physical-layer.

In this application, the nodes are *mobile* and need to deliver data to servers with low latency. LoRaWAN is the most suited wireless communication technology to be used in these applications due to its long-range free-to-use spectrum. The problem is that the communication range for LoRa is limited in urban scenarios because of multi-path fading and obstacles[48]. This implies a high-density of gateways are needed to cover an entire city. It is estimated that if every gateway covers approximately a 2 kilometre communication area, more than 130 gateways would be needed to cover the total area of London when the gateways are placed equidistant from each other without any overlapping areas. This makes the cost of deploying gateways non-trivial and so it is infeasible to expect high-density deployments to meet the communication requirements. With a low-density of gateways, the current LoRaWAN protocol fails to meet the reliability requirements as the nodes cannot communicate with the gateways. Chapter 3 proposes scheduling *node-to-node* (or V2V) communication between the moving vehicles to meet the requirements of *delay-tolerant reliable communication*.

In this node-to-node communication scheme, the nodes try to communicate directly with gateways and on failure, hand over the data to nearby nodes that have a better link quality to the gateways. The nodes work in an autonomous way to decide how much data to transmit to which nodes. This node-to-node opportunistic scheme increases the network throughput by upto 53% and reduces the end-to-end delays by upto 25%

compared to baseline LoRaWAN, which allows the network to have more data about its vehicles and road conditions. This work shows that opportunistic networking is a viable solution to unreachable links because of low-density gateways, fading, and obstacles in mobile LoRaWAN.

Chapter 4: Increasing Reliability by Leveraging Overlapping Networks
The second application studied in this thesis is multi-stakeholder reliable CPS applications. LoRaWAN in an urban area is following an ad-hoc deployment model where anyone can deploy their own LoRaWAN network as long as they adhere to the duty-cycle regulations. This has led to several simultaneously running applications with overlapping communication range and no knowledge of other networks.
Most of the currently deployed LoRaWAN applications have minimal reliability requirements. However, when CPS applications are run using these same resources, the reliability requirements will increase, increasing the load on the wireless spectrum and requiring better resource management schemes.

With this intention, the work in this chapter looks at multi-stakeholder applications with a mixture of typical sensing and CPS-like applications. These applications target the class of reliable sensing applications that require *reliable* communication, need to be *energy-efficient*, but most of all the system needs to be *adaptable* to the changing demands like addition of new nodes and varying transmission periods. Reliable sensing and actuation is an essential requirement of all CPS applications where sensor data needs to be delivered reliably to the *cyber-layer* and the actuation data sent back down.

LoRa's ad-hoc deployment model is unlike what has been seen in 802.11, Bluetooth and 802.15.4 where networks use overlapping frequencies. These short-range systems have traditionally tried to avoid interference by moving to different frequencies or using Carrier Sense Multiple Access (CSMA). However, this is not possible in LoRa as it uses multiple frequencies to enable scalability. So instead of trying to avoid frequencies in overlapping networks, this work leverages the overlapping networks to increase the reliability for all participating nodes. This work is to the best of our knowledge, the first of its kind to identify the opportunity with overlapping LoRaWAN networks and use the wireless spectrum to facilitate gateway-to-gateway communication. The gateway-to-gateway wireless communication is used to increase the uplink and downlink reliabilities by requesting for failed uplink messages and offloading excess downlink messages. The gateways use lightweight learning schemes to learn the node's

transmission period, and request failed messages from neighbouring gateways. They also handover downlink messages that they cannot handle to neighbouring gateways. The gateways work autonomously to decide when and what to transmit. This work's main contributions are lightweight learning schemes that can run on gateways and scheduling gateway-to-gateway communication with minimal interference to the already existing communication infrastructure. The algorithms run on the gateways and are completely distributed. The results show that our system reduces energy consumption of nodes by upto 50% by reducing the number retransmissions and increasing the minimum packet delivery ratio by upto 28% compared to baseline LoRaWAN.

Chapter 5: Control Communication Co-Design for slow-loop critical infrastructures

Slow-loop critical infrastructure is the last application of this thesis. This work specifically targets closed-loop control of water distribution networks (WDN). This application has stricter requirements compared to vehicular networks and reliable CPSs. It requires *high reliability*, *bounded delays* and be able to deal with node *adaptability*, needed to guarantee stability for any control system. WDNs are slow-feedback loop critical infrastructures because of their low sampling intervals which is a result of the system's physical components. The sensors can also be deployed in inaccessible areas where mains-power may not be available and so sensors need to be battery-powered. This slow-feedback loop and low-power operation is why LoRa is the most-suited communication technology for this class of application.

Due to the tight application requirements, the control and communication systems need to be co-designed. Chapter 5 proposes to the best of our knowledge, the first closed-loop control system using LoRa. The system needed to guarantee closed-loop stability is to design *node-to-gateway* scheduler and the controller together. The work proposes a distributed-queuing based channel access scheme with an event-triggered controller. The medium access control and the controller are chosen at design-time taking into account the low duty-cycle, time-on-air and multiple channels available in LoRa. At run-time, the access to the channel is controlled with the medium access layer and the decision to transmit is made by the event-triggered controller. The results show that our system can handle upto 10 District Meter Areas (DMA) where LoRaWAN fails at only 3 DMAs with upto 50% better packet delivery ratios and 80% lower average end-to-end delays. This work shows that LoRa can satisfy the requirements of a slow-loop

critical infrastructure and adapt to the changing demands of the underlying physical system caused by varying load and faults while ensuring system stability.

The contributions of the thesis will show that LoRa is currently under-utilised. The work will show that by designing scheduling schemes, it is possible to use LoRa for WA-CPS applications that are distributed over a wide-area, require bi-directional communication and have low datarates. The work will also highlight an important observation; namely, a single scheduling scheme or protocol cannot meet the requirements for all applications. Every application has some unique element, which we can leverage to meet application requirements instead of having a one-fits-all solution.

Summary of Contributions. The key contributions are described herein. The thesis focuses on bi-directional LoRa-based communication networks with symmetric uplink and downlink requirements, most useful for WA-CPS applications. It identifies several applications where low-datarate, long-range communication capabilities of LoRa can be used to meet application requirements. It also identifies various under-utilised or unused resources available in LoRa networks like multi-stakeholder deployments, multiple channels and similar uplink and downlink modulation schemes that make possible node-to-node and gateway-to-gateway communication, to design reliable communication protocols. The thesis then proposes data-driven scheduling schemes for node-to-node, gateway-to-gateway and node-to-gateway communication to meet the requirements for applications like opportunistic vehicular networks, multiple overlapping networks, and water distribution networks.

1.8 Thesis Outline

The thesis is organised as follows. First, in Chapter 2, we will review the background of LPWANs, LoRa and related works common to all contributions of the thesis. Every subsequent chapter is broadly organized as follows. We first introduce the problem, opportunity and proposed solution. Next, we describe the background needed for the chapter, followed by the main contributions and evaulations. Finally, we discuss the related works and conclude the chapter. In Chapter 3, we propose a opportunistic data handover protocol that deals with the problem of unreachable gateways in delay-tolerant vehicular networks. In Chapter 4, we propose a wireless overlay network to facilitate data handover between gateways of overlapping LoRaWAN networks to

overcome the problems caused by interference between overlapping networks and unavailable network bandwidth. In Chapter 5, we use a co-design scheme to design an event-triggered controller and a queuing based channel access scheme to facilitate closed-loop control of slow-loop critical infrastructures. Finally, we present the conclusions, discussions and future work in Chapter 6.

2

Background to LoRa, LoRaWAN and Related Works

Over the last decade, LPWANs have gained significant importance in industrial and research deployments. From fewer than 12 occurrences for all LPWAN related research in 2015 to approximately 2040 publications in 2021[1], LPWANs have become a vital research topic. Industry has also adopted LPWANs for several use-cases. Over 150 cellular companies in 60 countries around the world have deployed a mobile IoT network that follows the NB-IoT or LTE-M specification[2][3]. SigFox has managed deployments in 72 countries since its inception in 2010[3]. LoRaWAN's global coverage is 162 countries with first deployments dating back to 2015[49]. Such significant amount of investment by industry shows that LPWAN's target real-world use-cases. Most LPWANs have limitations in terms of their reliability, throughput, security and energy efficiency. E.g. LoRaWAN uses ALOHA channel access which has reliability issues and Sigfox does not encrypt packets implying security issues. These limitations are some of the actively researched areas of LPWANs.

This chapter covers the background on LPWANs, LoRa physical layer and its data link layer, LoRaWAN. This chapter also discusses LoRa transceivers and LoRaWAN architecture. The chapter is concluded by discussing the state-of-art research in LPWANs, LoRa and LoRaWAN. The background material covered in this chapter is common to all the work of this thesis. Every chapter also has a background and related works section that covers topics related to the proposed solutions. Chapter 3 will discuss

[1] https://scholar.google.com
[2] https://www.gsma.com/iot/mobile-iot-commercial-launches/
[3] https://www.gsma.com/iot/deployment-map/

works related to opportunistic mobile networks. Chapter 4 will discuss topics related to reinforcement learning, multiple gateways, overlapping networks in LoRaWAN. Chapter 5 will discuss works related to wireless control of industrial systems.

2.1 LPWANs

LPWAN are a class of wireless communication technologies. A characteristic of all LPWANs are that they communicate directly to a gateway over long distances and consume low energy. Several LPWAN protocols have been designed in the last decade. Some of them are NB-IoT[4], Sigfox[3], LoRa/LoRaWAN[2], Weightless[50], EC-GSM-IoT[51], 802.15.4v[52], Dash7[53], 802.11ah[54]. There are many factors to compare these LPWAN protocols, such as maximum datarates achievable, proprietary components, protocol maturity, encryption schemes, frequency band, deployment model, modulation schemes, number of nodes per gateway, latency, bandwidth and business model. These parameters mean different applications are suited to different LPWANs. Several authors in [1, 55, 56, 57, 58] have done a detailed comparative study of all LPWAN protocols. The parameter space for comparison is large, and so Table 2.1 focuses only on the ones relevant to current work like the modulation scheme, deployment models and medium access protocol.

The thesis' work is focused primarily on LoRa for several reasons. The first is that Semtech Corporation has only patented the wireless physical layer, and the users are free to design the communication stack on top of it. This allows for a flexible and open architecture that can be tweaked for every application. The second is that LoRa is one of the only popular LPWANs that use the same uplink and downlink communication modulation. The third is that LoRa follows an ad-hoc, free-for-all deployment model, which allows designing some exciting communication protocols.

2.2 Hardware

This thesis focuses on LoRa, so this section describes the hardware. The section also discusses the node and gateway design. The wireless physical layer and LoRa gateway's design give LoRa and LoRaWAN their unique appeal.

	LoRa-LoRaWAN	NB-IoT	Sigfox	802.15.4v
Propriety components	Physical layer	Full stack	Physical and MAC layers	Open standard
Maximum datarate/node	50kbps (FSK)	127kbps downlink, 159kbps uplink	100bps	Depends on modulation scheme
Frequency	sub GHz ISM Bands	LTE and GSM bands	sub GHz ISM Bands	sub GHz ISM Bands
Medium Access Protocol	ALOHA-based	OFDMA downlink, SCFDMA uplink	ALOHA-based	Superframe-based
Modulation Scheme	Chirp Spread spectrum	SC-FDMA (uplink), OFDM (downlink)	BPSK(uplink) , GFSK (downlink)	FSK, OFDM, O-QPSK
Deployment Model	Private or operator based	Operator-based	Operator-based	Private

Table 2.1: LPWAN comparisons [55, 58]

2.2.1 Nodes

LoRa nodes are embedded systems with wireless communication capability. A typical LoRa node design is shown in Figure 2.1. A node consists of a microcontroller, a radio transceiver, an energy source (battery or energy harvesting sources like solar, wind, radio-frequency, vibration), sensors and actuators.

The nodes have to be designed, so they work for several years with minimal intervention[4]. This node design enables the nodes to be deployed in remote and inaccessible locations. With energy-harvesting systems, it also allows the systems to run for prolonged periods even when no human access is possible [59].

[4] https://www.semtech.com/lora/what-is-lora

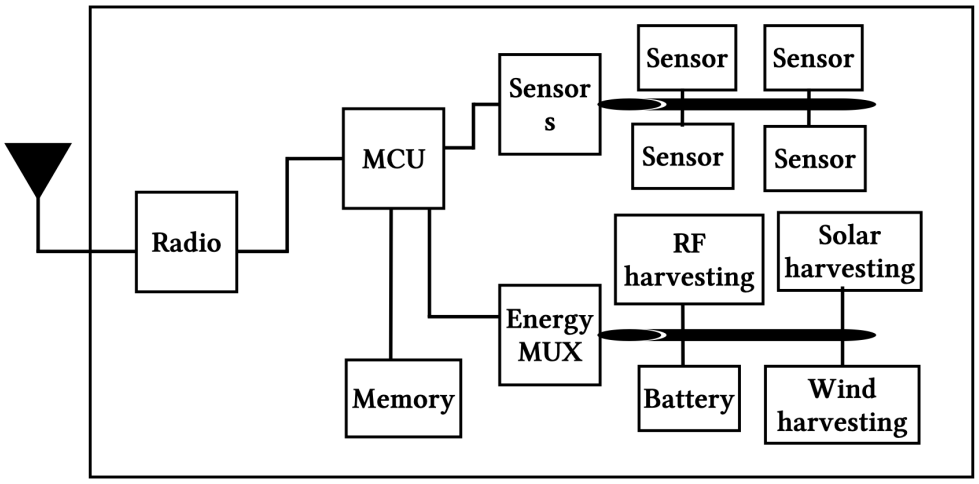


Figure 2.1: LoRa node example

The unique component of LoRa-enabled nodes is that they have a transceiver capable of communicating using LoRa physical-layer. The two most commonly used LoRa transceivers are SX1272 and SX1276[5]. These transceivers communicate over SPI to the microcontroller. They have several I/O pins capable of indicating to the microcontroller when a transmission succeeds or fails, a preamble is detected, or a message is received. The transceiver is half-duplex, which means it can either transmit or receive on only one frequency at one point in time.

2.2.2 Gateway

The hardware that sets apart LoRa systems from most other wireless communication systems is their powerful gateway. Most gateways in wireless systems like WiFi, Bluetooth, 802.15.4 are half-duplex systems that can only transmit or receive on one frequency. Half-duplex implies that wireless access to the gateways also needs to be scheduled just like nodes.

Gateways in LoRa deployments have a different architecture compared to nodes. The gateways have higher communication capability. They are usually mains-powered,

[5] https://www.semtech.com/products/wireless-rf/lora-transceivers

multi-channel receivers with two transceivers. Two transceivers allow for transmission on one frequency while the gateways receive messages on another frequency band. The architecture for a LoRa-enabled gateway is shown in Figure 2.2 and described below.

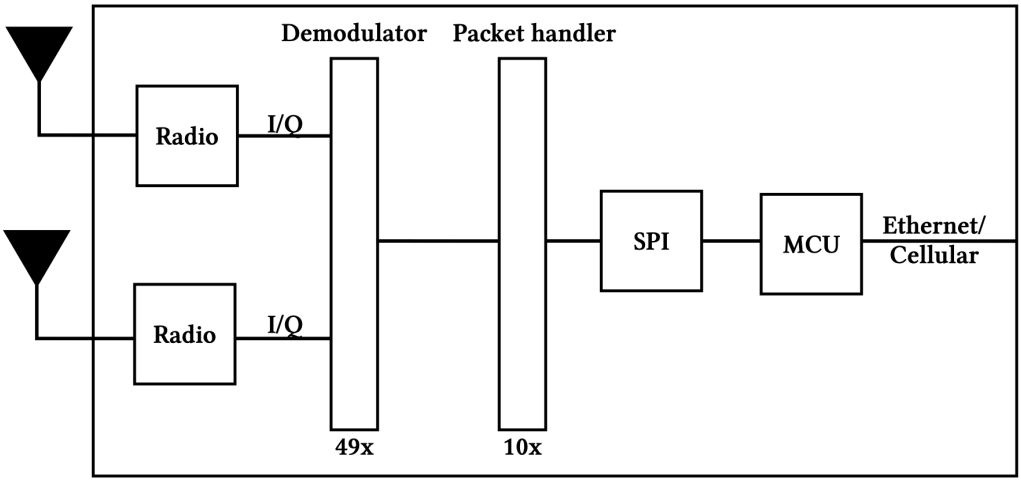


Figure 2.2: LoRa gateway architecture [60]

Every LoRa gateway has two transceivers connected to a concentrator which is in turn connected to a microcontroller. The microcontroller also has another high-bandwidth communication system like ethernet or a cellular connection. The transceivers listen to a wide-band signal, generally 3-4 times the typical LoRa channel bandwidth. The transceivers then transmit the I/Q signals to the concentrator, where the concentrator can decode them. The concentrator has 49x LoRa demodulators and can decode up to 10 LoRa packets with different spreading factors simultaneously. The concentrator forwards the decoded messages to the microcontroller, which are transmitted upstream to a server via the high-bandwidth communication link like ethernet, WiFi or cellular.

This architecture allows multiple nodes to communicate to a gateway at any point. The nodes need to transmit with the same preamble that a gateway listens on; if they do not, the gateway drops the message. The preamble filtering allows the nodes and gateways to drop irrelevant messages.

2.3 LoRa Physical layer

LoRa is the physical layer patented by Semtech Corporation in 2014. It uses the Chirp Spread Spectrum (CSS) modulation to transmit and receive data. CSS signals are modulated as chirps increasing linearly in frequency over time. CSS improves a signal's resilience to frequency selective fading and narrowband interferences by sweeping over a larger bandwidth. Two examples of symbols are shown in Figure 2.3. As can be seen, the frequency of a signal increases as time progresses. The symbol can be determined by starting frequency and the time of the signal frequency's rollover. The frequency (f) of a LoRa symbol ($symbol$) at time (t) can be mathematically expressed as follows[61]:

$$f(t, symbol) = symbol * \frac{BW}{2^{SF}} + \frac{BW}{T_s} * t \quad mod\ BW \tag{2.1}$$

where $symbol \in 0 \cdots 2^{SF} - 1$, T_s is the symbol interval defined as $\frac{2^{SF}}{BW}$, $t \in 0 \cdots T_s$, SF is the spreading factor and BW is the bandwidth of the signal. SF and BW are described below. As can be seen from Equation 2.1, the frequency of the transmission depends on the symbol and time.

Five parameters decide every LoRa transmission, *spreading factor, bandwidth, coderate, frequency and transmit power*. A combination of these parameters defines the datarate, transmission range and energy consumption.

Spreading Factor (SF) is the ratio between the symbol rate and chip rate. Spreading factor for LoRa varies from 5 to 12 in increments of 1. A lower SF has the highest bit rate but also a shorter range and requires a higher receiver sensitivity. A higher SF has a longer range and makes the signal more resilient to noise as it decreases the signal-to-noise ratio at which a packet can be successfully decoded and has a lower receiver sensitivity to decode the packets.

Bandwidth (BW) is the frequency range in which the signal is spread. Bandwidths can be one of 10.4, 20.8, 62.5 125, 250 or 500KHz. Bandwidth is directly proportional to the datarate. A higher BW means a higher datarate and a lower time-on-air. Higher BWs also have lower receiver sensitivity. Lower BWs require temperature-compensated crystal oscillators (TCXO) on nodes and gateway as they have a much lower error threshold which is only possible with TCXO.

Coderate (CR) is a Forward Error Correction (FEC)[62] code used in LoRa transmis-

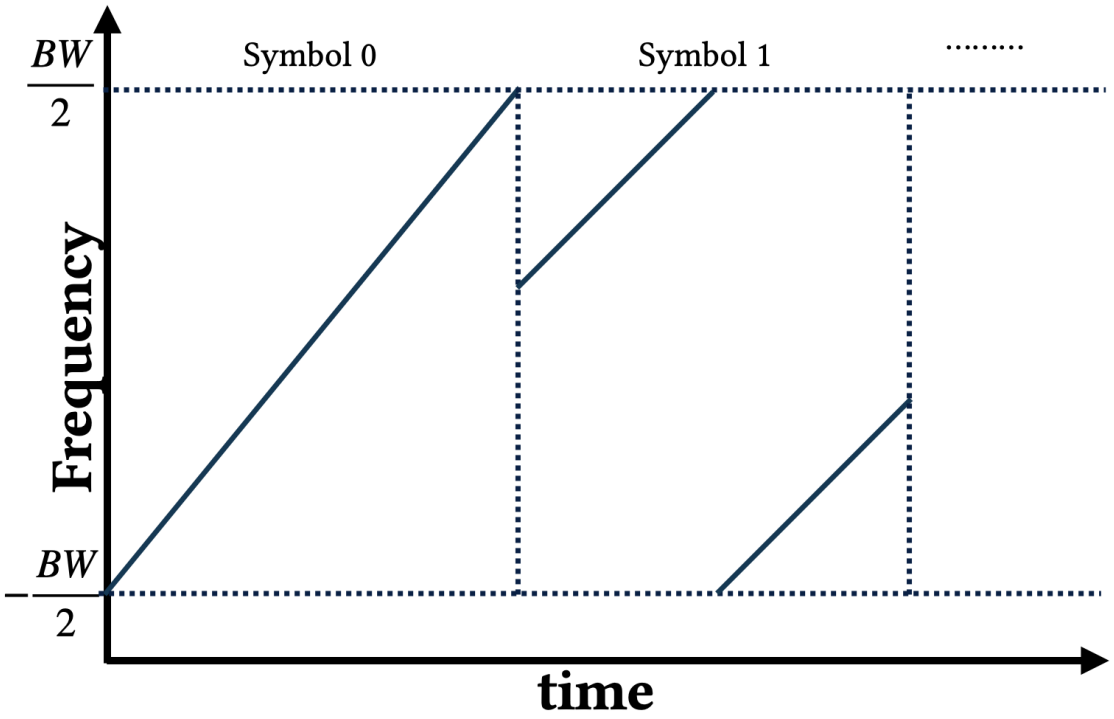


Figure 2.3: LoRa symbol examples

sions. CR can be selected from 1 to 4, which corresponds to an FEC-CR of $4/5$,$4/6$,$4/7$ and $4/8$. CR offers resistance against burst interference.

Frequency is the centre of the communication bandwidth at which transmission occurs. The frequency bands available depend on the communication region. This thesis's work is done in the European region, specifically UK, so OFCOM in the UK defines frequency usage. Two frequency bands are available in this region. LoRa can transmit in the low frequency (UK: $433.05 - 434.79MHz$) or the high-frequency band (UK:$863 - 870MHz$). The nodes defined in Section 2.2.1 can transmit on either of the two frequency bands; however, the gateways defined in Section 2.2.2 generally communicate only on the higher frequency band. This is because the lower frequency band has lower bandwidth (2MHz in lower frequency band compared to 7 MHz in higher frequency band), higher noise as several sensors operate in the band and lower maximum transmit power (10dBM in lower frequency band compared to 14dBM in higher frequency band).

Transmit power is the transmission energy for every message. The maximum transmit power is also defined by OFCOM in the UK and depends on the frequency band. In

SX1272 and SX1276 transceiver chips, the transmit power is between 2 and 25mW in increments of 2mW.

Based on the chosen SF, CR and BW, LoRa datarates (R_b) are calculated as:

$$R_b = SF \times \frac{CR}{\frac{2^{SF}}{BW}} \tag{2.2}$$

Using the above formula, datarate varies from 5.468kbps to 0.293kbps when SF is increased from 7 to 12 with $CR = 4/5$ and $BW = 125KHz$.

LoRa messages have transmission times from 61ms-2.7s when the payload sizes increase from 1 byte to the maximum allowed packet size (250 for SF7 and SF8, 123 for SF9, 59 for SF10, SF11 and SF12). Figure 2.4 shows the time-on-air for different SF, with a fixed CR (1), BW ($125KHz$) and maximum allowed packet sizes. As shown in the figure, large SFs increase the time-on-air.

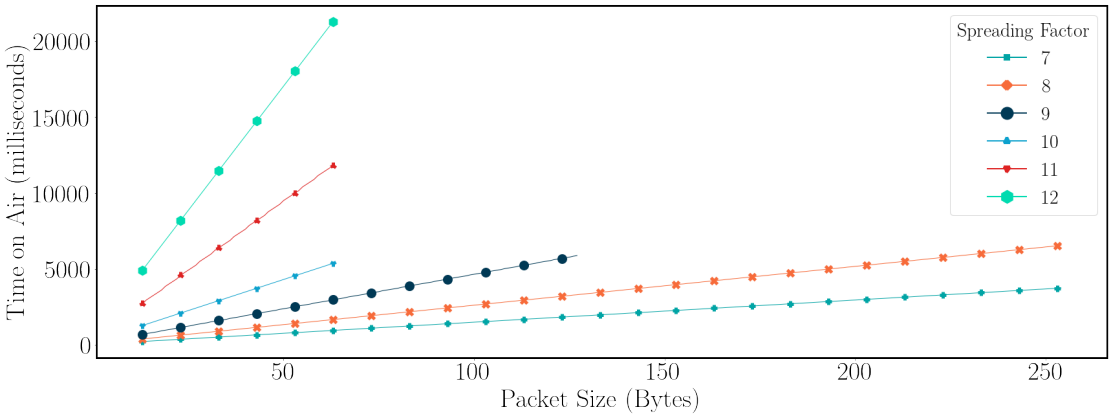


Figure 2.4: Time on air for different spreading factors with $CR = 1$ and $BW = 125$ [2]

The unlicensed spectrum ($863 - 870$MHz) significantly impacts the number of messages a LoRa node can transmit to comply with the duty-cycling regulations. LoRa communication in the UK generally happens in the $868 - 870$MHz, further divided into various sub-bands. A complete sub-band breakup with maximum allowable transmission power and duty-cycle is shown in Figure 2.5. Nodes transmitting on those channels need to adhere to a duty-cycling limitation which can be one of 0.1%, 1% or 10%.

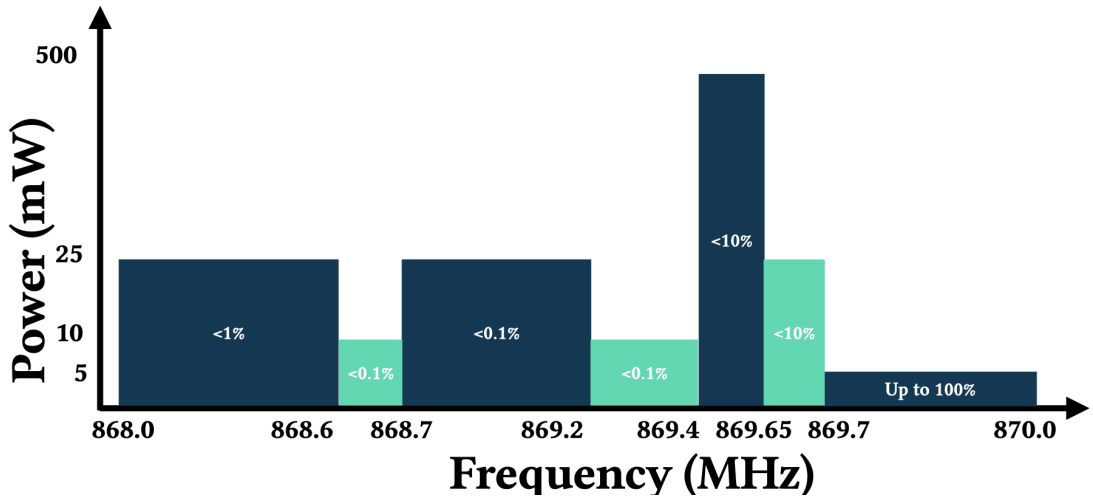


Figure 2.5: Spectrum limits in the 868.0-870.0 MHz plan[63]

2.4 LoRaWAN

LoRaWAN[2] is the data-link layer, standardised by the LoRa alliance, that runs on top of the LoRa physical layer. This section discusses the device classes, adaptive datarate, reliability mechanisms and the LoRaWAN architecture.

2.4.1 Device Classes

Device classes are different operational modes for LoRaWAN nodes. The device classes determine how the uplink (node-to-gateway) and downlink (gateway-to-node) communication occur. LoRaWAN specifications define three device classes.

In all three classes, nodes can transmit messages any-time using the ALOHA channel access method. Whenever the node wants to transmit a message, it chooses a frequency from a uniformly distributed random distribution. This random channel selection reduces the probability of all nodes transmitting in the same frequency.

The difference between all three device classes is based on their receiving and sleep schedules. The receive schedule essentially decides the downlink opportunities available between the gateway and the nodes. The three device classes can be seen in Figure 2.6 and they are:

- **Class A:** Nodes operating using this device class are the most energy-efficient.

In this class, downlink communication is possible only after an uplink message. After every successful uplink message, a node opens two receive windows ($1 and 2$ second after transmission). A gateway can use these windows to transmit a downlink message. The receive slots can be used for parameter update or sending acknowledgements. Whenever a node is not transmitting or has a receive window open, it goes to sleep to save its energy.

- **Class B:** Nodes operating using this device class have the same transmit and receive functionality as Class A. Additionally, the nodes open a receive window periodically to synchronise and receive any downlink messages. This device class consumes more energy and allows for more communication opportunities for over-the-air updates or time-synchronisation. This class is also valuable for CPS like applications that require periodic communication from gateways to nodes. However, all nodes open their receive window simultaneously, making this ideal for multicast applications but would significantly waste energy in a unicast scenario.
- **Class C:** Nodes operating using this device class consume the most amount of energy. These nodes are ideally mains-powered or have large batteries to provide the required energy. Nodes of this class can transmit a message anytime and open two receive windows like Class A devices. Additionally, instead of sleeping to conserve energy when not transmitting or receiving, these nodes open a receive window (i.e. continually listening when not transmitting). These receive windows are used for acknowledgements, parameter updates and over-the-air updates. Class C devices are also helpful for time-sensitive applications where gateways can request messages from nodes at any time. This device class trades off energy for the ability to communicate with any node at any time. This is most suited for CPS applications that do not have any energy constraints. However, systems with limited energy budget cannot utilise this device class. This device class like others suffer from low reliability due to the ALOHA channel access method.

2.4.2 Reliability

LoRaWAN provides two functionalities to increase communication reliability. The first one is to choose the suitable transmission parameters like SF, CR, and transmit power,

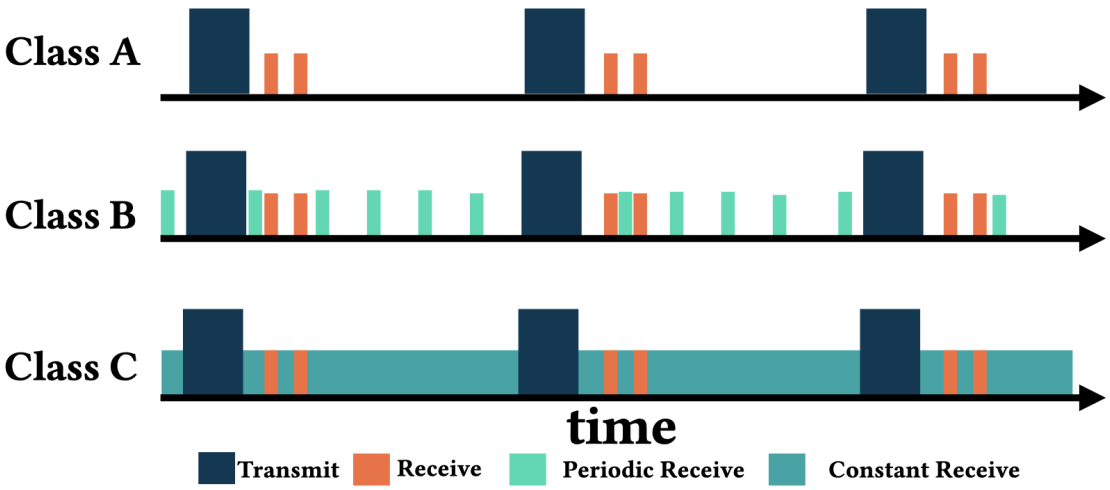


Figure 2.6: LoRaWAN Device Classes [2]

and the second one is to request acknowledgements for transmissions. The first method introduces reliability at the physical layer, while the second one is at the data-link layer.

LoRa physical layer allows a node to choose any SF, BW, CR, transmit power and frequency. All of these parameter combinations decide the datarate for communication and also the reliability of communication. A discussion about how the parameters determine the datarate and how every parameter determines the reliability is done in Section 2.3. Adaptive datarate (ADR) is discussed herein. ADR is the algorithm that decides which communication parameters the nodes should use. ADR algorithm adapts the SF, CR and transmit powers to ensure the desired reliability. Its functionality is divided into two parts; the first runs on the nodes and the second on the network servers. They can independently request a change in the transmission parameters based on the desired reliability levels. A detailed description of the ADR algorithm can be found in [33].

The second reliability mechanism defined by LoRaWAN specifications is requesting acknowledgements for uplink transmissions. Nodes can set a flag in their uplink transmission to request servers to acknowledge the message. The servers can send their acknowledgements in the receive windows. If a message is not successfully acknowledged, the nodes can choose to retransmit it. The retransmission algorithm and interval is not described in LoRaWAN specification and is left for the nodes to decide. Typically used retransmission algorithms are periodic and exponential backoff.

The nodes can also retransmit a message as many times as they want. LoRaWAN specification recommends a maximum retransmission count of 7 before discarding a message.

2.4.3 Architecture

LoRaWAN operates a 3-level architecture, consisting of *nodes*, *gateways* and *servers*. Nodes implement one of the three classes defined above. Nodes communicate with the gateways using a star topology. The gateways forward these messages to the servers. The LoRaWAN specification has a very comprehensive design to ensure security of the transmissions. It involves three servers that perform the encryption, decryption and verification of nodes and the node's data at different stages. The three servers defined by the LoRaWAN specification are *Join, Network and Application*. The complete architecture is shown in Figure 2.7.

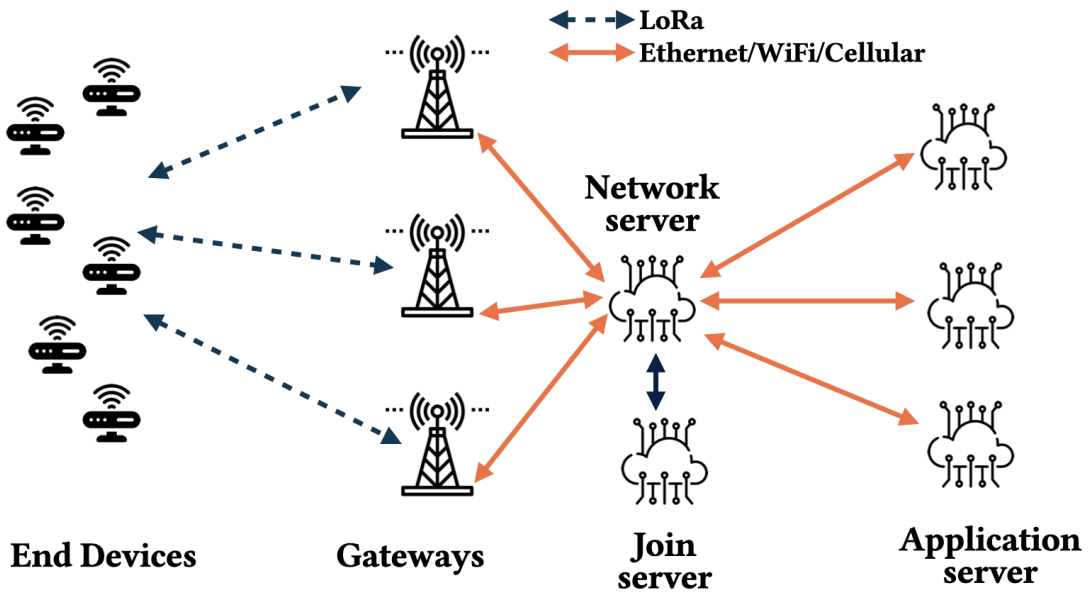


Figure 2.7: LoRaWAN architecture [2]

2.4.3.1 Servers

The Join server is responsible for joining new nodes in a network and distribution of the encryption keys. Every time a new node joins the network, the join servers generate the encryption keys used by the node, network server and application server. *The Network server* is responsible for verifying the messages and ensuring that the data is received from the intended node. It is also accountable for de-duplication in which the server drops multiple copies of the same message. The network server also runs the ADR algorithm and is responsible for transmitting acknowledgements. *The Application server* handles the node's data. This server is responsible for the encryption and decryption of the application payload using AppKey. The AppKey is only available with the Application server, so it is the only server that can read the application payload. The servers can run close to the edge or on the cloud, be on the same physical server, or be distributed to different servers. LoRaWAN provides the flexibility for users to decide how and where they want to run their servers.

2.4.3.2 Deployment architecture

LoRaWAN installations have been following an ad-hoc deployment model as anyone can deploy gateways anywhere without the need for a carrier like NB-IoT or Sigfox. The nodes, gateways, join servers, network servers and application servers can all belong to different stakeholders. This flexibility has allowed innovation to happen in different parts of the stack without affecting any other part. E.g. multiple gateways can be deployed at the same location by different stakeholders; the servers can run on the edge or cloud servers; stakeholders can choose from multiple implementations of server stacks.

LoRaWAN allows providers to deploy network servers that can be publicly or privately accessed. Several public network server providers exist like the things network[6], Semtech's IoT network[7]. These allow users to forward data from their gateways to the network server and manage their devices. Many users have also been deploying their

[6] https://www.thethingsnetwork.org
[7] https://ns.loracloud.com

private servers built around Chirpstack[8], Loriot[9], Everynet[10] and many more. Chirpstack is an open-source software stack that provides the join, network, application and geolocation servers. Semtech claims they have sold over 1.3 million gateway chips. As of June 2021, over 40000 gateways are connected on The things network, the biggest consortium of gateways distributed worldwide. However, this only accounts for around 3% of the gateways. Rest 97% are connected to different public and private servers.

This section discussed LoRaWAN specifications, device classes, reliability mechanisms and architecture. Inspite of its wide adoption, LoRaWAN suffers from a number of issues. A vast amount of research has been done in LoRa and LoRaWAN which are discussed in the next section.

2.5 Related Work

There has been a lot of research done in LPWANs, LoRa and LoRaWAN. The research ranges from increasing reliability at physical and data-link layers, improving security, handling mobility, looking at new applications. These research areas can be broadly divided into four sections:

1. Characterising and improving the performance at the physical layer
2. Parameter allocation and adaptive datarate at the data-link layer
3. Fusing of information from different gateways to increase network throughput
4. Miscellaneous work like multi-hop LoRa, mobility and time synchronisation

2.5.1 LoRa Physical Layer

Initial days of LoRa research saw many works designing mathematical models and running testbed experiments for LoRa. These works focussed on characterising LoRa links and validating the creators' claims about long-range communication.

One of the first and most prominent works is done by Petajajarvi et al. in 2015[64]. This work was the first study on the coverage area of LoRa. The experiments were

[8] https://www.chirpstack.io
[9] https://www.loriot.io
[10] https://www.everynet.com

performed in Oulu, Finland on land and water, with and without mobile nodes. The results showed that with $SF = 12$ and +14 dBm transmit power; the maximum communication range was 15 km on the ground and 30 km on water. With their real-world experiments, the authors estimated a path-loss model. This path-loss model has become a standard and used in several future works [65, 66, 67, 68, 69] to simulate and validate LoRa experiments in sub-urban scenarios.

Jörke et al.[70] perform similar experiments to Petajajarvi et al.[64]. They ran indoor and outdoor experiments for LoRa in 433 MHz and 868 MHz bands. They derive two new path-loss models based on the Okumura Hata Model used in a communication environment similar to Dortmund, Germany which is a sub-urban environment.

Linka et al.[71] further derive more realistic models based on Longley-Rice Irregular terrain model which uses real-world elevation to predict path loss. Their results show that terrain data is useful to increase the prediction accuracy, which would further help network planning.

Finally, Xu et al.[72] design LoRa propagation models for indoor scenarios. The authors conduct an extensive measurement drive to collect data from several multi-floor buildings. These measurements show that the path loss coefficients, fade margin, and energy consumption vary depending on where a node is and what communication parameters it uses to communicate. The energy consumption can vary up to 145 times depending on the communication parameter. The authors derive coefficients for the distance-dependent path-loss model that can be used by works that design indoor LoRa systems.

All of these works have been used in simulators to validate designs and help in deployment planning. The work in this thesis uses the Oulu path loss model for its simulations.

2.5.2 Increasing Physical Layer Decoding Rates

Another area of research in LoRa is increasing the decoding rates at a single gateway. Eletreby et al.[73] propose Choir, a system that uses cloud-based collaboration to improve the number of successfully decoded transmissions. The work uses the hardware imperfections of LoRa transceivers to disentangle and decode failed messages and increase network throughput by 6.84x. It has several flaws as it cannot

decode multiple transmissions with low Signal-to-Noise Ratio (SNR). This has led to several works[74, 75, 76, 77] that try to improve upon different Choir problems. FTrack: Parallel Decoding for LoRa Transmissions[75] is the first of these works. The work uses a parallel decoding scheme that resolves collision by considering both time and frequency domain features. FTrack increase the throughput of the network by 6x when 20 nodes simultaneously transmit. FlipLoRa: Resolving Collisions with Up-Down Quasi-Orthogonality[74] shows that Choir and FTrack can significantly increase the network throughput; however, they fail in real deployments as it is difficult to extract features from low Signal-to-Noise Ratio (SNR). FlipLoRa encodes packets with interleaved up chirps and down chirps instead of only using up chirps as in LoRa. FlipLoRa can improve the throughput by 3.84x compared to LoRa's physical layer. Another work is CoLoRa: Enabling Multi-Packet Reception in LoRa[77] that builds on the work of Choir. The work shows that the hardware imperfections fluctuate, reducing Choir's throughput significantly with many collided packets. CoLoRa solves Choir's issues by using a method to cancel inter-packet interference and extract information from low SNR signals. CoLoRa increase network throughput by 3.4x compared to Choir and up to 14x compared to LoRaWAN. All these solutions are designed using Software Defined Radios (SDR), making all currently deployed hardware redundant.

All of these works increase the decoding rates, and they can all be utilised by the higher layers to design protocols. The throughput for all of the works in this thesis would increase by increasing the decoding rates and the networks would be able to support more nodes.

2.5.3 Data-link Layer

A significant body of work is done to characterise the reliability of LoRaWAN and allocate transmission parameters. The first set of work creates an analytical and simulated model of every part of the LoRaWAN stack like acknowledgements, multiple gateways, transmission parameters, and the packet delivery ratio and delays. The second set of work looks at the static and dynamic allocation of transmission parameters, controlling access to the channel, varying how acknowledgements are sent. The first type of work has helped researchers understand the problems with LPWANs, LoRa and LoRaWAN, and the second set of works solve the identified issues.

Modelling LoRaWAN's capacity is another important area of research. Several

works [78, 79, 80, 81, 82] try to accurately model different components of LoRa to design an analytical model. Toro-Betancur et al.[82] combines several of the works to model the communication reliability of LoRaWAN with device-level accuracy. The work models information about capture effect, quasi-orthogonality, multiple gateways, duty-cycling and shadow fading. This work has laid down an accurate analytical model for the packet delivery ratio used in real LoRa deployments assuming the node's positions and information about the propagation models. The work misses the energy consumption at each node which would complete the model of LoRa and allow the users to derive what sort of packet delivery ratios to expect given the node and gateway positions. However, these works assume that all gateways communicate with the same network server, are deployed by a single stakeholder and ignore the more realistic scenario of multiple networks sharing the wireless spectrum.

The works mentioned above created analytical models of LoRaWAN. The next set of works look at simulated models of LoRaWAN. Bor et al.[65] conduct scalability analysis of LoRaWAN. They use experiments to model LoRa's communication behaviour which further informs a simulation's parameters to run a scalability study. The studies show that LoRa can support 120 nodes per 3.8 ha with data extraction rates (DER) of 90%. A high DER is especially important in CPS applications where every message needs to be delivered unlike sensor-only applications where a lost message might not lead to any major system failures. The works showed that quite a small number of nodes can be supported in a large area with normal LoRaWAN, further motivating the need for this dissertation. Haxhibeqiri et al.[67] conduct another scalability analysis in terms of the number of nodes per gateway of single-gateway deployments. The results show that as the number of nodes increases to 1000, the losses are up to 32%.

Another aspect of LoRa is its ADR capabilities. Li, Raza, and Khan[33] conduct a detailed performance analysis of the ADR algorithm. The results show that the convergence of ADR is slow (order of hours) and is sensitive to the user-defined delay parameter. Pop et al.[83] conduct an essential analysis of the effect of downlink traffic. The work looks at how acknowledgements affect the network throughput of the LoRaWAN network. The results show that network throughput reduces to 15% when all nodes request acknowledgements for their messages compared to when no nodes request it in a large-scale network. Capuzzo, Magrin, and Zanella[84] examine the reasons for packet failures in LoRaWAN. The results show that for high density

networks, with a light load, the probability of successful uplink message reduces as the maximum number of retransmissions increase, that is, if retransmissions go from 4 to 8, the probability of successful packet reception reduces from 50% to 25%. Therefore increasing maximum number of retransmissions is detrimental to the network and so should be avoided.

This downlink problem seriously impacts LoRaWAN to be used for CPS applications. The main focus of this thesis is WA-CPS applications that have similar requirements for uplinks and downlinks. The work by Pop et al. clearly shows that the current LoRaWAN needs to be improved to meet CPS requirements.

2.5.4 Network coordination

LoRaWAN's architecture is unique as, unlike previous wireless communication systems where a node would communicate with a single gateway, LoRaWAN allows a node's message to be received by multiple gateways. All gateways forward these messages to the network server, where they are de-duplicated. This in-network coordination is seen in cellular networks in Cloud Radio Access Networks but is relatively new in the low-power wireless communication space. The in-network coordination can happen at the signal, bit or packet level.

At signal level, the gateways transmit the I/Q signals directly to a cloud server, where they are collaboratively decoded. The first work of this type was Charm[85]. Dongare et al.[85] propose Charm, a system that uses coherent combining to increase the reception rates. The authors claim that Charm can increase the range by up to 3 times and battery life by up to 4 times. However, Charm requires a backhaul with 144 Mbps datarate if a gateway supports 64x125kHz LoRa channels[11], making it harder for the system to be deployed with low bandwidth backhaul networks. Liu et al.[86] build upon Charm's work by reducing the bandwidth required in the backhaul network to 24 Mbps. They do this by using compressive sensing with a custom dictionary that exploits the structure of packets. Both these works show that LoRa's decoding rates can be significantly increased, but they rely on having SDR devices that are not possible in current gateways.

At the bit level, packets are fully decoded at the gateways, and failed packets are

[11] 64 channels are commonly used in United States

transmitted to cloud servers to detect failed bits. An interesting idea is proposed by Balanuta et al.[87]. Disentangling messages by collaborative decoding requires an SDR, discounting all the current deployments as they do not have an SDR. The work in [87] uses opportunistic packet recovery to decode packets. It uses packets with failed CRCs at multiple gateways and uses these to decode the messages collaboratively. This helps with error detection and recovery and shows that it can decode up to 72% of packets that would have failed. This scheme significantly increases network throughput and does not require any hardware-level changes to the gateways. The above works assumed that gateways are deployed or controlled by the same entity to allow for such network coordination.

At the packet level, spacial diversity is exploited by LoRaWAN. Different gateways can receive and successfully decode packets from different nodes, so a failed message at one gateway could be successfully received at another. The messages are de-duplicated at the network servers. This is the approach taken by current LoRaWAN deployments. The approach can be further extended by offloading the network management to gateways belonging to different stakeholders, which is proposed by Delbruel et al.[88]. In [88], the authors tackle the assumption of a single stakeholder system by federating gateways of multiple stakeholders. The federated gateways distribute the load amongst each other by assigning nodes to specific gateways and holding that gateway responsible for encryption and decryption of their packets. Two problems exist with Delbruel et al.'s approach. The first is that they require a reliable internet connection that can be difficult in rural areas or hazardous areas. The second is that network servers have to hand over network encryption keys to gateways, which is a potential security issue.

The signal level network coordination is a good solution if all gateways belong to the same entity as the I/Q signals need to be sent to other servers who can decode all the packets. These solutions have currently been implemented with SDRs with a higher receiver sensitivity than transceivers designed specifically for LoRa. Although it is possible to implement signal level network coordination with commercially available transceivers, no work has explored this idea just yet. Bit-level network coordination is an interesting approach as it relies on gateways being able to decode partial information. However, the current approach can only resolve a maximum of 30-bit errors in a 25-byte packet and requires at least two gateways to receive the same packet. At the packet level currently used in LoRaWAN, approaches are mainly focused on network coordination

between multi-stakeholder networks. There is a limit to the number of gateways a single user can deploy, and so gateways of multiple networks coordinating is a logical way of increasing network throughput. This is an approach used in Chapter 4.

2.5.5 Miscellaneous

Several other works have been built using LoRa and LoRaWAN. This section focusses on some of these works like multi-hop LoRa and mobility.

2.5.5.1 Multi-hop LoRa

Cotrim and Kleinschmidt[89] conduct a review of mesh networks built on LoRaWAN. Even though LoRaWAN was not designed for multi-hop communication, there has been a significant amount of research developing multi-hop protocols for LoRaWAN. Multi-hop communication allows the system to communicate in areas where power supply is unavailable or access to the internet is difficult. These systems extend the communication range, improve reliability and energy efficiency.

Aslam et al.[90] argue in favour of multi-hop communication to increase the network's range and energy efficiency. They show that multi-hop communication can achieve 90% PRR and save 50% energy compared to single-hop network while also increasing coverage by 35%. Haubro et al.[91] implement Time Synchronized Channel Hopping (TSCH) over LoRa. The work looks at implementing the solution on Contiki-NG[92] and finding the appropriate slot lengths for communication. The authors claim 100% packet delivery ratio with SF7 in 6-hour experiments.

Tanjung et al.[93] propose an opportunistic and on-demand packet retransmission scheme. They use random linear network coding to replay packets missed by network servers and request it from other nodes. This requires nodes to keep the data received from other nodes. Although an interesting scheme, the work does not talk about the energy overhead for the nodes. The scheme would require nodes to be continually listening to receive messages and also reply to network servers. This may be possible only in scenarios where nodes are mains-powered.

Vázquez[94] propose a multi-hop communication scheme for uplink messages in LPWANs. The scheme uses a combination of TDMA, CSMA/CA and ring passing mechanism. The scheme also uses a beaconing system which requires nodes to be

awake to synchronise to the network. The work implements X-Mac[95] on top of LPWANs.

Dwijaksara, Sook Jeon, and Jeong[96] propose a multi-hop gateway-to-gateway communication scheme. They argue that gateways can be deployed in areas where no backhaul connectivity is available. In such situations, the gateways have to route packets to reach a gateway with backhaul connectivity. They use packet aggregation to reduce the number of transmissions needed between gateways. The results show that gateway-to-gateway communication can reliably provide communication in areas with no backhaul networks while still meeting all requirements laid down by LoRaWAN specifications.

These works show that multi-hop communication is a solution to many problems faced by LoRa deployments, like nodes being out of range and network exhaustion at gateways. These approaches are analogous to the approaches used in Chapters 3 and 4 where the multi-hop communication is leveraged between the nodes and gateways, respectively. Unlike the current approaches, which have used multi-hop communication for aggregating gateway information or using multi-hop in static nodes, the work in this thesis uses multi-hop for mobile applications and network coordination between multi-stakeholder deployments.

2.5.5.2 LoRa mobility

An important area of research that is slowly picking up is dealing with mobility in LoRaWAN. LoRa alliance and Semtech realised the potential of using LoRa for mobile applications quite early. With this intention, they added two crucial functionalities to LoRa's hardware and software. The first was a nanosecond precision clock[60] in their newest generation of LoRa's gateway chips. Nanosecond precision clocks provide highly precise timestamps which can be used for calculating the position of nodes using Time Difference on Arrival (TDOA). Secondly, they introduced mobility and roaming capabilities in LoRaWAN specification. Roaming is used in areas where multi-stakeholder LoRaWAN deployments exist as discussed in Section 2.4.3. Roaming allows gateways to handle data from nodes that do not belong to their network.

Inspite of the software and hardware capabilities designed within the LoRa ecosystem, there are still several flaws in mobile LoRaWAN deployments. Kousias et al.[97] conduct an empirical analysis of ADR capabilities of LoRaWAN in mobile scenarios.

They perform their analysis on a real-world dataset of LoRa messages collected in Belgium. The study shows that ADR works well in low mobility settings; however, ADR's benefit decreases as mobility increases. The reason for this is the slow update rate of the ADR algorithm. This slow update rate is one of the reasons that it is recommended to use a static spreading factor and transmit power for mobile applications.

Mobility also results in new challenges when applied in real-world scenarios. For example, authors show that velocity has an observable effect on the packet delivery ratio for speeds higher than 25 km/h due to Doppler effects[98], while Marcelis, Rao, and Prasad[99] find the relationship between packet delivery ratio, mobility and distances between devices and gateways. The results show that the outage probability for furthest nodes is as low as 0.004, but the frame loss measured at this distance can be 70%. Furthermore, the results showed that burstiness happens in both mobile and stationary scenarios. Without reliable links between mobile devices and gateways, opportunistic data forwarding between nearby devices becomes an interesting idea to explore, which is used in Chapter 3.

Zguira, Rivano, and Meddeb[100] propose using LoRaWAN in delay-tolerant mobile networks. They present a data handover scheme between bikes, which shows the benefits of an opportunistic data handover scheme in mobile LoRa systems. The results show that opportunistic data handover significantly improves throughput and reduces energy consumption. However, their approach does not take into account the last-contact time with gateways, which is leveraged in this thesis, which shows significant performance improvements.

Two works by authors in [101, 102] look at how to allocate transmission parameters to nodes in mobile LoRaWAN scenarios effectively. In the first work, the authors use Gaussian-ADR and an exponential moving average-based ADR. The proposed schemes are essentially a low-pass filter to deal with rapid changes in signal-to-noise ratio. The method increases the network throughput by 17% and 81% as compared to typical ADR. In the second work, the authors use an allocation scheme that depends on the received power and each spreading factor's sensitivity. The work conducts a thorough analysis of a single-gateway scenario with mobile and static nodes. The scheme is highly agile, and the spreading factor is updated at each uplink packet. However, it significantly increases the downlink capacity needed by a single gateway and does not consider a more realistic multiple gateway scenario.

2.6 Summary

The above works show that research in several areas are needed to improve LoRa and LoRaWAN's performance. The works show that LoRa and LoRaWAN's reliability can be significantly increased by assigning appropriate transmission parameters, communication slots, retransmission mechanisms, increasing decoding rates etc. However, there are other problems that cannot be handled like overlapping networks leading to collisions, unreliable communication due to bursty interferences, or de-synchronisation in nodes. Also, several of these works have only considered uplink transmissions and not much work has been done for downlink transmissions. This thesis looks at CPS applications with symmetric uplink and downlink requirements.

The following three chapters each have different applications and unique resources that we are leveraging to design reliable wireless communication systems. There are several distinctions between the three applications. E.g. the number of nodes reduces from 20k+ in Chapter 3 to less than 100 in Chapter 5, gateways from 100 to only 1, and the deployment area from 100s km^2 to the size of a park. These differences increase the network bit flux[103] of the system. Network bit flux measures an application's bandwidth requirements over a large geographical area. It is handy when dealing with multiple gateways in a deployment. In a single gateway scenario, comparing the application's data rate requirements and the network throughput are sufficient to decide if the application's needs can be served. However, this cannot be directly applied to scenarios with multiple overlapping gateways as we would have to individually ensure every gateway can meet the requirements complicating the analysis. By leveraging, network bit flux, we can compare the application needs with network throughput over an area, allowing us to compare networks with arbitrary number of gateways. In this thesis, we are increasing the network bit flux as we go through the various chapters. Increasing network bit flux implies that the amount of traffic a single gateway has to handle increases, and we need to design different protocols to meet the requirements. The first chapter we look at is opportunistic data forwarding in delay-tolerant vehicular networks with the largest number of nodes, gateways and communication area compared to other systems of this thesis, indicating it has the lowest network bit flux.

3

Opportunistic Data Forwarding in Delay-tolerant Vehicular Networks

Vehicular networks or Transportation CPS (TCPS) is an area of CPS that targets transportation systems in smart-city environments. Transportation CPS is being designed to facilitate a hassle-free, comfortable and safe transport experience for humans along with real-time asset tracking. TCPS takes information from the *physical domain* like road conditions, vehicles count and positions, busyness and uses that information in the *cyber domain* for applications like traffic management, toll collection, ridesharing, smart parking, pedestrian crossing, and autonomous driving. TCPS is a complex interaction of real-time and static data from vehicles, traffic and pedestrian information, and their interactions.

This chapter focuses on delay-tolerant high-throughput TCPS applications like information queries. Due to their city-wide span, delay-tolerability and low energy consumption, LoRaWAN is the ideal communication protocol to meet TCPS's requirements. LoRaWAN is currently unable to provide the required reliability as vehicle mobility introduces problems like deep fading and unreachable gateways because of communication links affected by weather, obstructions, distance to gateways and load on the gateways.

Opportunity. LoRaWAN nodes communicate directly with their gateways, which may not always be in range. Node-to-node communication can help deal with mobility-related issues discussed above. Instead of holding, or dropping data, the nodes can communicate with other nodes with a better probability of communicating with the

This chapter is based on [42]. I contributed to idea of the paper, algorithm and writing. I fully implemented the system, wrote the software and collected the data.

gateways. This data handover can reduce latencies and significantly improve network throughput.

3.1 Problem Introduction

Problem. Much work has been done to improve throughput and reduce delays in LoRaWAN networks as discussed in Section 2.5. Most of these works assume that nodes and gateways are static. Only recently have some works[97, 100, 101, 102] emerged discussing mobile nodes. A rising class of applications require low-power mobile solutions to provide new services, e.g. delay-tolerant information like weather reports, road conditions, traffic volumes, and micro-mobility vehicles. LoRa gateways also support nanosecond precision timestamps for obtaining geolocation information from LoRa-enabled nodes for asset tracking. All of this points to LoRaWAN being a candidate technology for mobile smart-city applications.

Mobile scenarios suffer from varying wireless channel as locations and environments change over time. Wireless channels can be unreliable due to obstacles or deep fading, which further depend on weather conditions, line-of-sight to gateways and speed of the vehicle.

The low duty cycling (i.e., 1%, 10%) regulated by OFCOM makes this situation even more challenging. If a node misses its communication opportunity, it needs to wait minutes before transmitting its data again, thus causing significant delays to data delivery. This problem is further aggravated because a large number of gateways are needed to cover a geographical area. Assuming a 2 kilometre communication radius[104], over 130 gateways deployed in a grid are needed to cover the city of London, without any overlapping communication areas. This makes the cost of deploying gateways non-trivial, and so it is infeasible to expect high-density deployments to meet the communication requirements. Consequently, new and improved protocols are required to ensure on-demand delivery of data during movement.

Solution. This chapter proposes that instead of trying to send data via unreliable wireless channels to gateways or awaiting for next communication opportunity with a gateway, LoRaWAN nodes can send their data to gateways via other nearby nodes that have a better communication link to a gateway. Typically, the nodes have a higher chance of communication than actual node mobility, so finding reliable neighbours for

data forwarding instead of retaining unsent data awaiting a good connection with a gateway is easier. By exploiting this observation, two new opportunistic data forwarding approaches are proposed, effectively reducing data delivery delays and improving network throughput.

A new network metric is first introduced to enable nodes to choose the best node to hand over their data to; namely, Real-time Contact Aware Expected Transmission Count (RCA-ETX). This metric seamlessly models node mobility as packet transmission delay, widely accepted in many objective-function-based data forwarding protocols.

We then combine RCA-ETX with a throughput-optimal stochastic backpressure scheme and propose Real-time Opportunistic Backpressure Collection (ROBC). ROBC can cope better with uncertain links in mobile scenarios and delivers improved performance compared to only RCA-ETX. Importantly RCA-ETX and ROBC operate in a fully distributed and autonomous manner without any knowledge of node mobility and only relying on node-to-gateway link conditions.

3.2 Problem Definition

This section first provides a formal definition to the mobile LoRaWAN network with static gateways (MLoRa-SG). We briefly describe Contact-Aware ETX (CA-ETX), the foundation to our approach, and the reasons why it is not directly applicable in MLoRa-SG.

3.2.1 System Model

Standard LoRaWAN networks consist of only node-to-gateway links communicating in a star topology. In TCPS, nodes move over time and the gateways are static which we term as Mobile LoRaWAN network with static gateways (MLoRa-SG). MLoRa-SG suffers from varying link conditions leading to an unreliable communication system. This chapter aims to provide a solution that allows nodes to send their data via other nearby nodes if their direct connection to the gateways is currently weak or unavailable.

The formal definition of dynamics of MLoRa-SG are described next. MLoRa-SG can be modelled as a time-varying weighted graph $G(\mathcal{N}, \mathcal{L}, \mathbf{C}(t))$. In this graph, $\mathcal{N} = \mathcal{D} \cup \mathcal{G}$ denotes all nodes in an MLoRa-SG, where $\mathcal{D}$ denotes the set of all nodes generating and relaying data, and $\mathcal{G}$ is the set of all gateways collecting data packets from the

network. $\mathcal{L} = \mathcal{L}_d \cup \mathcal{L}_g$ denotes the set of all possible wireless links. While $\mathcal{L}_d$ denotes the link between node pairs $\{x, y\} \in \mathcal{D}$, each entry in $\mathcal{L}_g$ represents the virtual links between a node $x \in \mathcal{D}$ and all gateways $g \in \mathcal{G}$. $\mathbf{C}(t)$ denotes a $|\mathcal{N}|^2$-dimensional matrix representing the channel capacities over all wireless links between nodes at time t. Each entry $c(t)_{x,y} \in \mathbf{C}(t)$ is dynamic and may change overtime. Figure 3.1 illustrates an example of $G(\mathcal{N}, \mathcal{L}, \mathbf{C}(t))$.

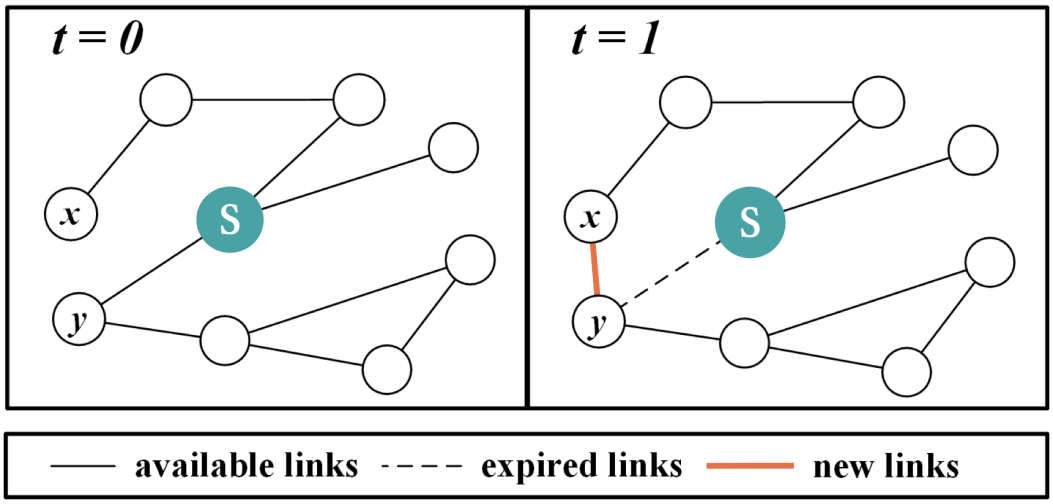


Figure 3.1: Compared to the network at $t = 0$, link $(x, y) \in \mathcal{L}$ regains communication (i.e. $c(t)_{x,y} > 0$), whereas device y loses its link with sinks $\mathcal{G}$ (i.e. $c(t)_{x,\mathcal{G}} = 0$) at $t = 1$.

3.2.2 CA-ETX and challenges when applying in MLoRa-SG

Contact-Aware ETX (CA-ETX) [105] is a network metric that extends Expected Transmission Count (ETX) [106], which is a widely accepted network metric used by many routing and scheduling protocols in WSN. ETX measures the number of (re)transmissions needed to deliver a packet to its destination. ETX is defined as $\frac{1}{PRR}$, where PRR is the Packet Reception Ratio which is the number of messages acknowledged to the number of messages transmitted. ETX is designed for static network deployments and does not take mobility into account. CA-ETX extends ETX by accounting for network mobility. CA-ETX performs on-line measurements of packet transmission delays over every

opportunistic link. It then uses these delays to make scheduling and routing decisions and in WSNs with mobile gateways and static nodes.

Although CA-ETX provides a novel way to incorporate mobility in the computation of ETX seamlessly, it cannot be directly applied in MLoRa-SG due to the following problems:

1. **Opportunistic contact between nodes.** CA-ETX is designed for systems where gateways are mobile and nodes are static. In contrast, in MLoRa-SG, nodes are mobile, and gateways are static. Therefore, routing protocols such as Routing Protocol for LLN (RPL) or Backpressure Collection Protocol (BCP) may never converge to a stable routing gradient and consequently will be unable to provide guarantees such as minimal delay in RPL or maximum throughput in BCP.

2. **Low maximum duty-cycle and limited bandwidth.** The maximum duty-cycle and link capacity for LoRaWAN networks are regulated by OFCOM. Therefore, nodes in MLoRa-SG can not exchange high-frequency periodic information as in WSNs. This prevents nodes from obtaining necessary information of one-hop neighbours needed to converge to the desired objective. Consequently, when computing CA-ETX, historical variance σ and average μ are likely to be outdated, which can significantly degrade overall routing and scheduling performance.

A new network metric that copes with sparsity of information and node mobility is needed.

3.3 Opportunistic Data Forwarding with RCA-ETX

We propose Real-time CA-ETX (RCA-ETX), a new metric that works with low-frequency communication, commonly found in LoRaWAN to model the communication links between nodes and gateways. With this metric, we propose a solution which overcomes the uncertain nature of MLoRa-SG while inheriting the simplicity and effectiveness of CA-ETX.

3.3.1 Operation Overview of Data Forwarding with RCA-ETX

LoRaWAN operates in the unlicensed spectrum and must adhere to the duty-cycling regulations. This means that nodes in MLoRa-SG can only send messages when their wait

times have expired. This makes opportunistic routing via neighbours more challenging. The simplest method of discovering a neighbour to route through opportunistically is via overhearing.

Given a mobile node x, its opportunistic neighbour is defined as $\mathcal{D}_x(t) := \{y, \forall \mathbf{C}(t)_{x,y} = 1\}$. On receiving a broadcast message from an opportunistic neighbour y at time t, x computes the RCA-ETX of the link RCA-ETX$_{x,y,\mathcal{G}}(t)$, which is a sum of RCA-ETX$_{x,y}(t)$ + RCA-ETX$_{y,\mathcal{G}}(t)$, that models the link between x and $\mathcal{G}$ via y. x then compares RCA-ETX$_{x,y,\mathcal{G}}(t)$ with RCA-ETX$_{x,\mathcal{G}}(t_x)$ that denotes the RCA-ETX of the link in between x and gateways $\mathcal{G}$, if the value of RCA-ETX$_{x,\mathcal{G}}(t_x)$ is larger than RCA-ETX$_{x,y,\mathcal{G}}(t)$, x tries to send the data stored in its queue to y; otherwise x holds the data until next transmission opportunity, which can be either its next transmission slot or on receiving a broadcast message from a nearby node. If the message is successfully sent to y, x drops the message from its queue.

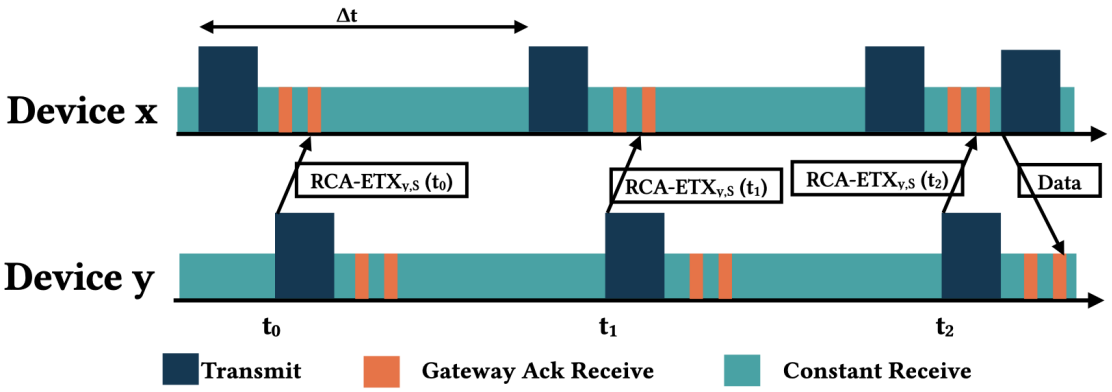


Figure 3.2: An operational overview of our solution with LoRaWAN Class-C sensors, which are able to consistently overhear data packets from nearby neighbours). Device x decides not to send its packets via y when receiving its RCA-ETX sent at time t_0 and t_1, whereas x sends its packet to y at time t_2.

An example of such an operation is demonstrated in Figure 3.2, where node y broadcasts its RCA-ETX$_{y,\mathcal{G}}(t)$ when trying to transmit data to the gateways $\mathcal{G}$. This message can be overheard by its neighbours $x \in \mathcal{D}_y(\cdot)$, that had no direct connection with the gateways $\mathcal{G}$ at times t_0, t_1 and t_2. On receiving RCA-ETX$_{y,\mathcal{G}}(t)$ from y, node x computes RCA-ETX$_{x,y}(t)$ of link (x, y) at time t, and decides to handover its data to y at t_2 after confirming that the overall RCA-ETX value of y is smaller than x, that is:

$$\text{RCA-ETX}_{x,\mathcal{G}}(t) > \text{RCA-ETX}_{y,\mathcal{G}}(t) + RPST_{x,y}(t) \tag{3.1}$$

.

3.3.2 Calculating RCA-ETX$_{x,\mathcal{G}}(t)$

Before introducing RCA-ETX$_{x,\mathcal{G}}(t)$, we need to first define Packet Service Time (PST). PST is defined as the time needed for a successful packet transmission over link (x, y) at time t. PST is used by CA-ETX to make routing decisions. Given a node x and a set of gateways $\mathcal{G}$ in MLoRa-SG, virtual link $(x, \mathcal{G})$ can be regarded as a queue with time-varying PST $\mu_{x,\mathcal{G}}(t) \geq 0$, which is the time duration for a packet to be successfully transmitted via link $(x, \mathcal{G})$ at time t. For every virtual link $(x, \mathcal{G})$ PST, $\mu_{x,G}(t)$ is computed at the beginning of a time slot reserved for its node-to-gateway communication. If the node-to-gateway is successful, i.e. the packets are delivered and acknowledged by the gateways, PST is reset to $1/c^{max}_{x,\mathcal{G}}$. If the node-to-gateway communication is unsuccessful, $\mu_{x,G}(t)$ is the combination of transmission time $1/c_{x,\mathcal{G}}(\dot{t}^{n+1}_x)$ (i.e. the PST when node x is in its $(n+1)$-th contact with gateways) and time delay $\dot{t}^{n+1}_x - t$ (i.e. time required before next gateway contact). This can be computed as

$$\mu_{x,\mathcal{G}}(t) = \begin{cases} 1/c_{x,\mathcal{G}}(t) & , \dot{t}^n_x \leq t \leq \ddot{t}^n_x \\ \dot{t}^{n+1}_x - t + 1/c_{x,\mathcal{S}}(\dot{t}^{n+1}_x)) & , \text{otherwise} \end{cases} \tag{3.2}$$

where $\dot{t}^n_x$ and $\ddot{t}^n_x$ denotes the first and last time slots of n-th contact between x and any gateway $g \in \mathcal{G}$.

However, when computing PST, $\dot{t}^{n+1}_x$ is not available when $\ddot{t}^n_x < t < \dot{t}^{n+1}_x$. To overcome this problem, we introduce a Real-time PST (RPST) $\mu'_{x,\mathcal{G}}(t)$, which is computed as follows:

$$\mu'_{x,\mathcal{G}}(t) = \begin{cases} 1/c_{x,\mathcal{G}}(t - \Delta t + t^{\Delta}_x) + t^{\Delta}_x & , c_{x,\mathcal{G}}(t - \Delta t + t^{\Delta}_x) > 0 \\ 1/c_{x,\mathcal{G}}(\ddot{t}^n_x) + t - \ddot{t}^n_x + t^{\Delta}_x & , \text{otherwise} \end{cases} \tag{3.3}$$

where Δt denotes the node-to-gateway communication interval, which is a given parameter; t^{Δ}_x and denotes the wait time before x can broadcast a message via $(x, \mathcal{G})$.

Figure 3.3 illustrates an example of computing $\mu'_{x,\mathcal{G}}(t)$. As can be seen, given a node x, we use *estimated delay* (i.e. $t - \ddot{t}^n_x + t^{\Delta}_x$) instead of *actual delay* (i.e. $\dot{t}^{n+1}_x - t$) that

cannot be calculated. It is worth noting that nodes use the capacity $c_{x,\mathcal{G}}(\cdot)$ acquired at the *previously* successful transmission time slot $t - \Delta t + t_x^{\Delta}$ or $\ddot{t}_x^n$, since $c_{x,\mathcal{G}}(t)$ is also not available.

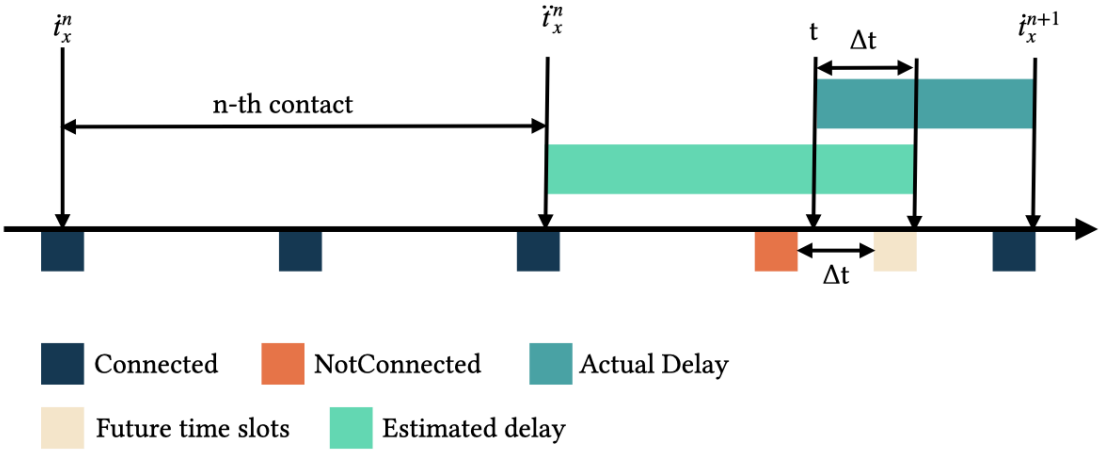


Figure 3.3: Node-to-gateway PST $\mu_{x,\mathcal{G}}(t)$ is estimated at every time t (Derived from [105])

RCA-ETX$_{x,\mathcal{G}}(t)$ can now be calculated from $\mathbb{E}[\mu'_{x,\mathcal{G}}(t)]$ at time t. However, since nodes are mobile and run at low-duty cycles in MLoRa-SG, the opportunity of transmission is much lower than the probability of topology changes in $\mathcal{G}$. Therefore, instead of the long-term average, we compute $\mathbb{E}[\mu'_{x,\mathcal{G}}(t)]$ as exponentially weighted moving average (EWMA) as below

$$\mathbb{E}[\mu'_{x,\mathcal{G}}(t)] = \begin{cases} (1-\alpha)\mathbb{E}[\mu'_{x,\mathcal{G}}(t-\Delta t)] + \alpha\mu'_{x,G}(t) & t > 0 \\ \mu'_{x,G}(0) & t = 0 \end{cases} \tag{3.4}$$

where α denotes a parameter that controls weight in between current RPST $\mu'_{x,G}(t)$ and last RPST $\mu'_{x,G}(t-\Delta t)$ computed at previous time slots. Higher the α, faster the RPST adapts to current RPST; however, this reduces the overall scheduling stability.

3.3.3 Calculating RCA-ETX$_{x,y}(t)$

In MLoRa-SG, each node must obey the maximum data send rate (i.e. 1% duty cycle for most general data channels) regulated by a relatively long interval (e.g. $\Delta t = 2$ mins). As a result, given a link $(x, y) \in \mathcal{L}$, we should not use the long-term average to estimate link capacity $c_{x,y}(t)$ as in classic ETX. The solution here is to *estimate* the link capacity

as per RSSI values, which can be obtained from every received broadcast message from other nodes. With the RSSI value $\gamma_{y,x}(t)$, $c_{x,y}(t)$ can be computed as:

$$c_{x,y}(t) = \begin{cases} c_{x,y}^{\max} \frac{\gamma_{y,x}(t) - \gamma^{\min}}{\gamma^{\max} - \gamma^{\min}} & \gamma^{\max} \geq \gamma_{y,x}(t) \geq \gamma^{\min} \\ c_{x,y}^{\max} & \gamma_{y,x}(t) > \gamma^{\max} \\ 0 & \gamma_{y,x}(t) < \gamma^{\min} \end{cases} \tag{3.5}$$

where $\gamma^{\max}$ denotes the RSSI value above which (x, y) should achieve its maximum capacity $c_{x,y}^{\max}$; $\gamma^{\min}$ RSSI value below which (x, y) has minimum capacities (i.e. 0 bits/s).

This equation is frequently used in many implementations such as the link stack in Contiki [107]. Consequently, $RPST_{x,y}(t)$ can be simply computed as follows:

$$RPST_{x,y}(t) = 1/c_{x,y}(t) \tag{3.6}$$

3.4 Real-time Opportunistic Backpressure Collection (ROBC)

The approach proposed in Section 3.3 can be regarded as a greedy solution which tries to forward data via a neighbour with a shorter potential delay (i.e. smaller RCA-ETX). However, this shortest-path approach typically suffers from poor throughput performance[108]. Furthermore, frequent topology changes also make it less effective for finding better candidates for data handovers. In contrast, solutions based on queue length (e.g. BCP) have proven more effective when coping with the uncertain dynamics of mobile scenarios[105, 108].

In this section, we propose another solution by integrating RCA-ETX with back-pressure algorithm for MLoRa-SG.

3.4.1 Backpressure routing

ROBC is built on the principles of backpressure routing[109]. Backpressure is based on network queuing and sees every node as a queue. It is commonly used for routing in non-static networks where statically defined routes may have some time-varying component that needs to be accounted (e.g. mobility, varying packet generation rates, service times etc). While using backpressure, the routing decisions are made explicitly

based on a normalized weight for every packet. This weight is calculated based on the current link conditions and the queue length of the next-hop. The next-hop is chosen such that the data flows from the longest to shortest queue which guarantees that no queue's overflow or messages are lost. This principle has been applied to several wireless routing algorithms [108, 110, 111, 112].

3.4.2 Queue Dynamics

In MLoRa-SG, we assume each node $x \in \mathcal{D}$ generates $r_x(t, \Delta t)$ amount of new data at time t over the past time interval Δt. Each node maintains a queue (i.e. data buffer) to store data that cannot be forwarded as there are no viable routes to gateways $\mathcal{G}$. Let $Q_x(t) \geq 0$ be the queue length of x at time slot $t \geq 0$. Given a node x, from time $t - \Delta t$ to t, its queue length is updated as below:

$$Q_x(t) = |Q_x(t - \Delta t) - f_x^{\text{out}}(t, \Delta t)|_+ + r_x(t, \Delta t) + f_x^{\text{in}}(t, \Delta t) \tag{3.7}$$

where $|\cdot|_+$ denotes a non-negative operator where given a real number a (i.e. $|a|_+ = a$ if $a > 0$, otherwise $|a|_+ = 0$); $f_x^{in}(t, \Delta t)$ and $f_x^{out}(t, \Delta t)$ are the total incoming and outgoing data of node x in between time t and $t - \Delta t$, respectively, which is defined as follows:

$$f_x^{\text{in}}(t, \Delta t) = \sum_{y \in \mathcal{D}_x(t, \Delta t)} f_{y,x}(t, \Delta t) \tag{3.8}$$

$$f_x^{\text{out}}(t, \Delta t) = \sum_{y \in \mathcal{D}_x(t, \Delta t)} f_{x,y}(t, \Delta t) \tag{3.9}$$

where $\mathcal{D}_x(t, \Delta t)$ denotes all x's opportunistic neighbours (from which x receives their RCA-ETX broadcast) over time $t - \Delta t_x$ and t; $f_{x,y}(t, \Delta t)$ denotes the amount of outgoing data from x to y over t and $t - \Delta t$.

3.4.3 ROBC Algorithm

The operation of ROBC is similar to what was proposed in Subsection 3.3.1.

3.4.3.1 Weight Calculation

Given a node $y \in \mathcal{D}$, it broadcasts its RCA-ETX$_{y,\mathcal{G}}(t)$ and queue length $Q_y(t)$ at every Δt interval. When device x overhears this packet, it computes its ROBC weight $\omega_{x,y}(t)$ as:

$$\omega_{x,y}(t) = (Q_x(t)/\varphi_x(t) - Q_y(t)/\varphi_y(t)) \tag{3.10}$$

where

$$\varphi_x(t) = \frac{1}{\text{RCA-ETX}_{x,\mathcal{G}}(t)}$$

Here $\varphi_x(t)$ is called Real-time Gateway Quality (RGQ) of node x. An upper bound and a lower bound (i.e. $0 < \varphi^{\min} \leq \varphi_x \leq \varphi^{\max} < \infty$) should be given for all node $x \in \mathcal{D}$ to guarantee ROBC stability [105]. The intuitive idea behind RGQ is similar to PST proposed in Subsection 3.3.2. Since $\varphi_x(t)$ can be regarded as the average rate a node sends its data to the gateways, it is exploited as a modifier to correct original queue lengths (i.e. how long a packet will have to wait until it is served).

3.4.3.2 Opportunistic Scheduling and Data Forwarding

Based on ROBC weight $\omega_{x,y}(t)$ computed in Equation (3.10), node x decides whether to handover data stored in its queue to node y. If $\omega_{x,y}(t) > 0$, x forwards $\delta_{x,y}(t) = Q_x(t) - Q_y(t)\varphi_x/\varphi_y$ amount of data to y, otherwise it keeps all of its data. This can be regarded as a scheduling problem where node x decides to forward its data via an opportunistic neighbour y or via itself x by comparing ROBC weights $\omega_{x,y}(t)$ to $\omega_{x,x}(t) = 0$.

It is worth mentioning that traditional queue-based approaches utilise full link capacity $c_{x,y}(t)$ for every transmission opportunity. However, due to the low duty cycle and the sparsity of available links in MLoRa-SG, nodes only send $\delta_{x,y}(t)$ amount of data to reduce recursive loops (where data packets are sent back and forth between two nodes). Therefore, node y will not send data received from x back even if y hears from x before its next forwarding opportunity to the gateways.

3.5 Customised Device Classes for Practical Operations

The solutions proposed in Sections 3.3 and 3.4 assume all nodes use the **Modified Class C** to ensure that node-to-node communications are always available as demonstrated in Subsection 2.4. **Modified Class C** operates in a similar manner to original Class C with one change. Instead of opening its receive slots for downlink communications from gateways (i.e. Rx2), it listens to its Tx channels (i.e. Rx1) to overhear the Tx from other nearby devices within communication range. This design is compatible with LoRaWAN Class-A, which needs to be supported by all LoRaWAN-enabled devices. The operational overview of this class is demonstrated in Figure 3.4.

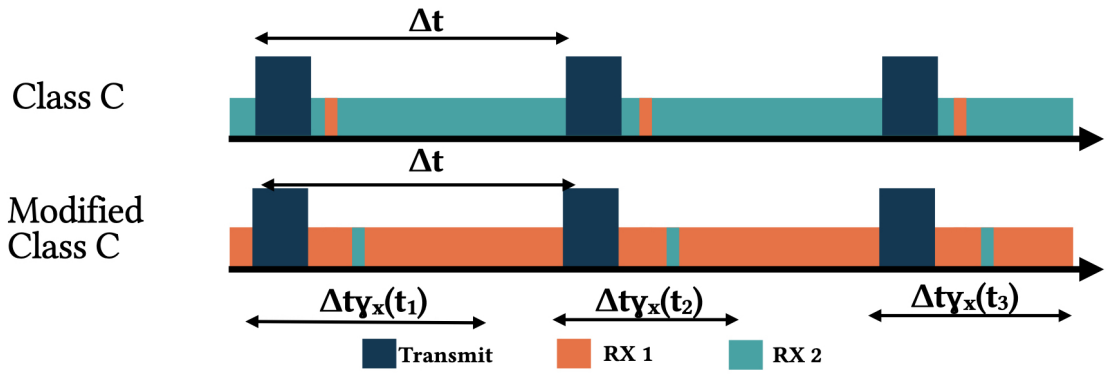


Figure 3.4: Modified Class-C is proposed to provide a reception feature required by node-to-node communications. Tx denotes the time slots reserved for device up-links. Rx1 denotes time slots for gateway acknowledgement via the same channel as Tx. Rx2 denotes the time slots for gateway down-links.

3.6 Evaluation based on London Bus Network

In this section, we describe our experimental setup, communication network, evaluation metrics and present the results of our simulations.



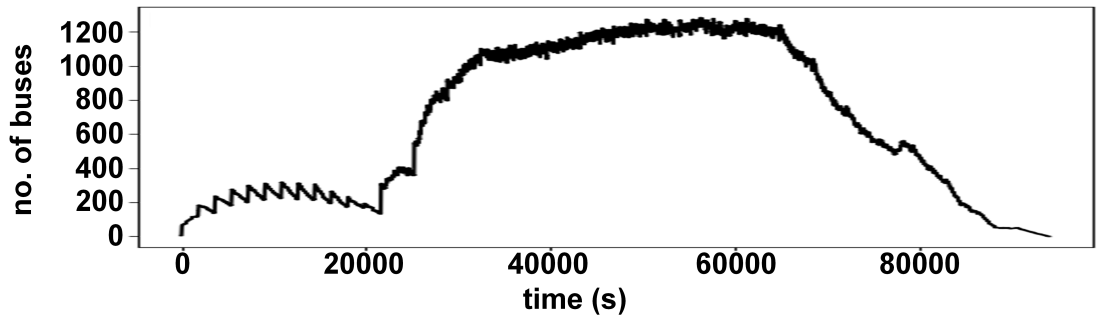
Figure 3.5: This map shows the simulation area where bus routes are the lines with colors.

3.6.1 Experimental Setup

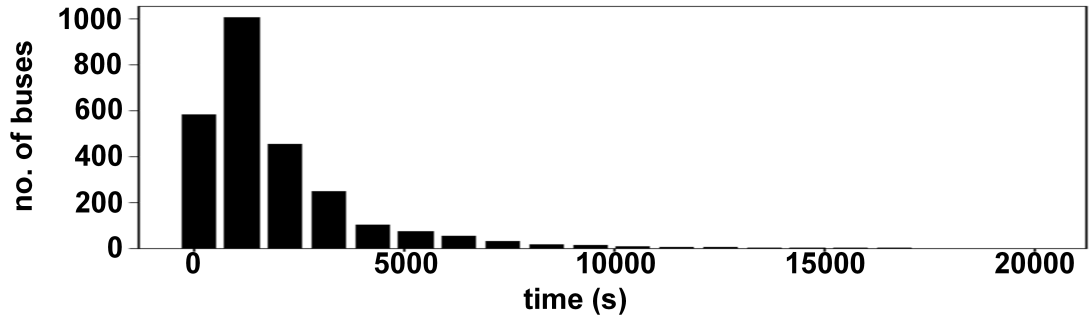
LoRaWAN is suitable for long-distance, low datarate communication, and so it is particularly suitable for many urban or *smart city* scenarios. The primary goal of our opportunistic data collection protocol is to collect data in areas of low coverage or over-used spectrum.

Scenario. TCPS applications can be simulated in several ways. The criteria for choosing the right environment depends on factors like number of nodes, the distance they travel, the speed variabilities, and their span in the interested area. A physical network that achieves all of the above requirements is a bus network. Bus networks generally connect every part of a city, have variable speeds and large number of nodes. Random node mobility patterns can be generated and tested, however, a network that is currently in use better represents realistic scenarios. For these reason, we chose to simulate the London bus network for our evaluation. However, the work does not rely on the mobility model. The protocol is evaluated in a 600 km^2 area with 24-hour simulations.

Bus Network. A trace-driven London bus network illustrated in Figure 3.5 is chosen to simulate TCPS mobility. The dynamic nature of the bus network provides an iconic



(a) Number of active buses in 24 hours.



(b) The distribution of bus active duration.

Figure 3.6: London bus network's (a) number of active buses and (b) distribution of buses' active duration[113] over 24 hours.

example of a complex vehicle network, where the distances between buses and gateways change with the speed of each vehicle. For evaluation, we collect routes from the London bus network based on the dataset provided by Transport for London (TFL) [113]. This dataset consists of the real timetable for each route, based on which we can simulate the mobility of each bus. The total number of active buses and their travel time distribution are illustrated in Figures 3.6 (a) and (b), respectively.

Simulation Framework. The SUMO simulator[114] is used to simulate the bus network. SUMO is a widely used simulator for modelling of intermodal traffic systems - including road vehicles, public transport and pedestrians. It is a microscopic and continuous traffic simulator designed to handle large road networks. The simulator takes the map and bus information, like start time, stop time and frequency, to create the buses. The communication network is simulated in OMNeT++ using FLORA[69]. For every bus generated by the SUMO simulator, a corresponding node is also created in OMNeT++. Veins[115] acts as the bridge between OMNeT++ and SUMO. The simulator is updated every 100 milliseconds. The speed of the vehicle is calculated based on the

distance between two points and the time taken by the vehicle between these two points (available from the timetable for the London bus network). All other are default parameters for all three software.

LoRaWAN Network. In our simulation, LoRaWAN is a one-hop communication network where all nodes can communicate with any gateway in the network. All the gateways are connected to a single central server (network server) where all the data is collected via Ethernet. Every bus in the simulation is equipped with a LoRaWAN node. These nodes generate a 20-byte message every 3 minutes and store it in a first-in-first-out (FIFO) data queue. The nodes are half-duplex, and so they are in a receive state unless they are transmitting. The nodes use modified Class-C. They are always listening on one channel which is the normal data transmission channel instead of the channel dedicated for downlink traffic.

LoRaWAN Physical and Network Parameters. There are three reasons we choose to use $SF = 7$. The first is that authors in [97] prove that the benefits of Adaptive datarate of LoRaWAN decreases as mobility increases. So using adaptive data rate has less advantage compared to using a static allocation. The second is that $SF = 7$ allows nodes to send a larger number of messages (as it has a lower airtime) and has the highest maximum packet size of 255 bytes. Although, packets using $SF = 12$ can travel longer distances compared to packets using $SF = 7$, the rate of collisions significantly increases as we increase number of packets per minute transmitted [116]. We also use one channel for transmission instead of multiple channels to increase the probability of overhearing a message. We use the log-distance path loss model with shadowing as the physical layer model with a path loss exponent of 2.32 as it is representative of a sub-urban environment for LoRa[64]. Upon message generation, nodes select up to 12 messages from the queue and create a new data packet. They also append their RCA-ETX value and data queue size to the data packets before transmissions. If any gateway successfully receives the packet, it sends an acknowledgement to the nodes. We assume that the acknowledgement messages are delivered instantly. If the packet fails, the nodes try to retransmit the message after the duty-cycle timer of 1% time-on-air runs out. Every node tries to transmit every message up-to eight times unless it generates a new packet in which case it resets the retransmission counter. Since the nodes are all using the same SF7 and listening on the same channel, they can overhear all the messages transmitted in their neighbourhood. Upon message reception,

nodes can choose to forward part of their queue or ignore the reception based on the chosen data forwarding scheme.

Network Deployment. The gateways are deployed in a uniform grid instead of a randomly deployed topology. Although it is highly unrealistic to have a perfect grid, it is challenging to discern the performance gain either from gateway locations or our data forwarding protocols if gateway locations are randomly chosen. The gateway-to-node communication range was set to 1 km. Two values of 0.5 and 1 km were chosen for node-to-node communications to simulate the effect of buildings, varying transmit powers, position of nodes and other wireless effects, which would vary the range of node to node communication as nodes are closer to the ground compared to gateways that are on top of buildings..

Schemes evaluated. Four different schemes are evaluated in this work:

- **LoRaWAN:** The nodes use **Class A** as described in Section 2.4.1. The nodes transmit a message periodically. If it is not acknowledged, they retransmit it upto it 7 times.
- **NoRouting:** The NoRouting mode builds on top of LoRaWAN. In addition to LoRaWAN, a queuing system is introduced at the application layer. The nodes keep all messages in queue that were not acknowledged. At every transmission, the nodes transmit a maximum of 12 messages from its queue in a data packet.
- **RCA-ETX:** In addition to the NoRouting scheme, the nodes also append their RCA-ETX value to the data packet. Using Equation 3.1, the nodes decide whether to forward data to another node or not.
- **ROBC:** In addition to the NoRouting scheme, the nodes also append their RCA-ETX value and queue lengths to the data packet. Using Equation 3.10, the nodes decide whether to forward data to another node or not.

3.6.2 Simulation Results

This section shows the results for the simulations comparing LoRaWAN, LoRaWAN with queuing, RCA-ETX and ROBC in MLoRa-SG. The moving average parameter (α) in Equation (3.4) is set to 0.5. Two metrics below are used for rest of the evaluations:

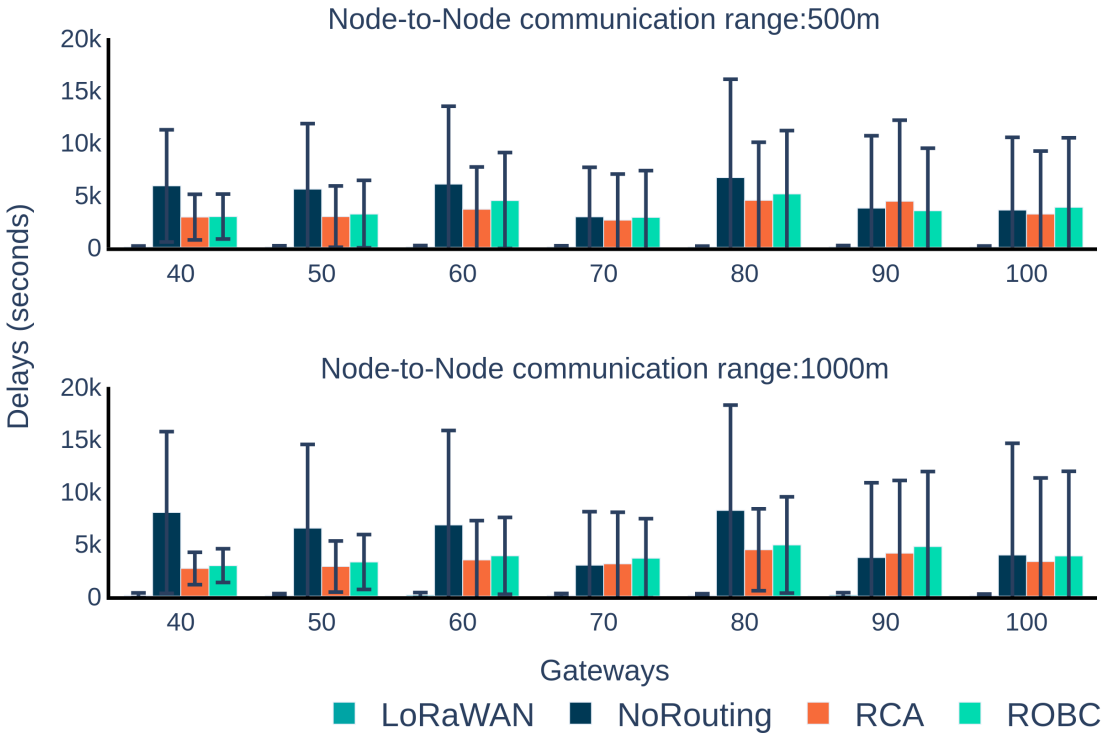


Figure 3.7: Average end-to-end delay with errors.

- **End-to-end delay:** End-to-end delay is measured by $\delta t(x) = t_g(x) - t_d(x)$, where $t_d(x)$ denotes the time when message x is generated, and $t_g(x)$ denotes the time when message x is received at the server. One of the major contributions of this work is to reduce end-to-end delays by utilising nearby LoRaWAN nodes for data forwarding. Reducing end-to-end delays allows the *cyber* layer to have information earlier making the decisions near real-time for the *physical* layer.

- **Throughput**. In contrast, throughput is a typical trade-off for the delay. Throughput is measured by the number of messages received at the server in a certain period. Higher throughput implies the *cyber layer* has more information which it can use to make better decisions for the *physical layer*.

The approaches are evaluated in networks with different gateway densities. The number of gateways deployed in the area illustrated in Figure 3.5 is varied from 40 to 100. Also, to simulate the MLoRa-SG networks in both two, the node-to-node communication ranges are set to 500 and 1000 meters, respectively. Figure 3.7 depicts the average end-to-end delays. As shown, RCA-ETX and ROBC successfully reduce 10%-25% delays compared to LoRaWAN with queuing in both settings with lower

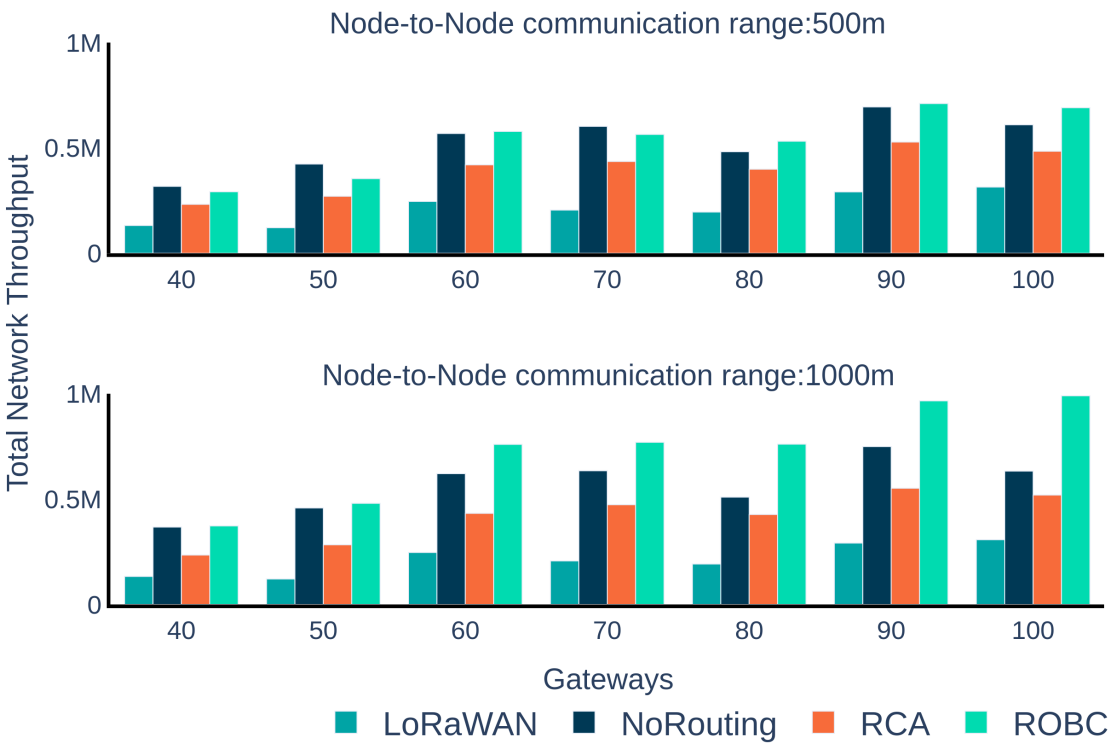


Figure 3.8: Total network throughput

gateway densities. However, they are less effective when gateway density increases (e.g. scenarios with 90 and 100 gateways). According to our analysis, this is because RCA-ETX is computed based on real-time PST depicted in Equation (3.3). This equation exploits past information to estimate the next successful gateway contact in the future, leaving higher errors when gateways are deployed in a grid topology. The delays for LoRaWAN remain quite low, as they retransmit a packet maximum of 7 times and then drop it. It is also worth mentioning that the performance gain in the rural environment is not as significant as expected. More extended node-to-node communication range only contributes to the scenarios where the number of gateways is less than or equal to 50.

All three of the approaches yielded longer delays when the number of gateways was set to 80. Further analysis reveal that the density of the grid has a significant impact on gateway locations, which further impacts delay and throughput. When the number of gateways is 80, their locations were less accessible for the buses (given their fixed route and mobility) compared to some other networks consisting of fewer gateways. However, we can still observe that RCA-ETX and ROBC still outperformed LoRaWAN

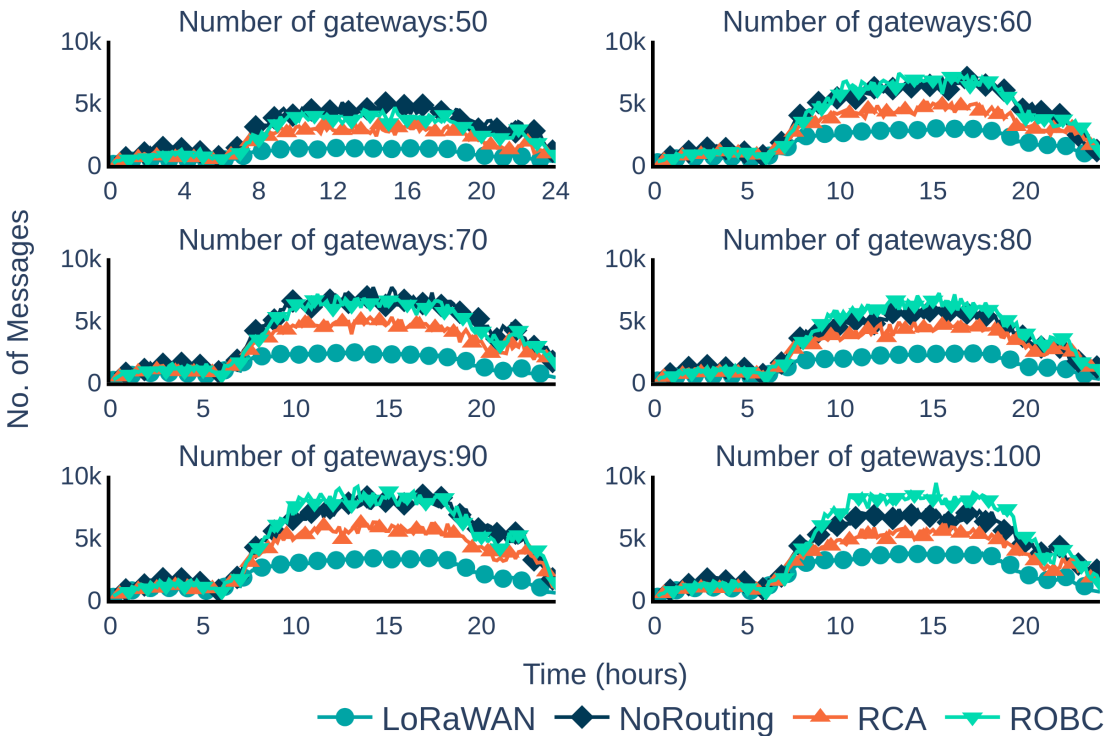


Figure 3.9: Throughput in environments where node-to-node communication range is 500m.

with and without queuing.

Figure 3.8 depicts the total throughput of the network under the same scenarios. As expected, RCA-ETX receives its performance gain by trading throughput. Fewer messages arrive at the gateways. In contrast, by exploiting queue dynamics, ROBC not only reduces end-to-end delay but further improves throughput when compared with LoRaWAN with and without queuing. The improvement in throughput of ROBC is more significant when applied to rural scenarios where node-to-node communication is 1000 m, identical to node-to-gateway communication. The throughput increases by 38% on average for a network with 100 gateways.

Figures 3.9 and 3.10 illustrate the number of messages arriving at the gateways every 10 minutes over 24 hours. As can be seen, in the first environment with a 500m node-to-node communication range, we observe RCA-ETX trades throughput for end-to-end delay performance. Meanwhile, ROBC can reach similar throughput performance as LoRaWAN with queuing while providing lower end-to-end delays. In rural environments, similar results are observed for RCA-ETX, while ROBC outperforms

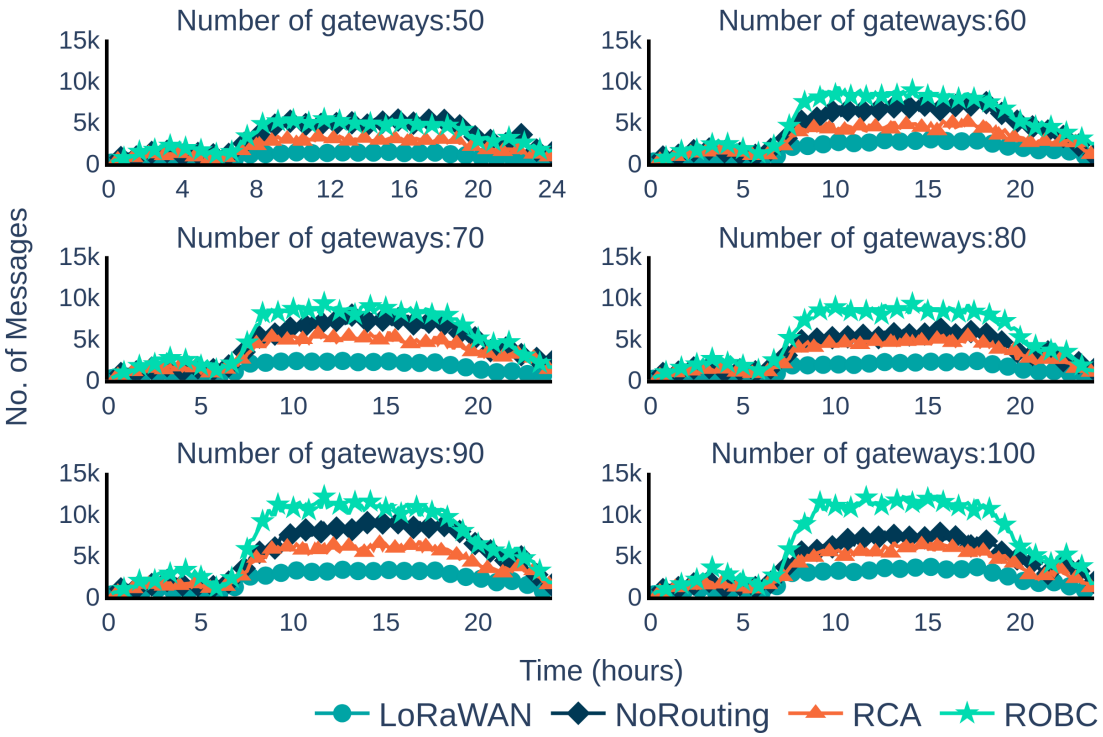


Figure 3.10: Throughput in environments where node-to-node communication range is 1000m.

LoRaWAN with and without queuing in both throughput and delay. Furthermore, higher vehicle density encourages better data forwarding between LoRaWAN nodes, resulting in higher throughput when ROBC is applied. One can easily observe that higher performance gain in throughput for ROBC happens during the daytime (i.e. 20,000 - 75,000 seconds), where more buses are active as shown in Figure 3.6 (a). The throughput achieved is up to 53% and 230% higher compared to LoRaWAN with and without queuing with 100 gateways in the rural environment.

Figure 3.11 shows the average number of hops a message travels before reaching its destination. The figure shows that all LoRaWAN messages have a hop count of 1. The average number of hops with ROBC is higher than RCA-ETX for large number of gateways. The results show that on average a message travels 5-6 hops. Results also show that a gateway has to listen to less number of nodes at the same time as a lot of messages are bundled together before reaching the gateway.

Figure 3.12 shows the average number of messages a node sends. This can approximate the energy overhead. In all our schemes, since the nodes are always listening to a

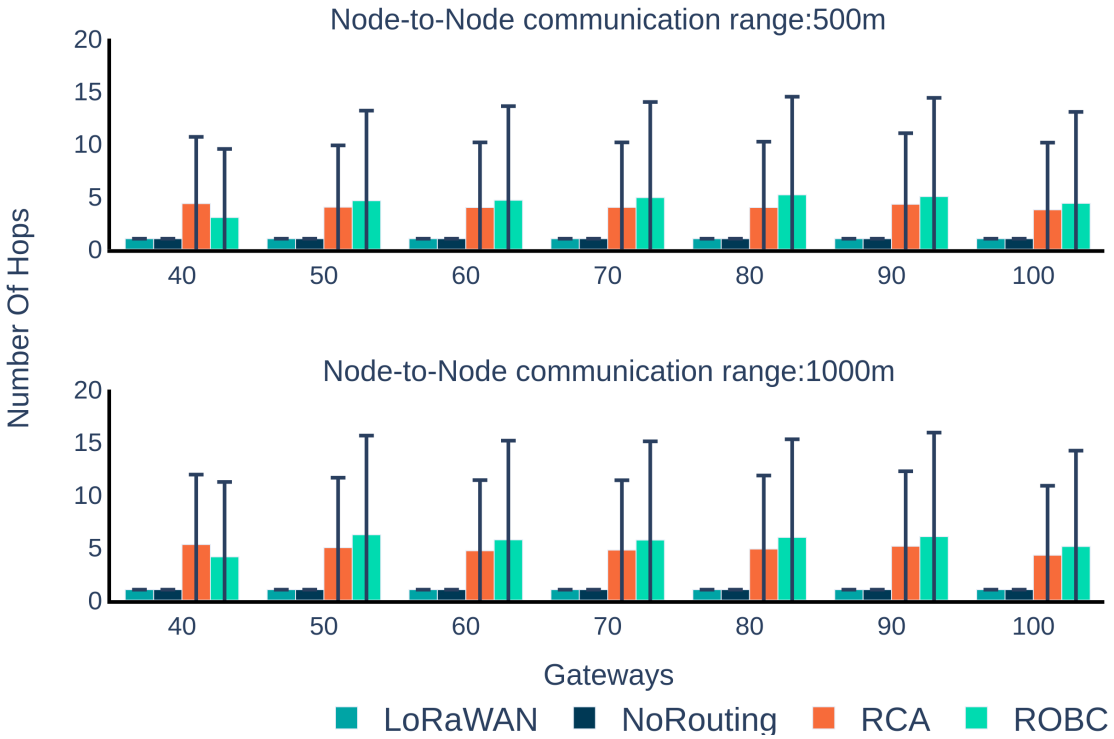


Figure 3.11: Average Number of hops for the entire network

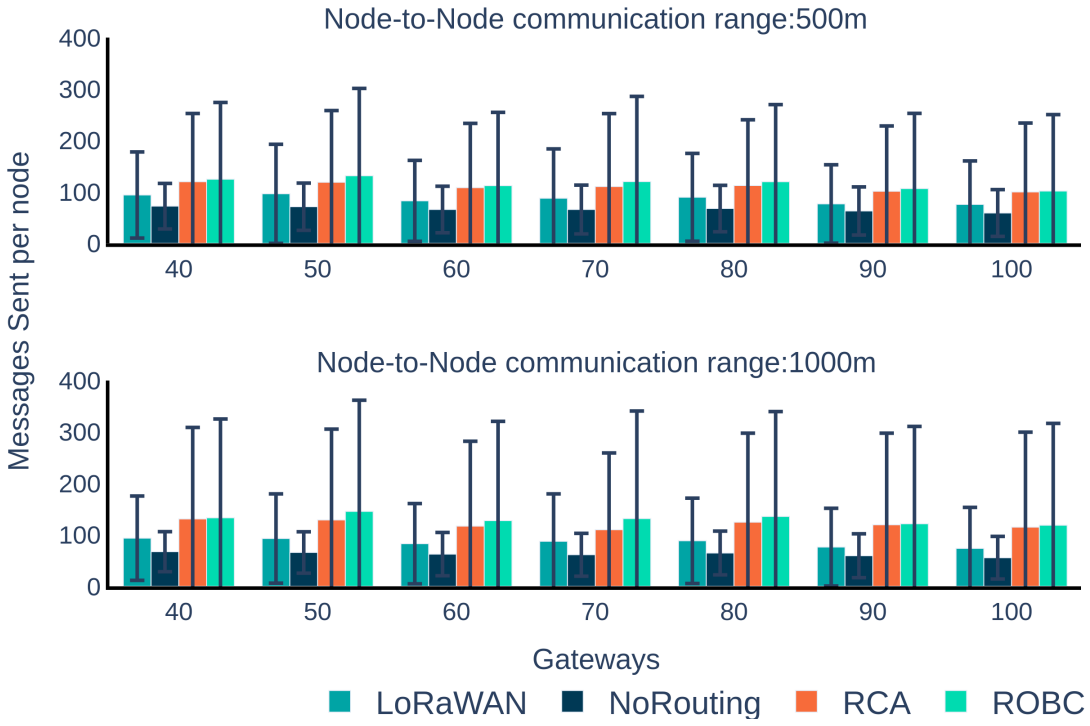


Figure 3.12: Average number of messages sent per node

particular channel to receive messages, we only consider the energy overhead as the additional number of messages sent by a node. The message overhead for RCA and ROBC is in the range of 1.6 to 2.2 times that of LoRaWAN. The network throughput can be improved by 230% by this scheme and so the energy overhead seems reasonable as it significantly reduces the load that a gateway would need to handle and also more data is available at the cyber-layer to take decisions.

3.7 Discussion, Limitations and Future Work

Effect of spreading factor. The chosen scheme uses a fixed spreading factor and one channel. Kousias et al.[97] showed the effectiveness of the adaptive datarate of LoRaWAN reduces as mobility increases. So, it may be better to use a single spreading factor for all vehicles instead of adapting it to the scenario. The scheme is oblivious to the chosen spreading factor or number of channels. As the number of spreading factors or channels used increases, the number of neighbours would reduce. We could in future optimise the system by implementing channel and SF sweeping algorithms to find neighbours using different settings.

Effect of gateway placement. We only report the results with grid-topology gateways, we have also run extensive experiments where gateways were randomly deployed. We observed that gateway locations have a significant impact on performance for these situations. Although ROBC almost always outperforms the other two approaches, significant performance variations were also observed. Since RCA-ETX and ROBC do not have prior knowledge of the mobility and gateway locations, they both rely on *estimated delay* instead of *actual delay*, resulting in errors when computing RCA-ETX values. For example, comparing with a LoRaWAN node which is moving *away* from gateways, a node moving *towards* the gateways might have higher delays (i.e. higher RCA-ETX value) since it had less contact with gateways in the past. Consequently, selecting better gateways positioning could be another valuable research topic where we aim to find the gateway location which can better support mobility and node-to-node data forwarding in both environments.

Effect of high latencies. The results show that the average latencies for RCA-ETX and ROBC are between 2600-5000 seconds and the average latencies for LoRaWAN with queuing is 3000-8200 seconds. As the average bus duration could be much lower than

8200 seconds, a further analysis could be done to study the number of failed messages (i.e. messages remaining in a buses queue at the end of its trip) and whether just hoarding data and transmitting when the bus is in range makes more sense. However, given the latencies of LoRaWAN with queueing, the delays of just holding and transmitting at the end of trip may be much higher than what RCA-ETX and ROBC provide. A further modification to the protocol could be done by introducing data dropping at local nodes, where nodes drop their own oldest data. The protocol could also be extended such that the nodes have two queues, one for their own data and other for neighbours', so they could prioritize their own traffic. All of these aspects need to be studied to understand further the benefits that RCA-ETX and ROBC bring to the system compared to just holding data until next communication opportunity.

This work also assumed acknowledgements have a 100% success ratio and gateways can acknowledge all messages. This assumption does not fit-well as gateways like nodes have to adhere to duty-cycling. We plan to extend this work to add this duty-cycling regulation and also choosing the best gateway to acknowledge a message.

3.8 Related Work

Opportunistic data forwarding in mobile LoRaWAN networks is a relatively new topic. In this section, we briefly review the recent research on opportunistic data forwarding in wireless networks. For an analysis on the mobility in LoRa networks, refer to Section 2.5.

Opportunistic Data Forwarding. In the areas of Internet of Things (IoT) and Wireless Sensor Networks (WSNs), many recent works have focused on finding fast and reliable ways to forward data given a network where the node therein are mobile [117, 118]. One notable direction focuses on data forwarding in networks consisting of static sensors and mobile sinks [119, 120, 121]. In these networks, resource-constrained sensors are typically disconnected from the Internet and have limited communication capability. These sensors therefore try to maximise the opportunity of forwarding data via nearby mobile nodes (e.g. mobile phones) which have a direct connection to the Internet. Some recent works try to find the better opportunistic routes maximising the probability of successful data forwarding [108] without making any assumption on mobility. Yang et al. [105] proposed a new metric, contact-aware ETX (CA-ETX),

to seamlessly merge mobility into classic scheduling protocols (specifically RPL and backpressure routing). However, their work focuses on networks which consist of mobile sinks and static sensors, and it relies on reliable radio channels and high duty-cycle communication between sensors. There are three well-known delay tolerant routing protocols: EPIDEMIC[122], Spray-and-Wait[123], PROPHET[124]. EPIDEMIC requires a large amount of signalling to exchange header information about available packets. Spray-and-Wait reduces the network load by reducing the number of copies of a message in a network by having a time-to-live attached to a message, such that older messages are dropped. The most similar to our approach is PROPHET, where the nodes calculate a delivery predictability metric to decide which nodes to exchange data. All three of these protocols create multiple copies of a message and require large signalling, which would consume the already limited bandwidth available in LoRaWAN.

Another notable direction focuses on data forwarding in disconnected mobile *ad hoc* networks consisting of static sinks and mobile nodes where connections between them are lossy [117, 125, 126]. Some recent studies exploit node contact patterns to reduce delays and improve successful data forwarding[127]. These patterns can be generated based on nodes' historical motion paths [128] and social behaviours[129, 130]. In [128], the authors train a k-th order markov chain to predict future contacts based on historical motion path. It also requires a significant amount of metadata exchange before data handover. RCA-ETX requires no pre-training or past information to operate and only needs two data points from neighbours to make a decision about handover. [131] further takes duty cycles into account by assuming that nodes may not be able to communicate with each other at every physical contact when they can communicate. However, these solutions rely heavily on extra information, such as geographical information and social behaviours, which are challenging to obtain and process for resource-constrained LoRa nodes.

Backpressure routing. Backpressure routing has been used in several wireless algorithms [110, 108, 112]. In [110], the authors propose BCP, a routing protocol built on backpressure routing. The protocol is designed for static nodes and is built for resource-constrained devices. Tahir, Yang, and McCann[108] further extend the work of BCP and propose QuickTheta and QuickBeta, that are designed for time-varying data loads and mobility in wireless networks. However, their work is designed for WSNs which have a higher bandwidth and no duty-cycle restrictions. In [112] authors propose

network coding backpressure routing protocol, along with a cluster-head selection scheme. This allows to reduce the load on one node and distribute the node among all nodes. All of these algorithms rely on high-signalling, are generally designed for static nodes and have no duty-cycle constraints. This is where our approaches of RCA-ETX and ROBC are needed as they are designed for low duty-cycle, resource constrained device with mobile nodes.

3.9 Summary

TCPS or Vehicular networks have varied requirements which are can be met by different communication protocols. A significant amount of research is going on in the V2X networks, in designing the communication protocols, interactions between the entities, reliability guarantees that can be provided etc.

This chapter targeted delay-tolerant applications in vehicular networks. These are applications that benefit from high throughput and reduced delays but do not require hard real-time guarantees. The information can be obtained from private and public vehicles, infrastructure systems but also from the new-age micro-mobility vehicles like e-scooters and bikes. The chapter proposed opportunistic data forwarding between nodes (vehicles) of a network. The data forwarding scheme is done based on a novel metric, namely, *Real-time Contact-Aware Expected Transmission Count*(RCA-ETX), that provides an indication of the connectivity of a node to the gateway. The metric improves resilience of mobile LoRa nodes that cannot communicate to gateways due to unreachable links. It reduced latency while improving throughput for nodes in mobile LoRaWAN network with static gateways.

RCA-ETX reduced the delays but also significantly reduced the throughput. An alternative routing scheme is proposed to improve the throughout. This routing scheme is based on real-time opportunistic backpressure collection scheme and handles the stochastic behaviours arising from node mobility. The experiments are performed on the London bus network as it is a good approximation of the kind of networks in an urban vehicular network. The results show that RCA-ETX effectively reduces delays by up to 25% by trading throughput. Meanwhile, ROBC can provide similar delay reductions with 15% to maximum 53% throughput improvement against LoRaWAN with queuing with up to 2.2 times message overhead.

This chapter focussed on mobile CPS applications running in a smart-city environment where all gateways were communicating with the same servers. However, in more realistic situations, a smart-city environment will consist of multiple simultaneously running applications with non-communicating gateways. The next chapter removes the assumption for a single server to which all gateways can communicate and deals with a more realistic scenario of multiple simultaneously running applications.

4

Increasing Reliability by Leveraging Overlapping Networks

With the increasing number of CPS systems, several applications will run in the same geographical area. These applications will have different sampling and transmission intervals, packet sizes and energy requirements. If these applications are using LoRaWAN, it will lead to 1000s of nodes running using the same communication protocol. To serve these 1000s of nodes, the stakeholders of these systems will deploy their own gateways using the same wireless resource leading to gateways with overlapping communication area. This idea is no longer hypothetical. 10s of gateways serving 1000s of nodes are actively deployed in urban areas. These applications are majorly sensing-only applications with minimal downlink requirements; however, as CPSs become more prevalent the reliability requirements will increase, leading to an increase in the load on the wireless spectrum.

The overlapping gateways will lead to performance degradation for all networks by increasing uplink packet collisions. This performance degradation leads to nodes retransmitting their messages to meet the reliability requirements, which wastes energy and wireless resources. The networks try to optimise their node's transmission parameters, but they do not control nodes belonging to other networks, which does not necessarily increase the reliability.

Opportunity. The opportunity with multi-stakeholder networks is to leverage their overlapping communication ranges to increase the reliability for all. Further, using the wireless medium allows the system to be used with unreliable backhaul networks,

This chapter is based on [43] and [45]. I was the main author. I contributed to the idea, system and algorithm design and writing. I implemented the entire system and drove the evaluation.

commonly found in hazardous environments or rural areas. Wireless also makes the system reactive to a gateway's status and the medium, like unavailable communication bandwidth or change in communication range due to environmental conditions[132].

4.1 Problem Introduction

Problem. LoRaWAN's network deployment model is leading to several applications running in vicinity of each other without knowing about other's existence. The deployment model is leading to performance degradation for all networks using the wireless resource. The duty-cycle regulation reduces this performance degradation by controlling channel access, however, with 1000s of nodes communicating in the same area, the problem still persists.

The problem of overlapping networks in wireless communication is not unique to LoRaWAN networks. Most ad-hoc wireless networks like WiFi, 802.15.4, Bluetooth share the same wireless medium. However, unlike LoRaWAN which uses multiple channels, most wireless technologies only communicate on one channel at any given point in time. These technologies therefore use channel hopping or blacklisting schemes to avoid interference[133, 134].

The problem of overlapping networks in LoRaWAN can be better understood by the following example. Two networks deployed in an area smaller than the reception range of their gateways will overlap. A gateway will receive all of the messages in its communication range and forward it to its server. The server will discard the messages of other networks. Gateways are not a shareable resource. A server cannot receive messages from or transmit downlink messages to a node in its network from a gateway that belongs to another network. Blindly adding gateways[135] and simple retransmissions[84] do not necessarily solve the issue, smarter methods are needed to deal with the available resources.

Solution. In this chapter, we propose ***I**ncreasing **R**eliability of **O**verlapping **N**etworks in LoRa**WAN*** (IRONWAN), a solution to the problems caused by overlapping LoRaWAN networks. It is a software approach that requires no additional backhaul networks, internet access or cloud services and can be readily deployed by only updating the gateways. IRONWAN organises the gateways of multiple networks into a wireless overlay network for gateway-to-gateway (G2G) message exchange (as shown in Fig.4.1).

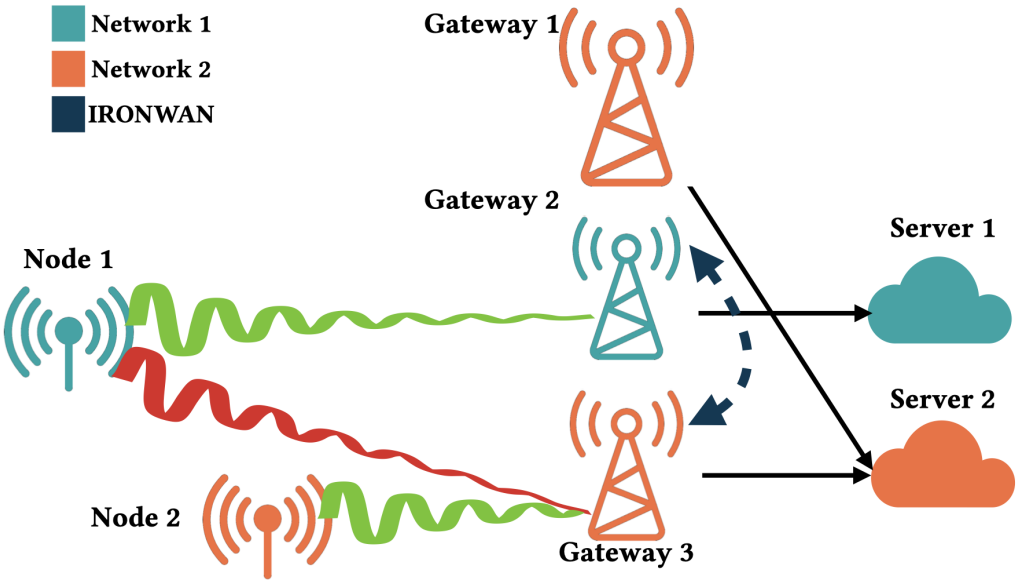


Figure 4.1: IRONWAN creates a wireless overlay network to link overlapping networks (networks 1 and 2) to exchange uplink and downlink transmissions. IRONWAN replays failed messages for uplink collisions(red colour) and can use the link for acknowledgements

IRONWAN allows a gateway from one network to act as a redundant receiver for gateways of another network and receive their uplink messages even when they do not share the same network server without revealing their encryption keys. IRONWAN also enables gateways in the overlay network to share their downlink capacity. Gateways that have exhausted their allocated duty cycle on the downlink channel can send downlink messages via other gateways with free capacity. The sharing of uplink and downlink messages allows all overlapping networks to increase the number of unique messages they can receive per node (which implies they have more data per node) and reduce the number of retransmissions for messages that require acknowledgements which saves a node's energy. As there is no need to share encryption keys there is no need for coordination between the network owners and no chance of the associated security threats. All that is required is the addition of some services on the gateways and no modification to the nodes making the adoption of IRONWAN trivial and anonymous.

Contributions. The main contributions in this work are:

- A LoRaWAN dataset, called LoED collected at 9 gateways consisting of $11,000,000$ packets.

- A Real-time Message Inter-arrival Prediction (RMIP) that gateways can use to adaptively predict message-arrival times from hundreds if not thousands of uplink messages from nodes with only O(n) computation and memory overhead, where n is a user-defined parameter (i.e. window size) that is typically small and independent of network and message sizes.
- To minimise communication interference caused by the G2G communications in IRONWAN, we propose InterPred. InterPred exploits reinforcement-learning techniques to predict the behaviours of other nodes to avoid message collisions during G2G communication in a fully distributed manner.
- We propose IRONWAN, a fully distributed wireless overlay protocol that facilitates G2G communication. Gateways in IRONWAN operates with only local information, and has a low memory overhead.
- IRONWAN is implemented on LoRa gateways for a testbed evaluation and on a simulator for large-scale parametric studies, and demonstrate that G2G communication between different stakeholders is suited for increasing reliability for multiple networks.

4.2 LoED dataset and Preliminaries

This section describes the LoRaWAN dataset LoED and discusses the issues with overlapping networks.

4.2.1 LoED Dataset

To validate the claims of gateways using different resources and design any learning based algorithms, a dataset is needed. We surveryed the existing LoRaWAN datasets available [136, 137, 116]. The fingerprinting dataset [136] contains over $120,000$ traces collected over a three month period. Each trace has the location of the device, the Received signal strength Indicator (RSSI) sampled by the gateways upon receiving a packet, the spreading factor used and the time of the received packet. This dataset provides no information about the gateways such as capacity, load or any information about the deployment environment. The LoRa underground link dataset [137] is

collected in an agricultural environment. It contains data from a low-density LoRaWAN deployment with five transmitters and two receiver base stations. None of the above datasets have a dense amount of data available at the gateways. With this intention, we collected LoED, a LoRaWAN at the Edge Dataset. LoED consists of traces gathered in central London, representative of a mix of dense urban and park environments.

Setup. LoED was acquired from nine LoRaWAN gateways in central London. The gateway locations were representative of typical dense urban and park environments and cover different deployment conditions as shown in Table 4.1. Five outdoor gateways were deployed on the roof tops of large buildings, with a clear line-of-sight (LoS) to devices. Four indoor gateways were located near windows with limited LoS. One of the indoor gateway's was placed on the ground floor of a college dormitory with no-LoS to any device. Each gateway forwarded received packets to a multiplexer which forwarded them to different Network Servers. Our server copied the packets and the gateway metadata to a time-series database. The gateway locations can be found in the dataset and Table 4.1 contains information about the quantity of packets received at each gateway.

Gateway ID	Description of location	Number of days	Total packets	max packets in a day	avg packets per day
00000f0c210281c4	Dense outdoor area, on top of a building	19	1326687	82781	69826
00000f0c22433141	Roof of a low building in a non-dense area	36	144777	6579	4022
00000f0c210721f2	Top of a building in a very dense area and large open spaces	56	5757575	121368	102814
00000f0c224331c4	Indoor in the ground floor a building, surrounded by buildings	15	17029	1625	1135
00800000a0001914	Deployed inside a university building	573	76706	2366	134
00800000a0001793	Deployed inside a university building	552	186592	9596	338
00800000a0001794	Deployed inside a university building	17	61080	4810	3593
7276ff002e062804	Deployed on top of a tall university building, with large open spaces	131	1201916	15254	9175
0000024b0b031c97	Urban area, top of building, dense deployment	131	2490639	25708	19013

Table 4.1: Gateway data and location overview.

The data was collected over a 2-4 month period generated by smart city applications and research deployments. The gateways received from $5,000$ up to $120,000$ packets per day. Overall, $11,263,001$ packets from $8,503$ unique device addresses with a Cyclic Redundancy Check (CRC) and an uplink packet type were collected including packets with failed CRC.

The Data. It was assumed that all collected packets were using the explicit header (EH) mode which enabled metadata extraction of: the device address, packet type, counter value, port number and other information like ADR. Upon packet reception at the gateway the timestamp, physical layer CRC status, frequency, spreading factor, bandwidth, coding rate, sampled RSSI and Signal-to-Noise Ratio (SNR) values were added to the packet information. Each trace consists of the following fields:

time	Time at which the packet was received
physical_payload	Raw payload received
gateway	Gateway where the packet was received
crc_status	Physical layer CRC
frequency	Radio frequency of the packet
spreading_factor	LoRa spreading factor of the packet
bandwidth	Bandwidth of the received packet
code_rate	LoRa coding rate of the packet
rssi	Sampled RSSI value at packet reception
snr	Sampled SNR value at packet reception
device_address	Device address for the packet
mtype	Packet type
fcnt	Counter value of the packet
fport	Port of the packet

The LoED dataset is publicly available at [138] and includes:

- all packets received at the nine gateways,
- one pre-processed CSV data file for every day of the collection campaign. The files are saved in **dd_mm_yyyy.csv** format,
- a set of scripts for processing and plotting the data and,
- a preliminary analysis of the data.

Dataset insights LoED exposes several insights about how LoRaWAN operates in real-world urban deployments. Some of the properties of the dataset are shown in

Figure 4.2. Figure 4.2a shows the LoRa spreading factor usage at each gateway. Of note are the dominate use of spreading factors: 7, 8 (at two of the gateways) and 12. This may increase the probability of collisions as the number of devices using the same spreading factor rises. As a consequence, the performance of the applications running on the devices in the network may decrease and require further, lower-level, investigation. Figure 4.2c shows the usage of frequencies at different gateways. Figure 4.2b shows the distribution of the frame type. The results show that the receive for gateways is dominated by the uplink data types (0,10,100).

LoED dataset is used in designing IRONWAN's algorithms.

4.2.2 Exploiting Gateways from other Networks

This subsection discusses the impact of overlapping networks with multiple stakeholders and helps better understand the problem of multi-stakeholder overlapping networks.

To study the effect of multiple overlapping networks, a 1000 node network is simulated. The nodes generates a packet every 3 minutes. The impact on the packet delivery ratio (PDR) and the number of unique uplink messages per node received at the server is studied. For a detailed description about the simulation environment and the metrics see Section 4.6. Figure 4.3 shows the results for a non-overlapping network with 1 owner and 4 overlapping networks with 4 owners with the same network topology. The networks are tested under three levels of message acknowledgement requirements (i.e. low:10%, medium:50% and high:90%). All three levels represent how LoRaWAN networks will move from uplink only transmissions to a more CPS-style requirement space. At all three levels, the 1-owner network outperforms the networks with 4-owners. Given networks consisting of 6 gateways, the number of unique messages received per node and PDR of the 1-owner network is 65% and 50% higher than the network consisting of 4 different stakeholders without gateway sharing. Our analysis shows that networks with 4-owners suffer from extensive message collisions caused by message retransmissions or lack of available duty-cycle to acknowledge these messages. The objective of this work is to introduce the performance advantages of 1-owner networks to networks with multiple owners by sharing gateway resource between multiple owners.

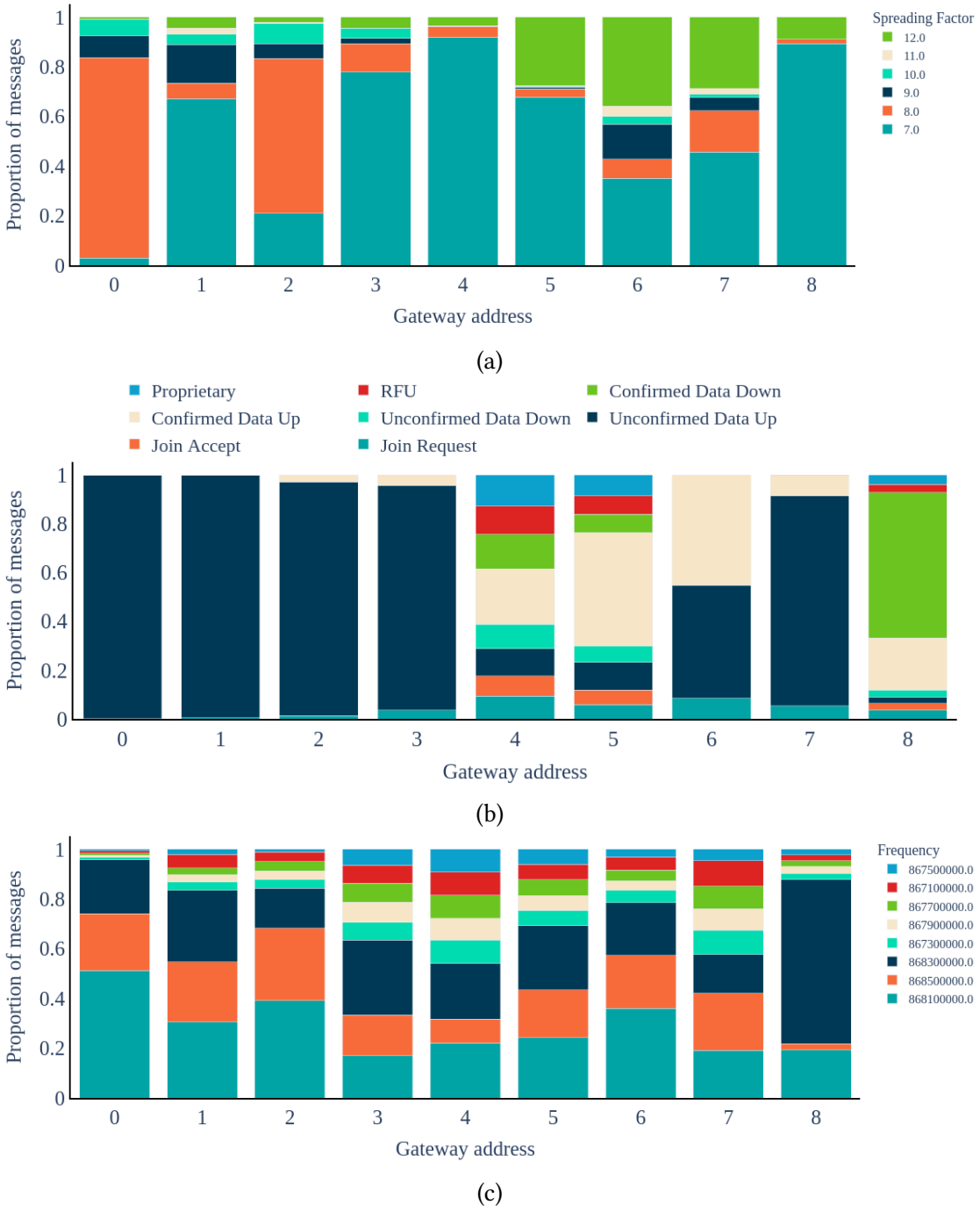


Figure 4.2: (a) Distribution of spreading factor at each gateway (b) Distribution of message types at each gateway (c) Distribution of frequencies at each gateway

This section discussed the LoED dataset and the problems that arise because of multi-stakeholder overlapping networks.

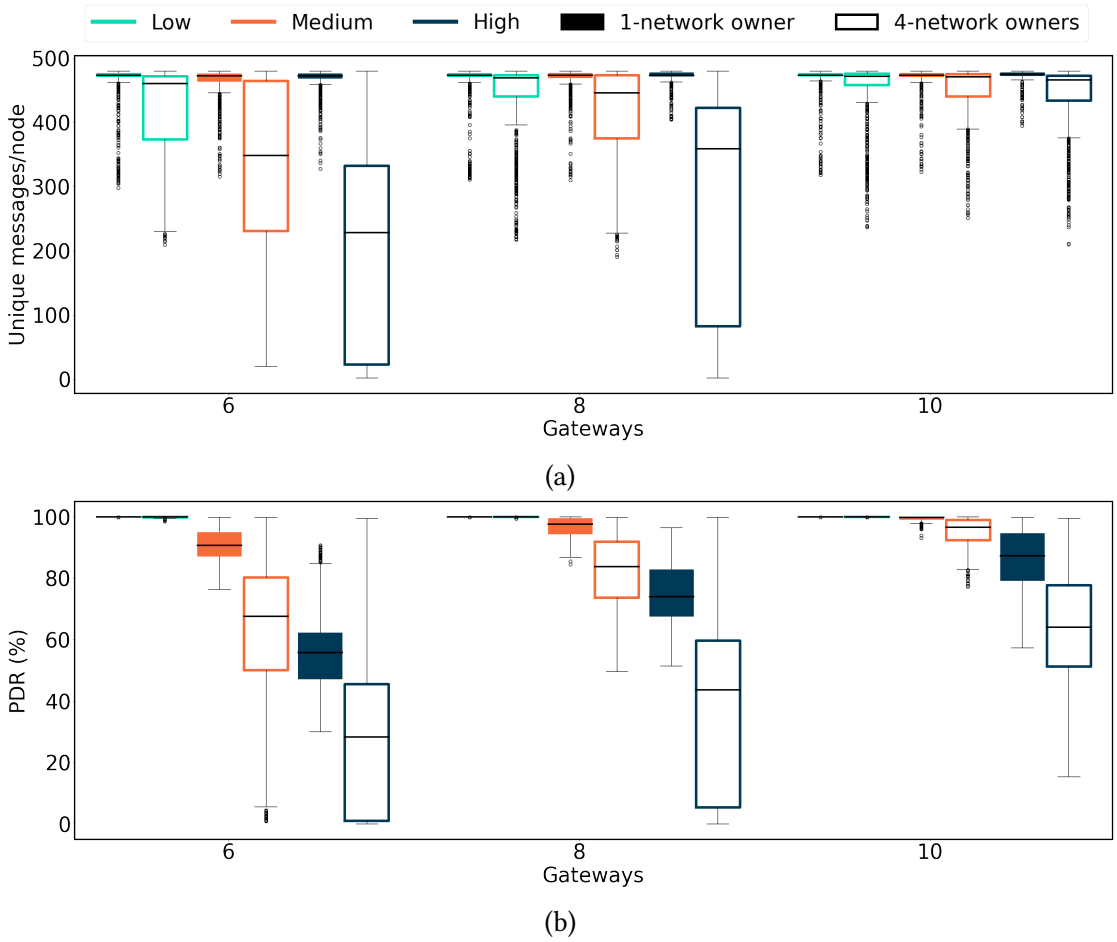


Figure 4.3: Effect of partitioning in a LoRaWAN network

4.3 IRONWAN Overview

IRONWAN is a G2G communication system that solves the message loss and downlink duty-cycle exhaustion problems caused by overlapping LoRaWAN networks. It does this by enabling gateways to coordinate to find lost uplink messages and share downlink capacity. Importantly, IRONWAN schedules G2G communication to minimise interference to all of the networks. IRONWAN runs as an add-on module on LoRaWAN gateways, is fully compatible with the LoRaWAN specifications, does not require any backhaul networks, does not incur cost for usage, and is readily deployable with a software update on the gateways. In this section, we present an overview to IRONWAN's architecture and the four sub-modules forming IRONWAN. We then provide an operational overview describing the interactions between modules, and the objectives

and challenges for IRONWAN.

4.3.1 IRONWAN Architecture

IRONWAN consists of four sub-modules: a manager, a cache and two learning algorithms that run on every IRONWAN-enabled gateway as shown in Fig.4.4. They are:

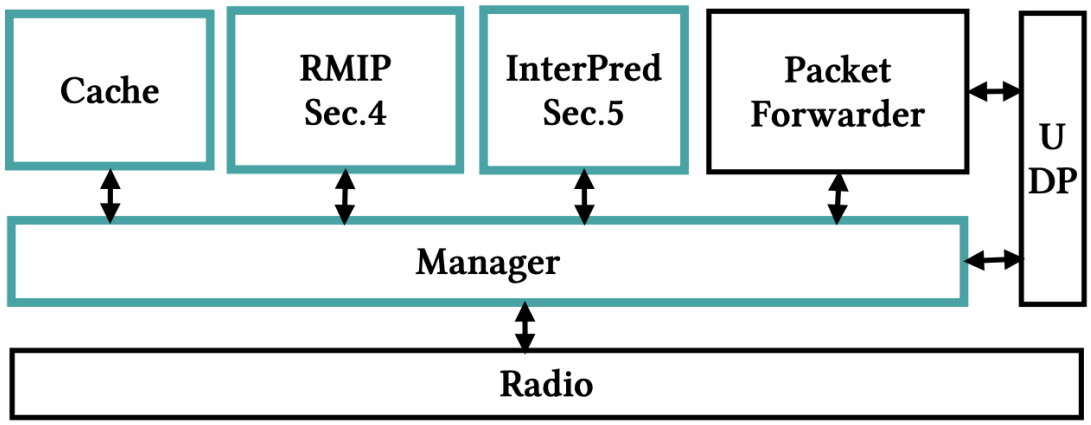


Figure 4.4: Interaction between LoRa radio, packet forwarder and IRONWAN's modules (shown in colour)

1. **Manager module** facilitates gateway communication. This module is responsible for scheduling and orchestrating access to the wireless channel for a gateway, tracking channel usage, querying RMIP and InterPred when needed and handling incoming G2G messages from other gateways. The Manager module is also responsible for requesting lost uplink messages (by querying RMIP), handing over downlink messages and ensuring that any G2G communication causes minimal interference (by querying InterPred). IRONWAN also introduces new G2G messages that are not defined in standard LoRaWAN and so the manager module is also responsible for prioritising and scheduling these messages. It also forwards messages received from the radio layer to the appropriate modules including the Packet Forwarder (part of the LoRaWAN specification). No encryption keys need to be shared by the manager modules of different gateways.

2. **Caching Module** caches the latest data packages received from nearby nodes within communication range. Upon receiving data requests (for lost messages)

from other gateways, the manager module accesses this cache to find the requested messages. The data stored in this cache are deleted after a user-defined time to limit memory consumption.

3. **Real-time Message Inter-Arrival Predictor (RMIP):** estimates the message-arrival time for every node within the reception range. If an expected message does not arrive in time, it triggers the manager module to request the lost message from gateways belonging to other networks, or the manager module can poll this module to get information about lost messages, see Section 4.4.

4. **Interference Predictor (InterPred):** predicts if a G2G communication can cause interference through training a value-based reinforcement-learning agent which learns channel usage in real-time. By querying this module, the manager can choose time-slots and channels for G2G communications without causing interference to N2G communication, see Section 4.5.

4.3.2 Operation Overview

Every gateway has to perform three operations alongside normal LoRaWAN functionality to be IRONWAN-compliant. The first is to duplicate and forward the received messages correctly. The second is to request for failed uplink messages if they haven't arrived. The third is to handover downlink messages if a gateway has no downlink capacity. These operations are described in detail below.

Received Messages. As shown in Figure 4.5a, the manager module forwards all valid LoRaWAN messages to the cache module, RMIP module, InterPred module and the packet forwarder. The packet forwarder sends the message to the server as per the LoRaWAN specification.

Request for a lost uplink message due to uplink packet collisions. As described in Figure 4.5b, a gateway's RMIP module indicates to the manager that an uplink message from a node was lost (1). The manager then sends a *Request for uplink message* with the node's address and last received message ID (3). The manager module of the receiving gateway (4), queries its caching module for a newer message than the message ID for that particular node (5). If there is such a message (6), the gateway responds to the requesting gateway with a *Rebroadcast an uplink message* (7). These messages are

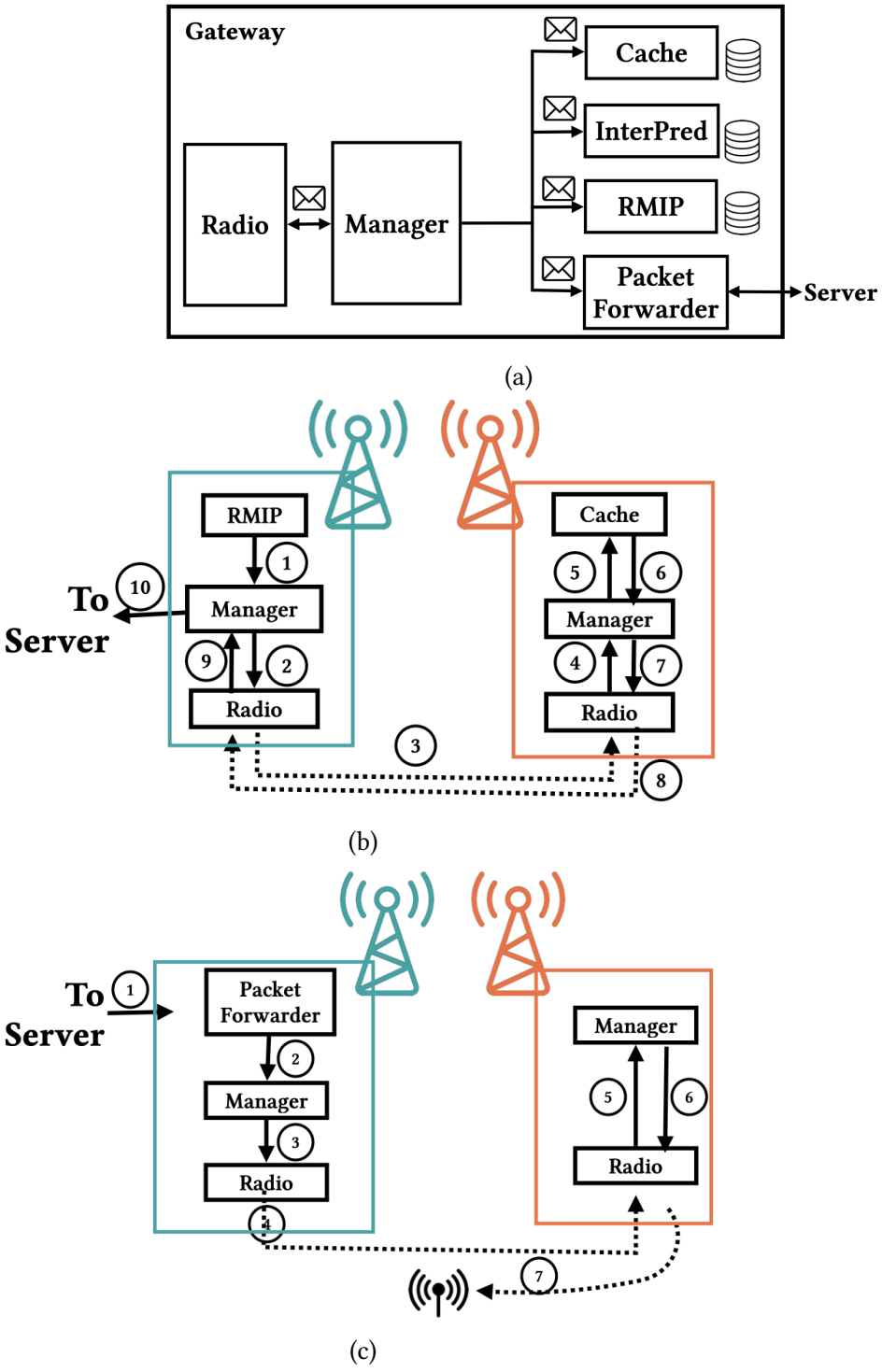


Figure 4.5: (a) All received messages are replicated by manager and forwarded to all modules, (b) IRONWAN's G2G communication for a gateway to request a lost uplink message from a neighbour gateway. (c) IRONWAN's G2G communication for a gateway with no remaining downlink duty-cycle to request that a neighbour gateway send a downlink message.

received by the gateway (9) and forwarded to the server (10). Essentially, the gateways act as store-and-replay intermediaries and this increases overall network throughput.

Request to send a downlink message due to duty-cycle exhaustion on downlink channels. As shown in Figure 4.5c, IRONWAN enables a gateway with no remaining downlink duty-cycle to handover the transmission of downlink messages to other gateways. Whenever the manager receives a message from packet forwarder (1), and it cannot transmit it, the requesting gateway encapsulates the packet in a *Request to forward Downlink* message and sends it (3). If a gateway belonging to another network receives this message, it checks its cache to verify if it has received a message from that particular node in the last two seconds. If the gateway has received a message from that node, it gets the time of when to transmit the message from its cache, and it schedules and transmits a *Neighbour Downlink Message* (6).

LoRa physical layer exhibits capture effect[79]. If two or more messages are transmitted at the same time, a message can still be successfully decoded and it depends on the Signal-to-Noise Ratio (SNR) for the receiving message. In IRONWAN, if two or more gateways transmit at the same time a *Neighbour Downlink Message*, they will arrive at the nodes at the same time. If the SNR levels are such that the message can be received successfully, there will be no collision [139]. However, if there is a collision (SNR levels not within desired levels [139]), the message will be lost and it would trigger a retransmission at the nodes end.

4.3.3 Objective and Challenges

IRONWAN solves the problem caused by overlapping networks by exchanging failed uplink and downlink messages between gateways. IRONWAN uses LoRaWAN for G2G communication. LoRaWAN operates in in the sub-GHz unlicensed band which is subject to a 1% duty-cycle on band 0 (868.0-868.8 MHz for both uplink and downlink) and a 10% duty-cycle on band 1 (869.40-869.65 MHz for downlink) in EU. As the access to the channel is duty-cycle limited, this makes communication a scarce resource and so message transmissions need to be planned and scheduled to efficiently use the spectrum. Gateways are also resource-constrained devices (e.g. raspberry pi) and may connect to hundreds if not thousand nodes at the same time. It is essential to ensure all of the solutions are light-weight and can scale to large networks. This creates two new

challenges:

1. Deciding when to hunt for missing messages. We aim to solve the problem of uplink packet collisions by requesting the gateways of other networks to retransmit a message that failed to arrive at its own network as is described in Subsec. 4.3.2. Gateways have to accurately estimate when to expect message arrivals for every node in their communication range. To understand if this could be estimated, we analyse 11-million real-world LoRaWAN messages to see if there was an observable trend, using the LoED dataset[45]. The data showed that message inter-arrival times per node are relatively consistent; 56% of the nodes in the dataset sent messages periodically, 5% of these nodes changed their transmission period at some point because of application layer changes or the reassignment of node ids, and only 4% of the nodes which transmit messages periodically require acknowledgements. This is in line with the results in [140]. Their results show that around 65% of the nodes are transmitting with intervals that have less than 10 seconds of standard deviation. These results show that gateways can predict the message-arrival time for nodes that send messages periodically. An additional solution is also required to cope with changes in inter-arrival periods and errors from undefined behaviours that could deviate from our predictions.

2. G2G communication scheduling. Whenever a G2G communication occurs to exchange uplink or downlink messages, it can only occur on band 0 as gateways only listen on that band. G2G messages will interfere with node-to-gateway (N2G) communication on band 0. A gateway can choose from several channels and times on this band to transmit a packet. If it transmits at the wrong time on a wrong channel, it will interfere with a N2G message which would lead to a lost packet, triggering a retransmission. We need an effective, light-weight solution to predict and avoid interference so gateways can schedule G2G transmissions which cause the least interference. The solution needs to take into account the changes in the environment like varying transmission periods and network conditions that affect communication parameters.

This section described the objectives and challenges of IRONWAN, IRONWAN's components, how they interact with each other and the communication protocol. The next two sections dive deeper into the design and evaluation of RMIP and InterPred modules.

4.4 RMIP Design

This section describes how the RMIP module estimates the message inter-arrival time for every node. If a message is not received in time (relaxed with a tolerant delay threshold), the gateway requests this possibly lost message from its neighbouring gateways. RMIP uses techniques from streaming and statistical methods. The computation and memory overhead of RMIP are both $O(n)$, where n is a buffer size that determines how fast RMIP responds to changes in inter-arrival times. The value n is typically very small. A gateway only needs $(n+1)*4$ bytes (assuming 32-bit floats) per node to predict message arrivals from hundreds if not thousands of nodes using low memory and computation.

4.4.1 Problem Statement

Assume a gateway with a set of nodes in its reception range. The nodes transmit messages periodically with an inter-arrival time Δt. Figure 4.6(a) shows that when every uplink message successfully arrives at a gateway on the first try (i.e. no retransmissions), the messages arrival time can be seen as a time series $\langle t_1, t_2, \cdots, t_l \rangle$, where $\{1, 2, \cdots, l\}$ are monotonically increasing message IDs. Figure 4.6(b) shows that for messages that require acknowledgements nodes will try to retransmit the message when a message delivery fails before the next message is generated. For messages that do not need acknowledgements, this message is lost. The actual arrival time at the gateway, shown in Figure 4.6(c), becomes $\langle t_1+d_1, t_2+d_2, \cdots, t_m+d_m \rangle$, where $\{d_1, d_2, \cdots, d_l\}$ are random numbers in range $[0, \Delta t)$ denoting the delays incurred because of these retransmissions. As messages may be missing and these delays are unknown to the gateway, message delivery intervals may vary and become $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_m \rangle$.

Our objective is, given the *observed* message inter-arrival times $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_m \rangle$, to predict the *actual* message arrival times $\langle t_1, t_2, \cdots, t_l \rangle$ at the gateway. The problem can be futher broker down into three sub-problems listed below:

- **Inter-arrival time prediction:** The first task is to predict the actual inter-arrival time Δt given $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_l \rangle$.

- **Reference-anchor prediction:** The second task is to find a reference point from where this inter-arrival time is valid. To predict $\langle t_1, t_2, \cdots, t_l \rangle$ from inter-arrival

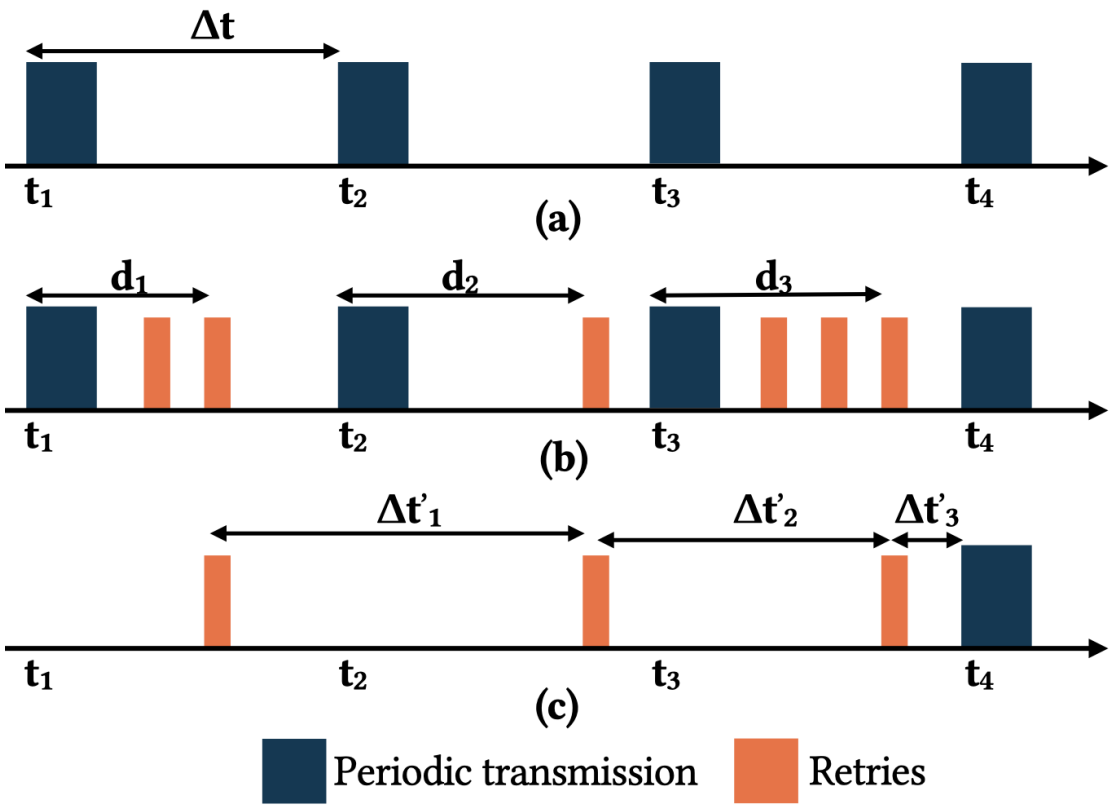


Figure 4.6: Examples of: (a) uplink messages sent and received by a node and the gateway, respectively, without retries. (b) uplink messages sent by a node with retries. (c) uplink messages received at the gateway with retries.

time Δt, we find a reference time point $t_{\varnothing}$ where a message arrives with *zero* retries.

- **Change detection in inter-arrival time:** The inter-arrival time Δt may change overtime. Our predictions based on previous observations may be skewed if the gateway is not aware of these changes. Consequently, a gateway needs to detect changes and adapt accordingly given $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_l \rangle$.

It is worth noting that, each LoRaWAN gateway may connect to thousands of LoRaWAN nodes and it is essential to minimise the extra overhead introduced by RMIP.

4.4.2 Estimating $\Delta t'$ for missing messages

Before starting the prediction algorithm, RMIP first checks if each data point in $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_m \rangle$ is computed from messages with two consecutive IDs (e.g. $\Delta t'_1 = t'_2 - t'_1$). In practice, some messages may go missing and never reach the gateway (delay

$d_i > \Delta t$) due to interference. The inter-arrival time computed from a message stream with lost messages can significantly deviate from the ground truth. For example, when a gateway only receives every alternate message, the inter-arrival estimation can be two times larger than the ground truth. To deal with missing messages RMIP fills in these missing $\Delta t'_i$ using the equation below:

$$\Delta t'_i = \frac{t'_{l+m} - t'_l}{m} \forall l < i < l + m$$

where l and $l + m$ denotes the message ID of the recent and last received messages, and m denotes the number of missing messages.

4.4.3 Inter-Arrival Time Prediction

Once the missing inter-arrival times are estimated, the next task is to estimate Δt given $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_l \rangle$. To do so, we have a closer look at the message inter-arrival time observed in real-world scenarios. The following property can be observed from the example shown in Figure 4.6:

$$(l-1)\Delta t < \Sigma_{i=1}^{l} \Delta t'_i < (l+1)\Delta t, \tag{4.1}$$

where l is the number of messages expected. Equation (4.1) shows that the error term is bounded by $\pm 1/l$ when using the *mean* as the estimator. However, using the mean has several drawbacks. First, this error term is not equal to zero unless the first and last messages both arrive without delay. This is a problem in practical scenarios where delays due to collisions, retransmissions or clock jitter are common. The estimation may deviate from the actual inter-arrival time. Second, the mean value is more sensitive to outliers, and a large delay may significantly skew the estimation. Using the *median* as the estimator makes the algorithm robust against the above-mentioned problems.

The median may still deviate from actual intervals when there are no *exact* inter-arrival times in observed samples. We overcome this problem by performing a statistical test on the inter-arrival estimate. Due to the small sample size (n) ($n = 10$ in our implementation), RMIP uses student's t-test[141] to test if we should use the computed median value as Δt. T-test produces a p-value (with regards to its degree of freedom) which we test to see if we can reject the null hypothesis (i.e. median is not valid). A

more restrictive p-value is adopted (0.703 representing a 50% two-side quantile when $n = 10$) for our implementation. If the null hypothesis is rejected RMIP accepts this median as the estimated inter-arrival time (Δt), otherwise it continues to collect new inter-arrival time samples while dropping the oldest one until the null hypothesis is rejected.

4.4.4 Reference-Anchor Prediction

Our next task is to find the reference time point $t_\varnothing$ with which $\langle t_1, t_2, \cdots, t_l \rangle$ can be acquired. RMIP uses a simple but efficient method to predict $t_\varnothing$ using the equation below:

$$t_\varnothing = \begin{cases} t_n & t_n - t_\varnothing < n\Delta t, \\ t_\varnothing & \text{else}, \end{cases} \tag{4.2}$$

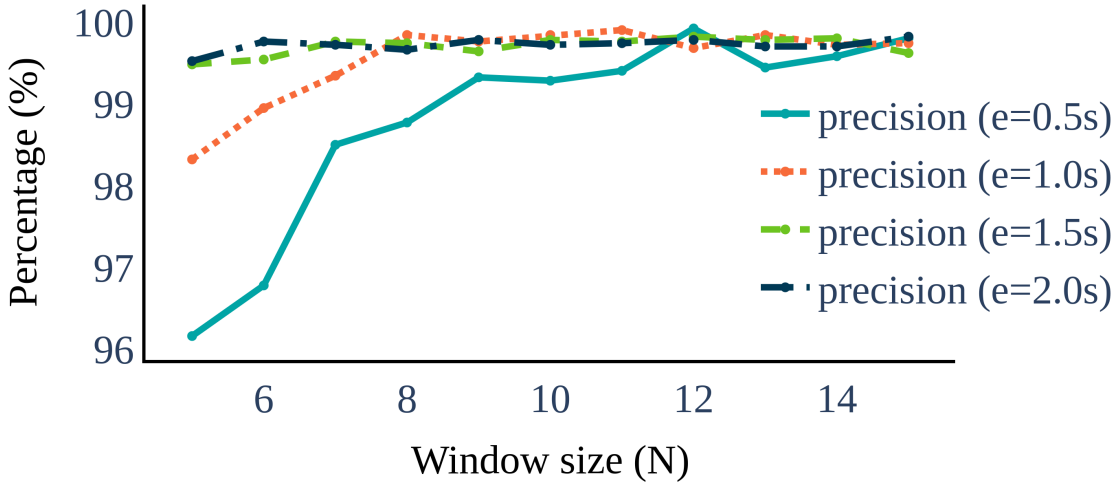
where n is the message counter values from $t_\varnothing$. As can be seen, it resets $t_\varnothing$ when messages are received before expected arrival time. This is valid because LoRaWAN message counter values are reset only when a new message is generated. This allows $t_\varnothing$ to quickly converge to the *first* transmission for any message of every node.

4.4.5 Change Detection in Inter-Arrival Time

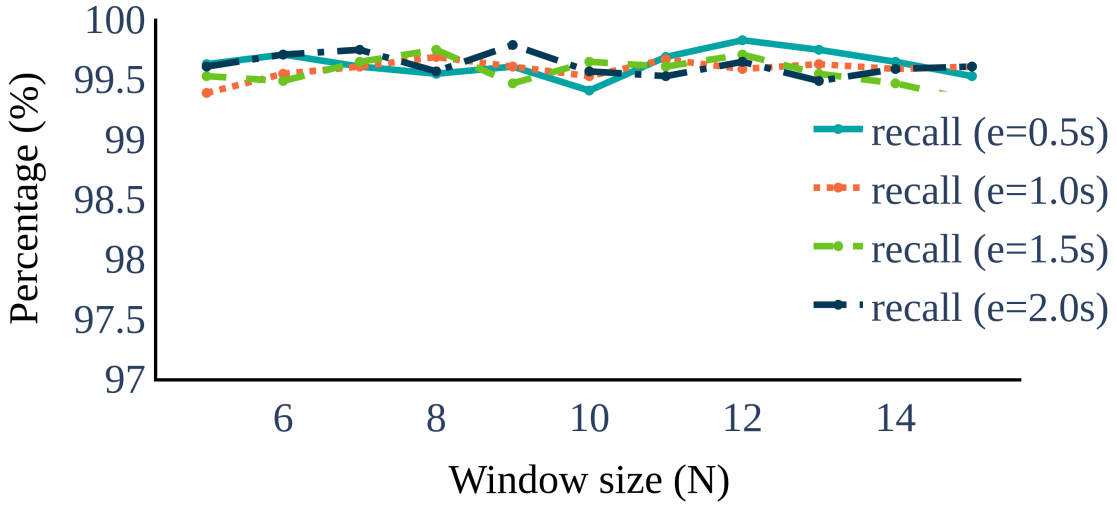
RMIP adopts the event trigger technique that is commonly seen in stream processing to minimise memory and computational requirements and detect changes in the inter-arrival time. Instead of updating Δt every time a new message is received, it compares the new inter-arrival time with Δt. If RMIP collects n *consecutive* inter-arrival times $\langle \Delta t'_1, \Delta t'_2, \cdots, \Delta t'_n \rangle$, where their difference with Δt is greater than a given threshold e (i.e. $|\Delta t - \Delta t'_i| > e, \forall i \in \{1, 2, \cdots, n\}$), it assumes that a node's transmission interval has changed. RMIP then recomputes the Δt and $t_\varnothing$, with the algorithms presented in Sections 4.4.3 and 4.4.4, respectively.

To better understand the impact of n and e on RMIP, we run an experiment on LoED. As the ground-truth changes are not available in this dataset, changes were simulated by concatenating inter-arrival timeseries of different nodes (to simulate artificial inter-arrival changes). The results are shown in Figure 4.7. As can be seen, RMIP captures changes with a very high accuracy. Precision and Recall are more than 96% in all of our experiments. There is no discernible changes in recall with respect to

both n and e. Precision improves with larger n and e; however, the difference become negligible when $e \geq 1.5$s. In our experiments, we choose $n = 10$ and $e = 1$s as default. This is because LoRaWAN devices open two receive windows 0.9-1.1 and 1.9-2.1 seconds for acknowledgements from gateways as presented in Section 2.4.3.



(a) Precision



(b) Recall

Figure 4.7: The precision and recall of the change detection in RMIP given given different window size n (5-15) and error threshold e (0.5s-2.0s)

4.5 Interference Prediction

In this section we describe InterPred. Each gateway using IRONWAN has an InterPred agent, when requested, determining a channel and timeslot for G2G communication that will not interfere with N2G communication.

4.5.1 InterPred Overview

InterPred trains an *interference predicting agent* on each gateway by overhearing the network traffic that reflect the communication usage of the neighbouring gateways and nodes. The agent learns an interference model to decide the channel and timeslot for G2G communication that will not interfere with other communication. To formalise LoRaWAN interference prediction as a *reinforcement learning* problem, InterPred introduces novel definitions of state, action, policy, and reward, and uses the value-based reinforcement learning method, State-Action-Reward-State-Action (SARSA)[142], to train the gateway agent. We use SARSA because of its merits for embedded applications (i.e. fast, efficient, and no pre-trained model requirement). It is non-trivial to use SARSA for agent training to predict wireless communication. Most wireless communication systems (including but not limited to LoRaWAN) are *half-duplex*, i.e., messages can be either sent or received on a channel but not both simultaneously. A half-duplex system does not have the immediate feedback about its success or failure required by SARSA. To address this issue InterPred trains the agent based on *pseudo actions* (discussed in Section 4.5.3).

The workflow of InterPred is illustrated in Figure 4.8. An InterPred agent on a specific gateway chooses an *action* (i.e. decides the channel and timeslot inducing the minimal interference with N2G communication, and conducts G2G communication correspondingly) given a specific *state* (i.e. the network status defined by spectrum usage information during a past period of time). Since the optimal acting *policy* (i.e. best actions to take in different states) is unknown after the system initialisation, the agent continuously interacts with its *wireless environment*, and iteratively optimises its policy according to the environment *reward* determined by the interference caused by actions taken. We present detailed definitions as follows.

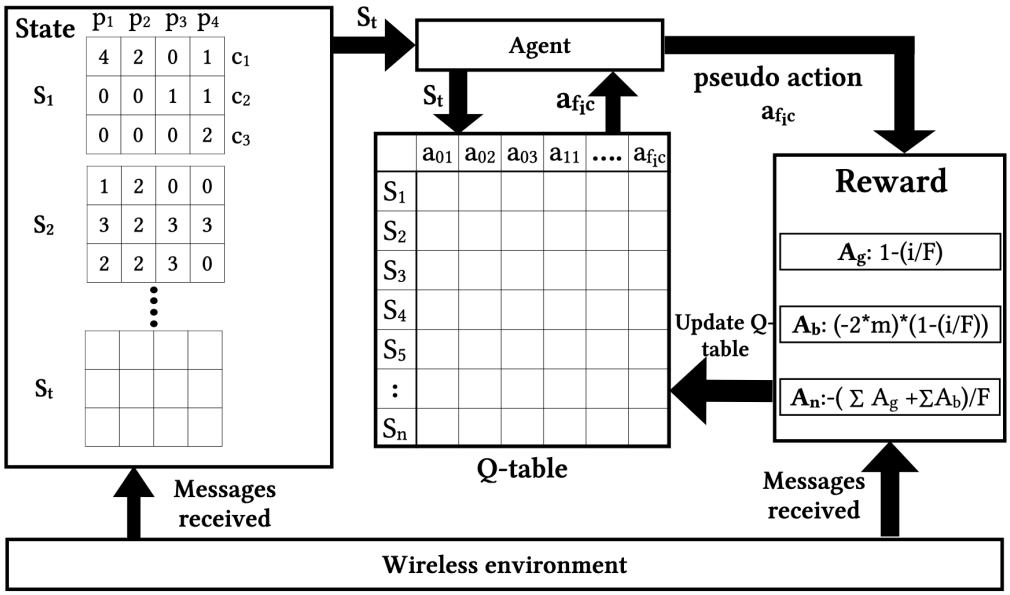


Figure 4.8: InterPred overview

4.5.2 State, Action, Policy, and Reward

Formally, we assume that IRONWAN operates in discrete timeslots $t \in 1, 2, \cdots$ of 0.1 seconds. InterPred stores the communication spectrum usage information of up to P past slots. When a gateway needs to conduct a G2G communication, it uses the information collected in these P slots to choose one slot from future F slots on one of C channels to communicate with the least possibility of causing interference with other N2G communication. The gateways have no information about interference at other gateways, and they can only make locally optimal decisions.

State. At a specific timeslot, we define the current *State* $\boldsymbol{S}$ as a matrix of information about messages received on each of C channels in each of past P timeslots (see Figure 4.8). Each timeslot is labelled as p_i, $i \in \{1, 2, \cdots, P\}$, and each channel is labelled as c_j, $j \in \{1, 2, \cdots, C\}$. p_1 and p_P represent the oldest and the current timeslots in $\boldsymbol{S}$, respectively. Each state matrix element $s_{p_i c_j}$ contains information about the number of messages received on c_j at p_i. Whenever the gateway receives a message in p_P on c_j, it extracts information about the airtime of a packet, and calculates the number of timeslots where packet reception was undergoing (k) by dividing airtime with slot

length (0.1 seconds). $s_{p_i c_j}$ is then updated as:

$$s_{p_i c_j} = s_{p_i c_j} + 1, \quad i \in [max(0, P - k), P]. \tag{4.3}$$

We use this approach as the gateway's radio only forwards a message when it has been completely received. The maximum value of $s_{p_i c_j}$ is bounded by 5, which implies that c_j is congested and not suitable for transmission. Such a bound objectively reflects the property of LoRaWAN communication, and it helps to restrict the state space for better tractability.

Action. In a specific state $\boldsymbol{S}$ at timeslot p_P, if G2G communication is required, the InterPred agent needs to take an action. We define an *Action* a as the gateway conducting a G2G transmission at a future timeslot $f_i \in \{0, p_P + 1, p_P + 2, \cdots, p_P + F\}$ on channel $c_j \in \{c_1, c_2, \cdots c_C\}$. Note that the agent only decides to transmit in one of next F timeslots, and $f_i = 0$ represents the case where no transmission is conducted. Specifically, $a_{f_i c_j}$ denotes transmit at timeslot $f_i = p_P + i$ on channel c_j.

Policy. In state $\boldsymbol{S}$, the InterPred agent needs to choose an action from $((F + 1) * C)$ available candidates to minimise the interference on N2G communication. To achieve this, we use a *Q-value* ($Q_{\boldsymbol{S},a}$) to denote the impact of taking action a in state S on the N2G communication interference. A higher Q-value represents less interference. The *Policy* of an InterPred agent is a table of the Q-value of each action in each state, or a *Q-table* (see Figure 4.8). Agents take actions in an ϵ-greedy manner according to the Q-table, i.e., selects the action with the highest Q-value with a probability of $1 - \epsilon$, or a random action with a probability of ϵ. $\epsilon \in [0, 1]$ can be adjusted for a desirable trade-off between exploitation and exploration.

Initially, all elements in the Q-table are assigned with 0 since there is no prior information about each action's impact. The InterPred agent updates the Q-table by iteratively taking actions according to its latest policy and the feedback from the wireless environment. Each element $Q_{\boldsymbol{S},a}$ is updated as follows:

$$Q^{new}_{\boldsymbol{S},a} = Q_{\boldsymbol{S},a} + \alpha * (reward + \gamma * Q_{\boldsymbol{S}',a'} - Q_{\boldsymbol{S},a}). \tag{4.4}$$

Here, $\boldsymbol{S}'$ denotes the state transferred from $\boldsymbol{S}$ after a is taken, a' denotes the action taken in $\boldsymbol{S}'$ according to the current policy, $\alpha \in [0, 1]$ is the learning rate that controls the stepsize of Q-value update, and $\gamma \in [0, 1]$ is the discount factor that controls the weight

of future Q-value. Both α and γ are empirically tuned to improve the performance of the learnt policy under our experimental settings. *Reward* is a real value quantifying how to adjust the Q-value of each action in a state considering its impact.

Reward. Intuitively, if an action causes no interference, we call it *good*, which should impart a positive reward. Similarly, a *bad* action causing interference should impart a negative reward. Moreover, if *no* G2G transmission is conducted, the agent should be punished or rewarded based on whether potential chances are missed or not. Therefore, we have following definitions:

- **Reward of a good action (A_g):** If action a_{f_ic} interferes with no N2G message, its reward is:

$$reward_{a_{f_ic}} = 1 - i/F. \tag{4.5}$$

  This encourages the gateway to carry out G2G communication as soon as possible.

- **Reward of a bad action (A_b):** If action a_{f_ic} interferes with m N2G messages, its reward is:

$$reward_{a_{f_ic}} = -2 * m * (1 - i/F). \tag{4.6}$$

  This introduces a penalty to discourage the gateway from taking bad actions in the current timeslot.

- **Reward of a no-transmission action (A_n):** If action a_{0c} is taken, its reward is:

$$reward_{a_{0c}} = (\sum A_g + \sum A_b)/F. \tag{4.7}$$

  where, $\sum A_g + \sum A_b$ represents the total reward imparted by all other actions, implying whether it was a lost transmission opportunity or a good choice to not transmit.

4.5.3 Agent Training based on Pseudo Actions

The real challenge of training the InterPred agent with SARSA is that, LoRaWAN communication is half-duplex. The gateways can only transmit or receive on a channel at any given point, which makes it impossible to get feedback to calculate rewards.

Pseudo actions are used for training to address this issue. Once deployed on a gateway with the initial policy (i.e., all Q-values are 0), the agent first enters the *training phase*,

where it iteratively selects actions in different states according to its policy. However, no G2G communication is actually conducted. In the meantime, the gateway keeps overhearing all communications on all channels, and continuously provides the agent with the wireless spectrum usage information. With this, interference that **would have been caused** by corresponding actual actions can be inferred and used to update the Q-table according to Equation 4.4.

After trained based on pseudo actions for 3 hours in a 24-hour simulation, the agent is able to start actual predictions. When the gateway needs to transmit a G2G message the agent determines a timeslot and a channel according to its Q-table for G2G transmission. The Q-table is not updated after a real transmission. When there is no G2G transmission request the agent continuously learns based on pseudo actions. There may be no transmission decision that the agent can take that gives it a positive reward. In this case, the agent will take the no-transmission action, and the gateway will not send a G2G message. This lack of a decision helps InterPred to deal with network conditions of extreme overcrowding to prevent complete resource starvation.

4.5.4 InterPred Validation

InterPred is implemented and compared with two naive policies, namely, *random* and *next-used* to validate the correctness of the chosen method. Three agents with different prediction methods were placed under the same virtual wireless environment (defined by the same set of simulated and real-world datasets). They were evaluated in terms of *fulfilling G2G communication requests* and *preventing N2G interference*. Only the InterPred agent was required for this validation.

Comparatives. The following two methods were used to compare InterPred's performance:

- **Random:** In a state, the agent randomly chooses a timeslot and channel with a uniform probability for G2G communication.
- **Next-Used:** In a state, the agent chooses the next timeslot on a channel where there was no message in the last timeslot of that channel for G2G communication.

The next-used policy requires little storage (the number of messages sent on each channel in the last timeslot). These comparatives were chosen to demonstrate how

trivial solutions perform under real-world wireless communication conditions.

Scenarios and Metrics. All agents are tested in virtual wireless scenarios defined by three simulated datasets (i.e., low, medium, and high load) and the real-world dataset, LoED. Load is defined as the proportion of nodes that require acknowledgements for all messages. Traffic amounts of the medium and high load scenarios are 1.5 and 2.5 times as that of the low load scenario and that of real-world dataset are lower than the low load scenario.

In all scenarios the system parameters are set to $P = 4$, $F = 8$, and $C = 3$. P is chosen based on the available memory and the convergence time of a network. Increasing P increases the convergence time that reduces the accuracy of the system. $F = 8$ is chosen to give enough time to facilitate G2G communications because LoRaWAN nodes open their receive windows after a fixed period LoRaWAN specifications require all nodes to operate on at least 3 channels, so $C = 3$ is chosen. For InterPred parameters, $\alpha = 0.8$, $\gamma = 0.1$, and $\epsilon = 0.2$ are chosen based on empirical studies of all datasets.

The system operates for 24 hours under each scenario. We collected the *numbers of different actions* taken by each agent to quantify its ability to fulfil G2G communication requests, and the *total reward* received by each agent to quantify its ability to prevent N2G interference.

Memory requirements InterPred has a low memory footprint so that it can run on low-resource gateways. The memory requirements for InterPred depends upon P, F, C and the maximum number of messages per slot. With the chosen values, the maximum number of bits required to encode the counter value of 5 is 3 and with $P = 4$ and $C = 3$, only $P * C * 3 = 36$ bits are needed (rounded to $5bytes$) to encode a single state. Total number of states is $((5^P) * C) = 1875$. Assuming every action is a 32-float value, the memory needed to encode all actions of a state is $((F + 1) * C) * 4 = 108bytes$. The total memory needed is $1875 * (108 + 5) = 210Kbytes$. The memory requirement can be reduced by using a 16-bit float or reducing number of future states.

Results. Experimental results are illustrated in Figure 4.9. In scenarios defined by simulated datasets, according to Figure 4.9(a), the InterPred agent fulfils between 92% and 97% of all G2G communication requests. The ratio of bad actions to all actions is between 7% and 13%. The random agent fulfils 88% of all requests in all scenarios, but its bad action ratio increases from 16% to 30% as the network load increases. The fulfilling ratio of the next-used agent is close to 99% all the time, but its bad action ratio

is between 19% to 39%. In Figure 4.9(b) the total rewards for InterPred and next-used agents are more than 1.9 times of that of the random agent in the low load scenario. However, as the network load increases, the total reward for either the random or the next-used agent goes below 0, implying that they have on average a negative impact on the network. On the other hand, the total reward for the InterPred agent always remains positive.

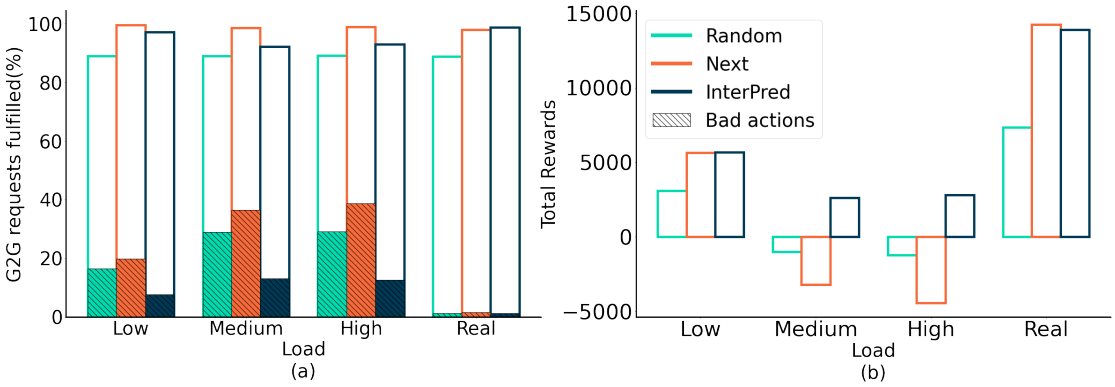


Figure 4.9: InterPred validation results

In the real-world dataset scenario the InterPred and next-used agents perform similarly since the system load is lower than that of the low load simulated scenario. The real-world dataset results are an aggregate of gateways with varying network usage (as shown in Section 4.2.1). This could have a bias on the results as a few gateways with low bad actions can offset some gateways with a large number of bad actions. More analysis could be performed by looking at the results of different gateways individually. The random agent performs much worse than the other two in terms of the total reward. Its bad action ratio, however, is similar to the others. This shows that the random and next-used agents may work well in low load scenarios, but the InterPred agent performs better than both regardless of the load on the system.

It is clear from the results above that IRONWAN has more communication opportunities and causes less interference when using InterPred over the other policies. InterPred uses a SARSA continuous learning model to train a gateway agent to learn a model of its local communication traffic. A single model would not work on all gateways due to the multitude of wireless interference and placement issues. InterPred also enables a gateway agent to deal with changes in the environment like new nodes, dynamic inter-arrival times or environmental conditions.

4.6 Implementation and Evaluation

This section describes the evaluation of IRONWAN. Four performance studies of IRONWAN are performed. The first one is on a small-scale testbed to show that IRONWAN works on gateways without making changes to LoRaWAN. The experiments were trying to show that a handover scheme (i.e. two gateways can communicate) is possible with LoRa gateways. The next two studies are simulations to evaluate how IRONWAN compares against LoRaWAN in highly-dense environments. The final simulation study is to compare IRONWAN against state-of-art protocols, a wired centralised solution (WCS) like PacketBroker[143], and FLIP[88].

Evaluation Criteria. Three evaluation criteria are used:

- **Unique messages/node**- The number of unique messages received at the server per node. Each node sends approximately 480 messages per experiment (depending on start time). Each message is identified by a counter number $0, \cdots, 479$. A unique message has a unique counter number, and is received only once by the gateway. This is an application level metric and shows how much unique information a server receives from a node. In terms of reliable CPSs, this metric shows the amount of unique information available at the *cyber layer*.

- **Packet Delivery Ratio (PDR in %)** - Packet delivery ratio is defined as number of messages acknowledged over the total number of unique messages sent by the node. This metric captures how many messages that needed acknowledgements were successfully acknowledged. In reliable CPSs, this would increase the reliability of messages being delivered to the *cyber layer*.

- **Number of retransmissions (NoReTx)** - The average number of retransmissions needed for messages that require acknowledgements to be acknowledged. Reduction in NoReTx implies that nodes have to retransmit less and conserve their energy.

The above three metrics are derived from the requirements of our multi-stakeholder application which is to ensure all nodes can deliver messages while conserving energy.

Evaluation algorithms. IRONWAN is first compared with a baseline algorithm, LoRaWAN:

- **LoRaWAN:** This is baseline LoRaWAN that represents a typical use-case of LPWANs. Nodes implement Class A specification (Section 2.4.1) and use ADR. There

is no G2G communication in LoRaWAN.

- **IRONWAN:** This inherits all properties of LoRaWAN and implements the G2G communication.

4.6.1 Testbed Evaluation

Setup. IRONWAN is evaluated with two overlapping LoRaWAN networks, each with 5 nodes and 1 gateway connected to a server. Collisions were forced by having all of the nodes transmit at the same time, on the same frequency, using spreading factor 7. The duty-cycle on one of the gateways was reduced to 0%. The nodes transmitted a message every 20 seconds and the experiments were run for 5 minutes and repeated 10 times. The average and standard deviations are reported.

We use Adafruit Feather M0 RFM95 as LoRa nodes communicating over USB to a host running Linux Ubuntu 18.04. All of the hosts were synchronised with NTP, and they instructed the LoRaWAN nodes to transmit at the same time to force message collisions. Two MultiConnect Conduit Gateway were used as LoRaWAN gateways each with an 868MHz +3dbi whip antenna. Each gateway was connected to its own network server.

Results. Figure 4.10 shows the results of testbed experiment. The results show that the average unique messages per node increases by 2-6% and the average PDR increases by 15-88%. The results indicate that IRONWAN works on gateways and can improve the total messages received and PDR.

4.6.2 Large-scale Simulation Evaluation

Next, large-scale simulations are performed to study how IRONWAN works in dense urban scenarios.

Setup. IRONWAN is evaluated on the FLORA[144] LoRaWAN simulator using OMNET++ and the INET framework. 1000 nodes are uniformly deployed in a $4km^2$ simulation area. Gateway density is chosen based on an experimental study where an average of $2-3$ gateways should receive a message for any node with a gateway density of one every $1.5km^2$ [145]. For e.g. one of the densest LoRaWAN gateways in thethingsnetwork (largest public LoRaWAN network) is in Zurich. There are 218

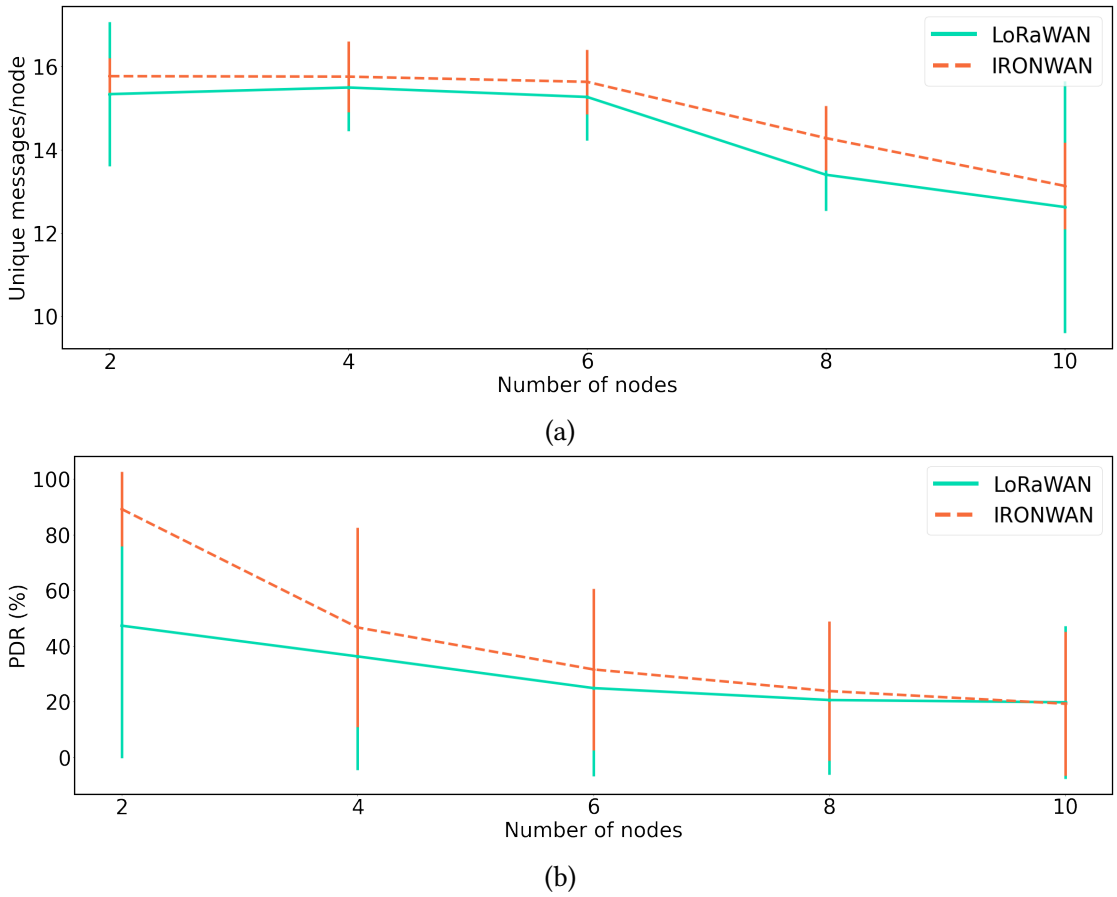


Figure 4.10: Testbed evaluation

gateways deployed in Zurich (area $90km^2$), that's an average of $2.4gateways/km^2$. These numbers are similar to the scenario where 10 gateways are deployed in a $4km^2$ deployment. Six, eight, and ten gateways are simulated, $G \in 6, 8, 10$, to give an average of $1.5, 2, 2.5$ gateways per network. The simulated networks had G gateways connected to 4 independent networks with their own servers. The nodes transmitted a new message every three minutes (maximum of 480 messages). The experiment is run for 24 hours of simulated time. The network load is defined as the percentage of nodes that require acknowledgements for all of their messages. Three network loads are evaluated: low, medium and high that correspond to 10,50 and 90% of nodes requiring acknowledgements. The low load scenario is representative of the current LoRaWAN deployments and as more CPS applications start using LoRa, the network load would increase to 50% and 90%. The nodes choose a spreading factor to transmit on from a uniform random distribution, they have a 14dBm transmit power, 1% duty-cycle and

choose one of three channels to transmit on. The nodes retransmit a message for upto 8 times before dropping it.

4.6.2.1 Study 1: Impact of increasing gateways

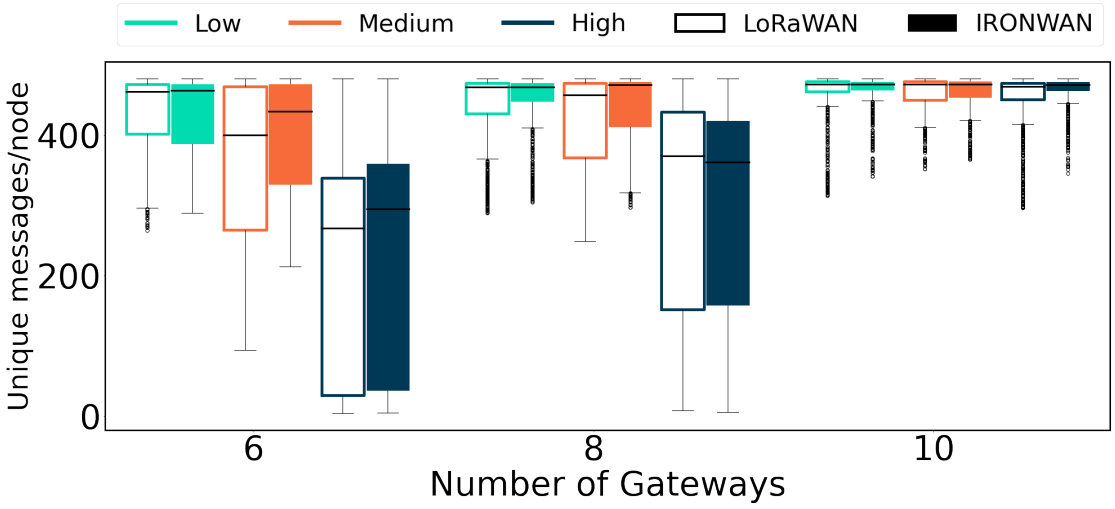
The first study compares LoRaWAN and IRONWAN while increasing the number of gateways and the network load.

Unique Messages Received and PDR. Figure 4.11a shows that the total messages received increases with an increase in gateways for both LoRaWAN and IRONWAN. IRONWAN receives 12% more messages than LoRaWAN for a medium load scenario with 6 gateways. For other scenarios, the average total messages received by IRONWAN is from 1-7% better than that for LoRaWAN. The low throughput gain can be attributed to the LoRaWAN Adaptive datarate which reduces the transmission power to reduce the number of nodes heard at multiple gateways. IRONWAN increases the minimum number of messages received from a node in all but the high load with 8 gateways and reduces the 25th percentile (lower line of the box) for all scenarios. IRONWAN reduces the starvation of nodes and the servers have more information from every node.

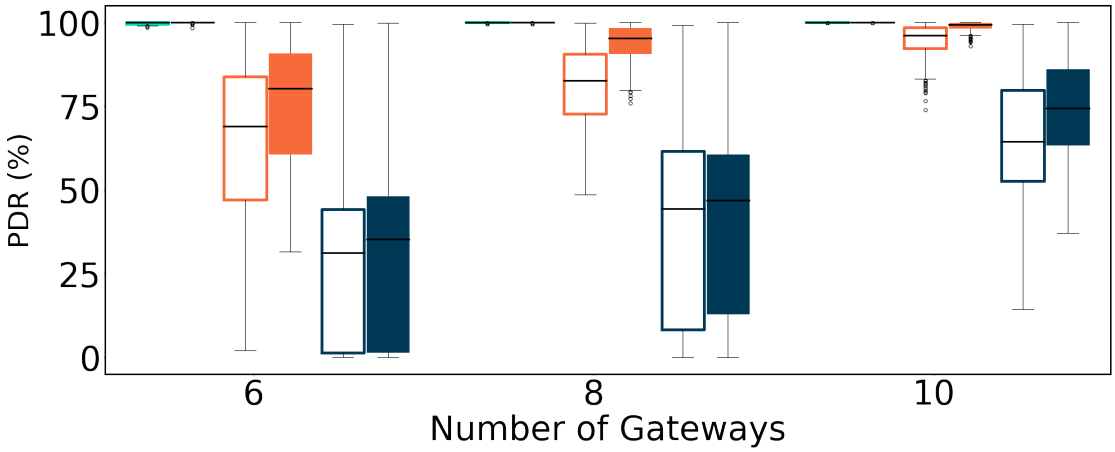
Figure 4.11b shows that the average PDR in low-load scenarios is above 99.7% and 99.9% for LoRaWAN and IRONWAN respectively. With a medium-load IRONWAN has 11%,12% and 5% higher PDR for 6,8 and 10 gateways. A similar but smaller PDR performance increase is seen for IRONWAN when the network has a high load. The minimum PDR does not change in low-load scenarios. For medium load IRONWAN has a minimum PDR that is better than the LoRaWAN's minimum PDR by 28% for 6 gateways, 27% for 8 gateways and 20% for 10 gateways. IRONWAN's minimum PDR for a high network load with 10 gateways is 23% higher than that of LoRaWAN. The results show that networks with a medium or high load have a higher PDR with IRONWAN than with LoRaWAN. The results also show that for networks with a high load and 10 gateways IRONWAN enables gateways to handle more acknowledgements and improve its minimum and average PDR.

Figure 4.11a and Figure 4.11b show that as the load on the network increases, IRONWAN increases the reliability by handling more acknowledgements which reduces the load on the network. Next, the impact of IRONWAN on NoReTx is studied.

Number of retransmissions IRONWAN and LoRaWAN have very similar messages



(a)



(b)

Figure 4.11: Evaluation of LoRaWAN and IRONWAN with increasing load and number of gateways

received per node with high PDRs is as seen in Figure 4.11a and Figure 4.11b. There is a difference in the number of retransmissions required to achieve the messages received. Figure 4.12 shows the NoReTx per node. IRONWAN reduces the NoReTx in low load scenarios by 1%,21% and 32%, for medium load by 8%,29%,39%, and for high load by 2%,5% and 6% when compared to LoRaWAN. This shows that IRONWAN can achieve better performance to LoRaWAN while consuming lesser energy in low and medium load scenarios.

Overhead of IRONWAN Figure 4.13 shows that the additional communication overhead caused by the G2G communication used in IRONWAN when compared to Lo-

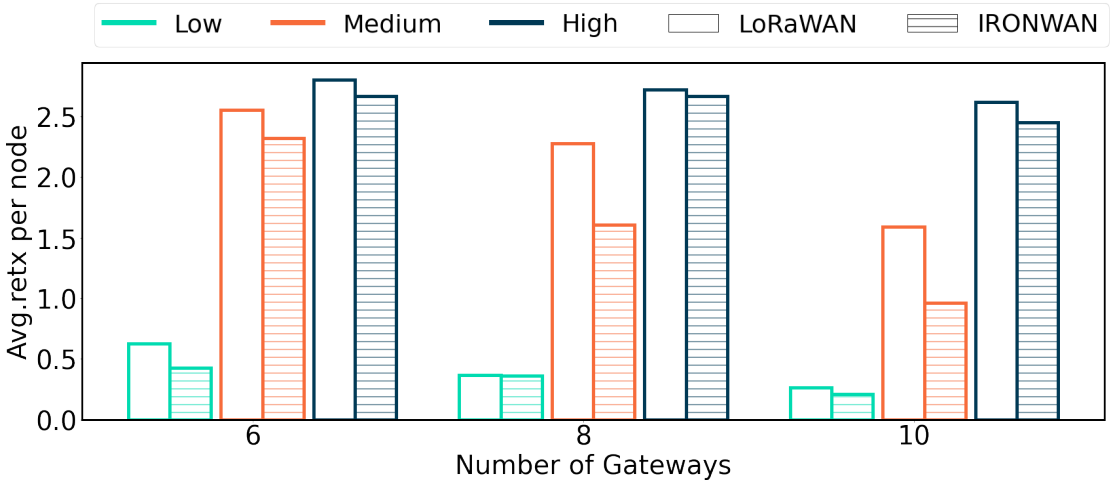


Figure 4.12: Average number of retransmissions

RaWAN. IRONWAN's overhead is the number of messages transmitted on bands 0 and 1 and LoRaWAN's overhead is only messages transmitted on band 1. IRONWAN's overhead is 12-14% for low load scenarios, 0-7% in medium load, and 2-5% in high load scenarios. An interesting observation is that when IRONWAN is used in a network with a medium load and 8 gateways it has its highest gain in PDR (Figure 4.11b) and transmits less messages per node. This occurs because IRONWAN provides a better redistribution of resources and reduces acknowledgements. The results show that overhead of IRONWAN is not high and that it uses spare gateway duty-cycles to the benefit of all networks.

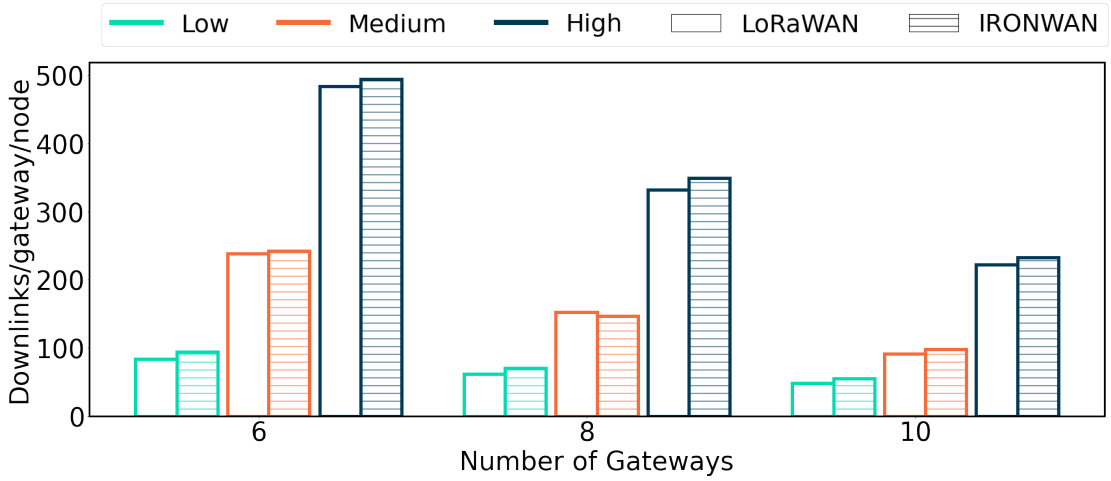


Figure 4.13: Overhead for IRONWAN compared to LoRaWAN

4.6.2.2 Study 2: Impact of increasing number of retransmissions

The second study tests the effect of varying the maximum number of retransmissions on LoRaWAN and IRONWAN. LoRaWAN allows users to choose a policy that limits the maximum number of retransmissions($retx$). LoRaWAN specifications recommend a retransmission limit of 8 which is what we use in all other experiments. Capuzzo, Magrin, and Zanella[84] show that increasing the number of retransmissions increases the probability of lost packets. This experiment studies the effects of varying the retransmission limit on the performance of IRONWAN.

Results. Figure 4.14 shows the unique messages received and PDR for a maximum of 2,4,6 and 8 retransmissions in a network with 10 gateways distributed between 4-servers. The total messages received is similar for all networks loads. This allows us to see how the PDR changes to achieve the same performance. A clear trend emerges where the PDR reduces to 78% for a medium load and 50% for a high load for 2 retransmissions for LoRaWAN. For IRONWAN under the same conditions the PDR only drops to 90% and 63%. Another observation is that IRONWAN with $retx$ retransmissions has a 5% to 10% higher PDR than LoRaWAN with $retx + 2$ retransmissions. Instead of increasing $retx$, IRONWAN could be used instead which would reduce the load on the network and the number of messages sent by the nodes by replacing node retransmissions with G2G messages requests. This can also be seen in Figure 4.14b. In all scenarios, the average PDR of the system increases by up to 20% and the minimum PDR increases in range of 25-160% for medium and high load scenarios. Increasing the retransmissions increases the load on the network and does not significantly increase the PDR. This is evident from the high-load scenario where the average PDR is always higher in IRONWAN compared to cases in which the retransmission limit is increased. This study shows that IRONWAN increases the total messages received and PDR of the system when compared to LoRaWAN with an increasing retransmission limit.

4.6.2.3 Study 3: Comparison to state-of-art

The final study compares IRONWAN with two systems. As both of the algorithms under comparison have components that tackle different issues, the focus of the study is only on the parts of solution that are comparable to IRONWAN. The two solutions are:

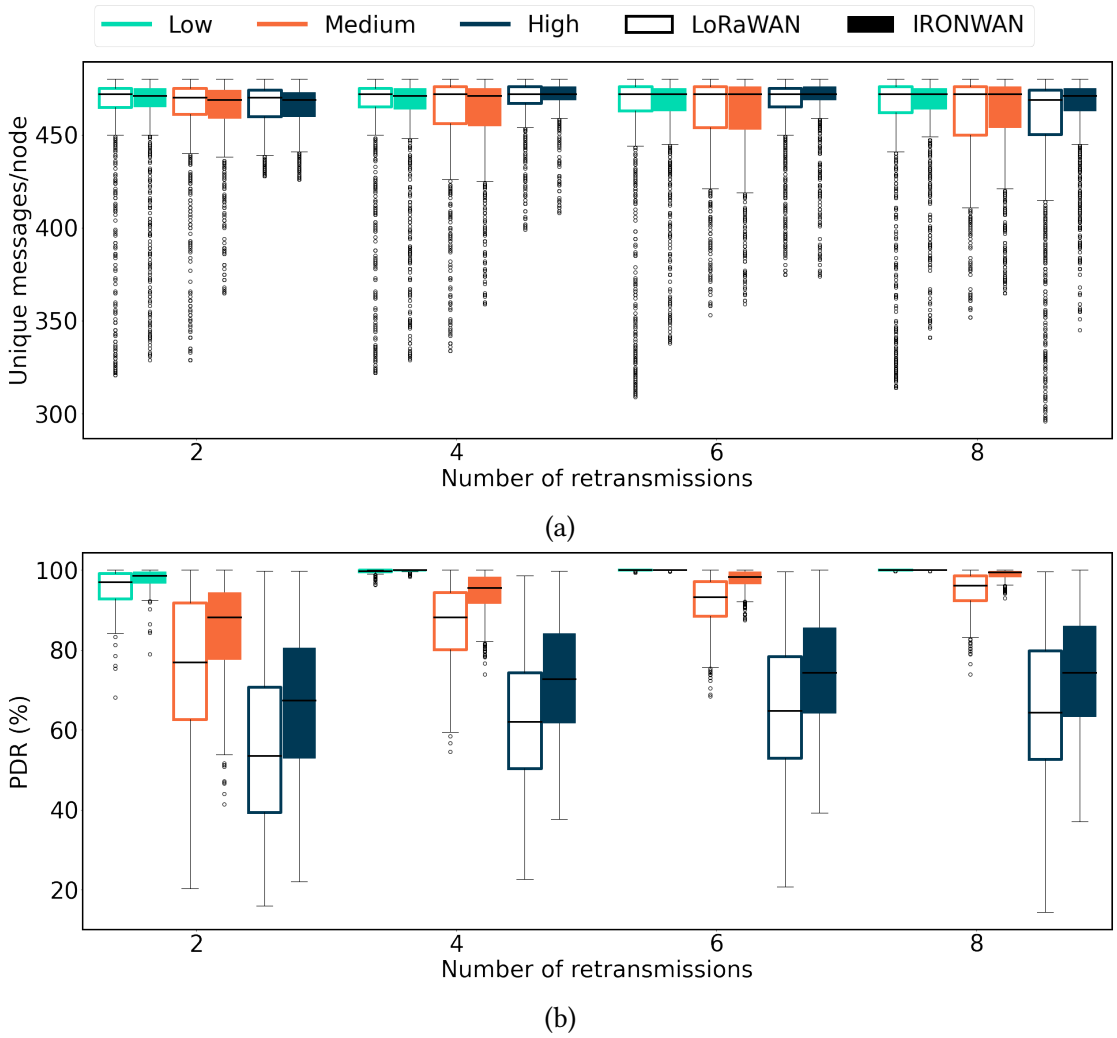


Figure 4.14: Varying maximum number of retransmissions

1. Wired centralized solution (WCS). This is a centralised system that facilitates message exchange between multiple servers. To replicate this in our system, every time a server receives a message, it forwards messages not destined for it to all other servers. So, the servers have information from all gateways that received their message, and they can also use those gateways for acknowledgements. However, they only choose to use those gateways if none of their gateways had received a message. The overhead of this system is calculated as the number of times it used a message received from another server or used another server for acknowledgements. This is the most significant difference between a network with 1-stakeholder and ideal centralised wired solution.

2. FLIP. FLIP is a peer-to-peer network between gateways to distribute the load of the network. FLIP does this by minimising Shannon's entropy and coming to a consensus about what nodes are handled by which gateways. For our comparison, we implement a centralised server that receives all of the messages and assigns nodes to gateways such that the entropy is minimised. We use this technique as it mimics the operation of FLIP which minimises entropy by sharing the load.

Results. Figure 4.15a and Figure 4.15b show the results of this study. Only results for medium and high load are shown as all three algorithms perform similarly under the low-load scenario. Ideal centralised wired system always has the highest number of unique messages received and PDR as it provides the behaviour of an unpartitioned network. The cost required by this system to achieve this can be seen in Figure 4.15c. For a medium load scenario, the total messages received by IRONWAN is as good as FLIP with 6 gateways and outperforms it when 8 and 10 gateways are used. The total messages received by FLIP is higher than that for IRONWAN in the 6 and 8 gateway scenarios. IRONWAN receives more messages than FLIP in the 10 gateway scenario. The PDR for medium load follows a similar trend, where IRONWAN has a higher PDR and much higher minimum PDR when compared to FLIP and is comparable to WCS in the 10 gateway scenario. For a high load scenario, IRONWAN has a lower PDR when compared to FLIP and WCS, however, it increases the maximum PDR from 80% in FLIP and to 100% in WCS. IRONWAN's PDR is higher than FLIP in 10 gateway scenario and closer to WCS's. Figure 4.15c shows that the cost of IRONWAN is significantly lower than WCS in all scenarios where cost for IRONWAN is number of messages it requested or responded for other networks.

The results of our evaluation show that IRONWAN increases total messages received by a limited amount while increasing the PDR in most scenarios. It also increases the minimum PDR in all scenarios. IRONWAN also reduces the number of retransmissions in all scenarios with minimal overhead per gateway. IRONWAN does not perform as well as an ideal centralised wired system (which is similar to an unpartitioned network) but performs better than FLIP in terms of PDR. Finally, the results show that IRONWAN is a candidate solution for dealing with multi-owner overlapping networks commonly found in LoRaWAN.

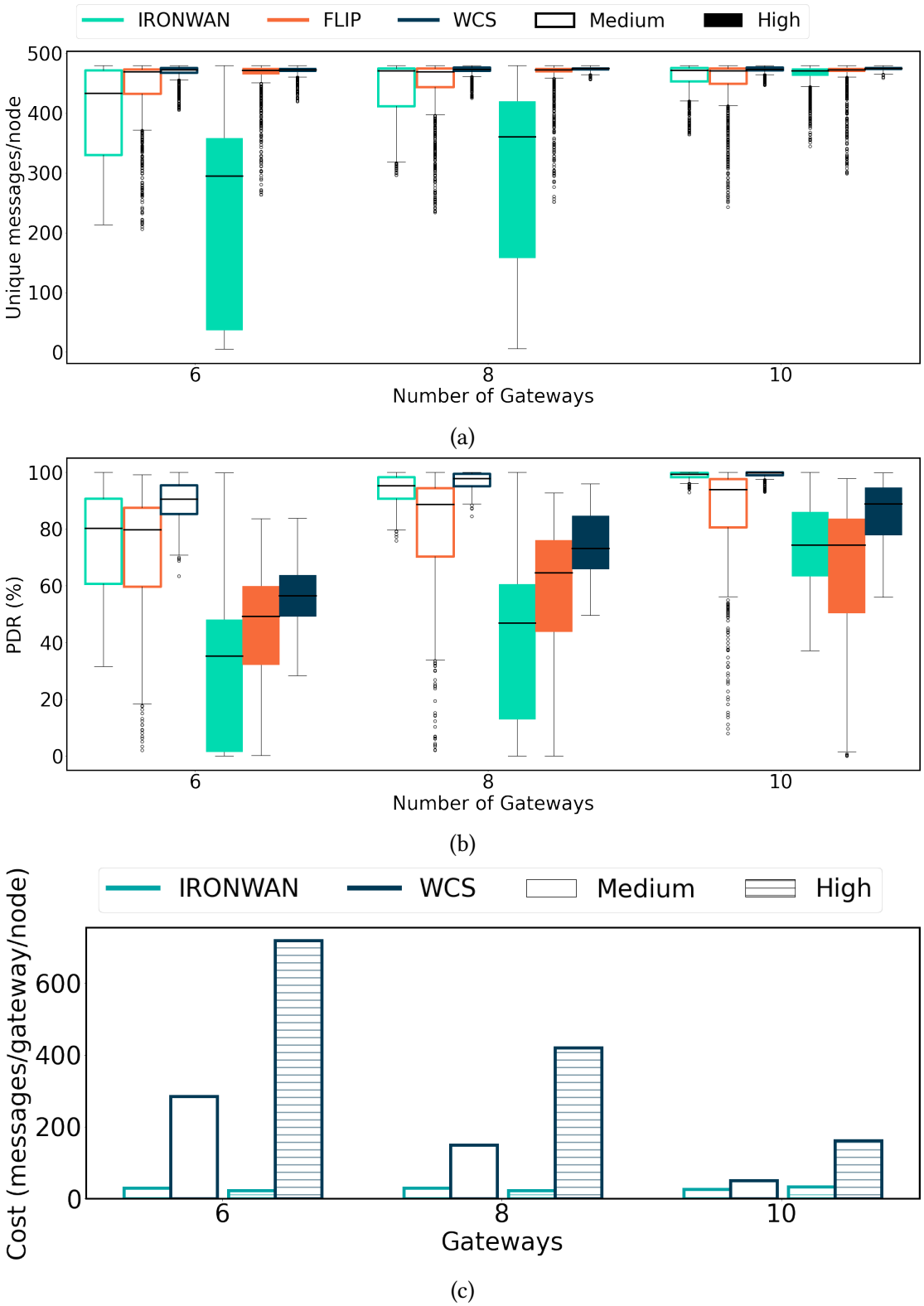


Figure 4.15: Comparison of IRONWAN, FLIP and ideal centralised system

4.7 Discussion, Limitations and Future Work

IRONWAN currently has no idea about how many gateways are in its communication range or whether they can be trusted, or are malicious. This leads to a few problems in its functioning. IRONWAN gateways running in an environment could endlessly keep pining other gateways and not get a reply leading to a wastage of communication bandwidth. A neighbourhood discovery protocol could help gateways form a map of its neighbouring gateways. An important assumption for IRONWAN's functioning is trust in the neigbouring gateways. Although there are no security issues from an encryption standpoint, replay attacks, denial-of-service and other network attacks are still feasible. These problems are inherent in LoRaWAN protocol, and some of these attacks are already accounted for in the LoRaWAN specifications by rejecting old frame counters, using channel hopping and implementing randomness in transmission times. A map of neighbouring gateways could be extended to create trust between gateways to deal with free-loading or malicious gateways. Free-loading gateways are gateways that exhaust duty-cycle of other gateways without offering their services and malicious gateways could request fake handover requests to exhaust a gateway's duty-cycle. Currently, the gateways also have no incentive to exchange messages. Their only benefit is a higher node throughput at lower-energy consumption. For the system to be commercially employed, some sort of monetization schemes can be explored. The G2G messages are also not encrypted nor can they be authenticated currently. The neighbourhood discovery can be used to create keys to ensure some form of authenticity.

RMIP has been designed for periodic data based on the LoED dataset. Other patterns that we haven't found in our dataset could exist. Traffic patterns from event-based systems also cannot be dealt with RMIP. An extension of RMIP for event-based traffic could be designed.

We have tested InterPred for a limited time and it is unknown how it would perform when run for a long-time. A real-time parameter tuning maybe needed to deal with drastic changes in the wireless environment or a periodic reset. This sort of information can only be gauged after more data or a real-world deployment.

We also need to study the closed-loop effects of the system. As we are interacting with the environment (by transmitting messages) while learning from it, we are changing the transmission patterns of the system. This could lead to nodes retransmitting

a message at a later time which would lead to an incorrect prediction of the channel usage.

In this chapter, we were interested in studying how multiple overlapping networks can be leveraged to increase network reliability. We wanted to study this reliability increase on LoRaWAN. If LoRaWAN (assuming class A only) was used directly for CPSs, sensors and actuators both would need to send and query information from the server, where IRONWAN's current approach would directly work. If we assumed actuators were Class C devices (assumed in Chapter 5), they would be constantly listening, in which case, we can leverage InterPred to find the best slot to send data to actuators, which would have a higher chance of getting successfully delivered.

4.8 Related Work

In this section we review current research that tackles aspects of the reinforcement learning, multiple gateways, overlapping networks in LoRaWAN.

Benefits of multiple gateways: Di Vincenzo, Heusse, and Tourancheau [135] show that increasing LoRaWAN gateways from one to four improved the messages that were acknowledged from 24% to 40%. The authors of [85, 75] used multiple gateways to directly forward all of signals received from all of the nodes in their range to a central server where they are combined and decoded. This is an orthogonal approach to ours as it changes the role that gateways play in a LoRaWAN network by offloading decoding of packets to a server which requires precise time-synchronisation, high bandwidth wired links and most importantly all gateways talking to the same server. To the best of our knowledge, all previous work has assumed that multi-gateway LoRaWAN networks share the same server which is contradictory to LoRaWAN deployment methods where multiple overlapping networks exist. Our work removes this assumption and allows networks to retain their autonomy while still working together to improve performance for all.

Merging overlapping networks: FLIP[88] federates the gateways. The federated gateways create a distribution of the nodes in overlapping communication regions at initialisation. The gateway of a neighbour network will not forward the message of an unallocated node if the network changes to make the neighbour gateway favourable. The federated gateways share decryption keys for the allocated nodes which poses a

threat in the case of malicious gateways. FLIP assumes a wired internet connection for gateway to gateway communication which may not be available, reliable, or secure. In IRONWAN gateways do not require the keys of neighbour nodes. IRONWAN uses the wireless spectrum overcoming the problem of an unreliable wired connection[146]. As shown in Section 4.6, IRONWAN performs better than FLIP in medium load scenarios and high load scenario with ten gateways without the need to exchange encryption keys.

Reinforcement learning in wireless networks: Reinforcement learning has found several applications in wireless networks. Wang et al.[147] use deep reinforcement learning to model dynamic multichannel access. They use this model to then choose which channel to transmit on. This approach is analogus to ours with one major difference that we have chosen to use SARSA, which allows us to keep the algorithm lightweight to run on embedded gateways with low memory footprint. Sandoval, Garcia-Sanchez, and Garcia-Haro[148] use reinforcement learning to replace ADR to allocate transmission paramters for every node. Their results show an improvement of 147% in per-node throughput.

4.9 Summary

This chapter targeted the problem of multiple CPS applications running in an urban area leading to performance degradation of all. We proposed IRONWAN and its novel components RMIP and InterPred. IRONWAN leverages overlapping LoRaWAN networks to enable wireless gateway-to-gateway communication which reduces the negative effects of node-to-gateway message collisions and efficiently shares gateway-to-node communication capacity. RMIP uses statistical learning based algorithms to predict the inter-arrival times between messages, and to trigger a hunt for failed messages. InterPred ensures that any gateway-to-gateway communication has no negative effect on the underlying networks.

The results show that IRONWAN works well in low-load and medium-load scenarios (can be considered typical use-cases of LoRaWAN and CPSs). It meets the requirements for multi-stakeholder applications by increasing the messages received per node and the packet delivery ratio. It reduces the starvation of nodes in a partitioned network, where they were unable to get even a single acknowledgement. While achieving

performance gains, it reduces the number of retransmissions, in-turn reducing the energy consumption of nodes. IRONWAN works without making any changes to LoRaWAN.

The results show that IRONWAN requires a high number of gateways to achieve a good level of performance in the high-load scenario. This implies new systems are needed to handle high-load LoRa scenarios. In the next chapter, we will look at dealing with applications that have a sampling interval of a few seconds instead of minutes that targets a closed-loop feedback system, considered as a high-load scenario.

5

Control Communication Co-Design for slow-loop Critical Infrastructures

In the previous two chapters, we discussed CPSs that required high throughput, high reliability and low latencies but had no hard-real time requirements. This chapter focuses on WDN that require high-reliability and bounded delays to ensure system stability. WDNs are a system consisting of sensors, actuators, pump, motors, pipes and tanks deployed over a city-wide area. Despite focus on WDNs, the work in this chapter can be used for any slow-loop wide-area infrastructures that have stricter requirements.

Augmenting WDNs with wireless communication allows for demand management and fault detection that would save water and money. LoRa's long-range capabilities are highly suited to cover the city-wide area and inaccessible deployment locations of these WDNs. However, LoRaWAN's channel access scheme can not guarantee successful transmission or bounded delays as it uses a ALOHA channel access scheme.

Opportunity. The opportunity WDNs is to supplement them with wireless communication. However, a off-the-shelf wireless system cannot be used as the application has stricter requirements. There is a tighter coupling between the physical and cyber layers compared to the previous two chapters. A wrong actuation command could lead to a sudden change in pressure leading to pipe bursts. The first step for LoRaWAN systems being deployed in WDNs is already underway[1]. However, these are sensor-only systems with no feedback loop, which is needed to guarantee stability by changing

This chapter is based on [41] and [44]. I am the main author of the paper. I contributed to the main idea and writing. I implemented and designed the wireless protocol and simulation tools and performed the evaluation. The control model, its analysis were the contributions were of a co-author.

[1] https://www.semtech.com/uploads/technology/LoRa/app-briefs/Semtech_HB_WaterManagement_AppBrief-FINAL.pdf

valve positions with varying demands or network faults. A co-design of the control and communication system is needed to meet application requirements and ensure system stability.

5.1 Problem Introduction

Problem. The previous two chapters discussed CPSs that required reliability but did not require hard-real time guarantees. This chapter deals with wide-area critical infrastructures that require high-reliability and bounded latencies to ensure system stability. The applications dealt with mutli-gateway scenarios where sender and receiver diversities were exploited to ensure a high throughput with low latencies and energy consumption. However, as the expectations of the network increases in terms of bounded latencies, relying only on the communication protocol or an ALOHA channel access scheme is no longer a solution.

The challenge of engineering WDNs is the tight coupling between the requirements of the *physical system's* controller, sensor and actuator and the communication between them[18]. Moving the controller to the *cyber layer* instead of it being close to the sensors allows the system to use information from multiple sensors to make appropriate decisions for the actuators anywhere in the system. This capability can also enable fault detection and reduce wastage of resources like water or electricity. For designing applications with high-reliability and bounded latencies, abstract notions of delay cannot guarantee closed-loop stability. A better understanding of reliability and latencies is needed to choose the controller's and data-link layer's parameters.

The WDN model used in this chapter is shown in Figure 5.1. It consists of several District Meter Areas (DMAs). Each DMA has three or four water tanks that supply on average 10-30 customer connections via a set of pressurised pipes, pumping stations, and valves. Their dynamics are complex, non-linear, difficult to model and analyse. If the complexity of the system is reduced to a problem of stabilising the tank water level to a pre-defined set point, the non-linear model can be linearised as in [149]. It is difficult to model the complex interactions of a WDN. The network consists of 100s of sensors and actuators. These networks have a sampling interval of a couple of seconds due to the slow-dynamics of the actuators[150]. The messages need to be delivered within a maximum delay or it could lead to an unstable system. The sensors

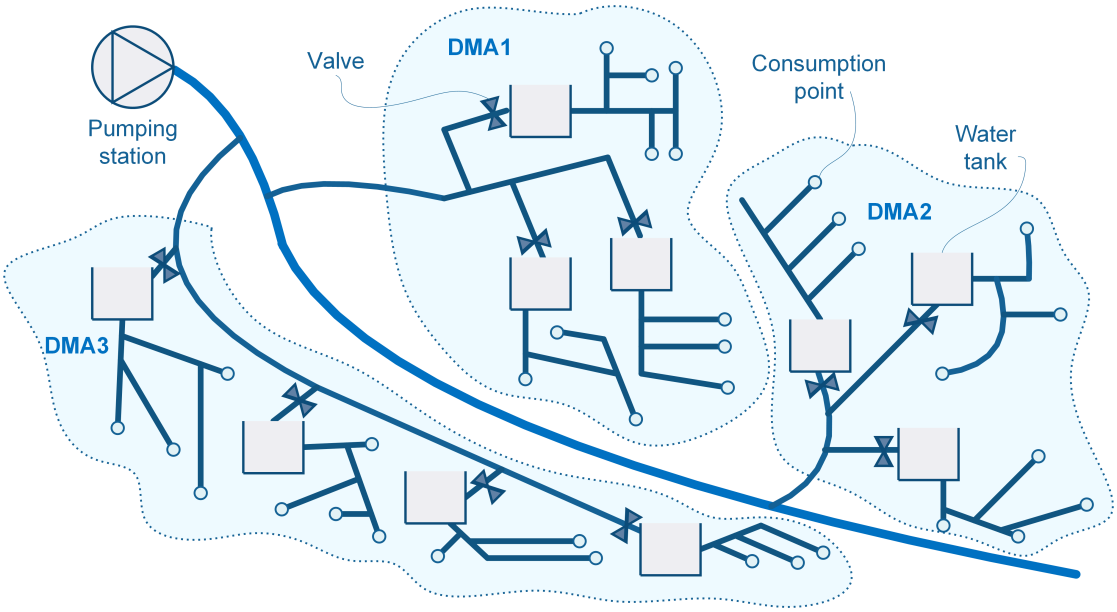


Figure 5.1: The water distribution network.

can also transmit more data if an unstable condition is detected. Finally, the sensors are deployed in areas where continuous supply of energy can be difficult and so energy must be conserved.

Solution. To meet the requirements of a WDN, we use a *Control Communication Co-design approach* (referred to as the C^3 approach). C^3 is a co-design approach for designing control and communication systems. The approach's core is that the control and communication parameters are defined, and the system constraints are specifically designed for the application. At design time, the C^3 approach accounts for the effect of control and communication parameters on each other. The parameters are chosen to ensure that the system is globally exponentially stable when used at run-time and does not require further updating. The outcome of the C^3 approach is *Ctrl-MAC*, a new data-link protocol and its associated event-based controller. *Ctrl-MAC* addresses the imperfections of single-hop LoRa networks that are not designed for control purposes in terms of delay bounds, message loss, two-way traffic and duty cycling. The event-based controller reduces the load on the communication network. *Ctrl-MAC* and the associated event-based control model are the first to the best of our knowledge, to demonstrate a provable, workable solution for closed-loop control of a WDN using LoRa in resource-constrained sensor and actuator nodes.

Contributions. The main contributions in this work are:

- A C^3 approach is used to design a Ctrl-MAC and the associated controller. The work practically demonstrates that LoRa can ensure system stability of wide-area WDN.
- *Ctrl-MAC* is a queuing-based data-link layer that can provide a backbone for guaranteed simultaneous control of multiple physical processes that require bidirectional data communication. *Ctrl-MAC* can also handle aperiodic and bursty traffic patterns.
- An event-based control model is designed and formally analysed to guarantee the closed-loop stability of large-scale WDN.
- We built a *cyber-physical co-simulation environment* that connects Matlab/Simulink and OMNeT++. There are no suitable simulation environments that can fully model a LoRa network while also providing an accurate water process model.

5.2 System Overview

This section explains our control communication co-design approach C^3. Firstly, the architecture and a model of WDNs is presented. Then the set of key design parameters from the control system and the communication system are presented that are mapped into the co-design constraints.

5.2.1 System Architecture

Our C^3 approach focuses on a WDN with an architecture shown in Figure 5.2.

Physical System and Controller. The physical system consists of N subsystems each labelled i, where $i \in \{1, \ldots, N\}$. Each subsystem consists of a single plant and controller, and both are labelled with the same i as their subsystem. We describe the ith subsystem's plant using a linear time-invariant model:

$$\dot{\xi}_i(t) = A_i \xi_i(t) + B_i v_i(t). \tag{5.1}$$

The vectors $\xi_i(t) \in \mathrm{R}^{n_i}$ and $v_i(t) \in \mathrm{R}^{m_i}$ are the state and control input vectors

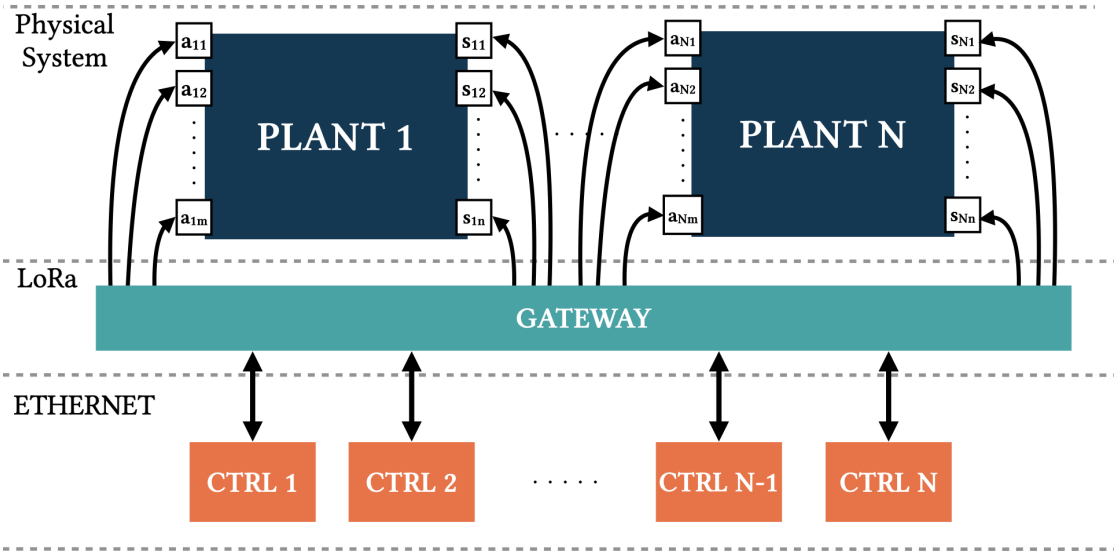


Figure 5.2: System architecture.

of subsystem i, respectively. n_i and m_i correspond to the number of sensors and actuators of the subsystem i, respectively. A_i and B_i are the matrices appropriate to the application.

The plant i is instrumented with sensors $\{s_{i1}, s_{i2}, \cdots, s_{in_i}\}$ to measure its physical processes (individual states of $\xi_i(t)$). The controller receives these measurements and processes them according to an underlying control scheme. The goal of the controller is to maintain the values in vector $\xi_i(t)$ in Equation 5.1 at a certain fixed value, a reference or set-point. We use a linear input-to-state feedback controller given as:

$$v_i(t) = K_i\hat{\xi}_i(t), \tag{5.2}$$

where $\hat{\xi}_i(t) \in \mathrm{R}^{n_i}$ are the measured values of states transmitted to the controller. K_i is a designed control gain for which the closed-loop matrix $A_i + B_iK_i$ is a Hurwitz matrix.

The output of the controller is an action that is sent to actuators $\{a_{i1}, a_{i2}, \cdots, a_{im_i}\}$ to influence the dynamics of the plant. The communication between the plant and the controller forms a single feedback control loop and spans potentially kilometres.

The feedback control loops in WDNs have dynamics where changes occur in tens

of seconds rather than milliseconds. The slow-dynamics are a function of the actuators that can take up to 9 seconds to go from fully closed to a fully open position[150]. We use a two-step process to engineer a controller that can achieve and maintain a stable operation. First, we design the control gain K_i in Equation 5.2 so that the controller has a slow response to system changes. The slow response reduces the overshoot (deviation from the reference or set-point) of the system's state. Second, we choose an event-based control system because it minimises the overall number of data transmissions and saves bandwidth. The definition of event-based control is presented in Section 5.4. The set of parameters that guarantee the stability of the subsystem i (based on the Corollary 1 in Section 5.4.3) are:

- *Sampling interval,* h_i - The time between individual samples. We address systems whose dynamics change slowly. The sampling time supported by the controller is in the order of tens of seconds (from 1 to 50 seconds).
- *Maximum allowable transmission delay,* τ_{di} - The maximum interval between node state measurement and the time the actuator nodes receives updated controller actions. The delay supported by the C^3 approach is in the order of seconds (from 0 to 15 seconds).
- *Event trigger parameter,* σ_i - It defines the size of the error between the estimated and measured states. A small σ_i is more sensitive to change and triggers new control inputs faster, leading to a larger number of events. We chose to use the largest σ_i that can still guarantee stable system operation and minimise energy consumption.
- *Convergence rate of the control system,* ρ_i - The application does not have strict restrictions on convergence speed as long as they can meet the delay tolerances, hence $\rho_i = 0.001$ for all i.

These parameters are core to our C^3 approach as they are shared with and influence the communication system behaviours described next.

This work assumes that sensor nodes and actuator nodes are not collocated. Another assumption is that all sensors are battery-powered embedded devices. Actuators are battery or mains-powered as actuation requires relatively high power. However, the approach is directly applicable to wireless networks where the sensors and the actuators

harvest energy for energy-neutral operation, which is relatively deterministic and appropriate[151].

Wireless Communication System. We focus on LoRa networks because they are designed as low-power one-hop wireless solutions for long-distance communication. LoRa enables the communication between the sensors, the actuators, and the controller over multi-kilometre distances. Single-hop wireless networks are better than multi-hop wireless networks for control-based systems due to their increased reliability and lower (easier to determine) communication delays[152, 28].

All N subsystems in Figure 5.2, and therefore N feedback control loops, are coupled through the shared LoRa communication network using a single gateway. The term gateway and controller are used interchangeably and treated as one device. The gateway and the controllers communicate via wired communication which is assumed to be instantaneous and reliable. In practice, the controller and gateway may reside on different devices, the controller on a server along with other processes (logging and management) and the gateway in a position favourable to receive radio communication (on a rooftop).

It is also assumed that packet loss occurs only as a result of message collisions when two packets arrive simultaneously. Nodes always send up-to-date information and retransmit if no acknowledgement is received. If information is delayed due to retransmission and a new message is generated, the original information is replaced by the most up-to-date information.

The parameters of LoRa that affect the function of the network and the controller are in the list below.

- *Channel access time,* t_{ca} - The maximum time that a sensor node takes to access the channel to send, dependant on the channel access scheme used. The goal of the C^3 approach is to minimise t_{ca} and any other waiting times (delays) due to the time-criticality of information in controlled systems.

- *Time to complete the transmission,* t_t - The maximum time that it will take for a sensor node to transmit its data to the controller (assuming it accessed the channel beforehand) and for the controller to update and transmit its action to an actuator node. We assume reliable communication, so a lost packet triggers a resend. If newer information is available on the sensor node, the old information

is replaced by the newer information.

- *Duty cycle,* DC - The maximum percentage of time spent in communication per node per channel for a given time interval. For instance, a 1% duty cycle allows a sensor node to send for 1 second out of every 100 seconds.

In the next section, we will discuss how the key parameters for the control system and communication system influence each other. The design of the system by the selection of their values is the core of our C^3 approach.

5.2.2 Co-design Constraints

C^3 maps an event-based control scheme to a LoRa data-link protocol, and vice versa, through their coupled parameters. The set of key design parameters from the event-based controller is $\{h_i, \tau_{di}, \sigma_i, \rho_i\}$ and the parameters for the communication are $\{t_{ca}, t_t, DC\}$. The coupling between two sets of parameters is twofold and given via two co-design constraints. These are:

C$_1$ The control parameter τ_{di} is directly affected by the communication parameters: t_{ca}, t_t and DC. The stability can be guaranteed (based on Corollary 1 in Section 5.4.3) only if the service time for each event is less than τ_{di} (0 to 15 seconds) while complying with the DC restriction. We define the service time as the sum of delays due to t_{ca} and t_t.

C$_2$ The choice of control parameters h_i, σ_i, and ρ_i determine the number of events that cause transmissions. These constitute the load on the communication network. The network load cannot be larger than the maximum network capacity. Again, the choice of these parameters is dependant upon **C$_1$**, and stability has to be guaranteed (based on Corollary 1 in Section 5.4.3).

The C^3 approach focuses upon these co-design constraints to enable the design of our system. The following two sections will focus on the two sides of the C^3 approach separately. First, the focus is on the design of the communication protocol, then on the design of the controller. Next, the solution shows how the use of C^3 ensures that the resulting communication and control systems meet the above co-design constraints and guarantee stability.

5.3 A LoRa data-link protocol for Control

In this section, we present a novel MAC protocol called *Ctrl-MAC*. *Ctrl-MAC* is the result of the application of the C^3 approach. . Firstly, the operating principle and the implementation details of *Ctrl-MAC* is presented. Then, a delay analysis is performed to show that *Ctrl-MAC* bounds communication delays and adds guarantees to the low datarates and unreliable communication typical of all of the current, major LPWANs. In doing this, *Ctrl-MAC* ensures that the communication requirement of control systems (the co-design constraint $\mathbf{C_1}$) is met. Finally, *Ctrl-MAC* is evaluated against LoRaWAN. The results confirm that *Ctrl-MAC* achieves lower, bounded communication delays, reliable two-way communication and higher throughput when compared to LoRaWAN.

5.3.1 *Ctrl-MAC* Operating Principle

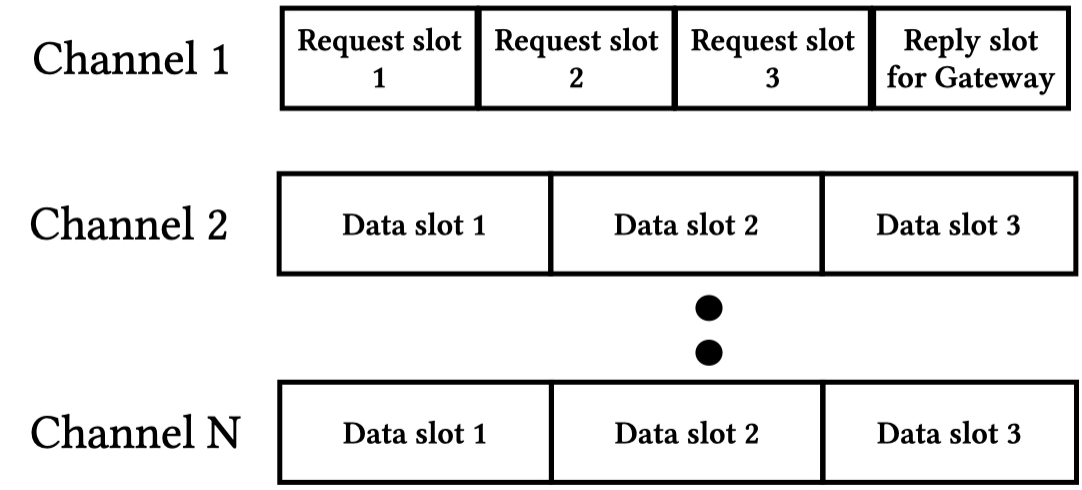
Ctrl-MAC operates in two phases:

Phase 1: Sensing and Data Transmission. Each node (within each subsystem) goes through three stages in the sensing and data transmission phase.

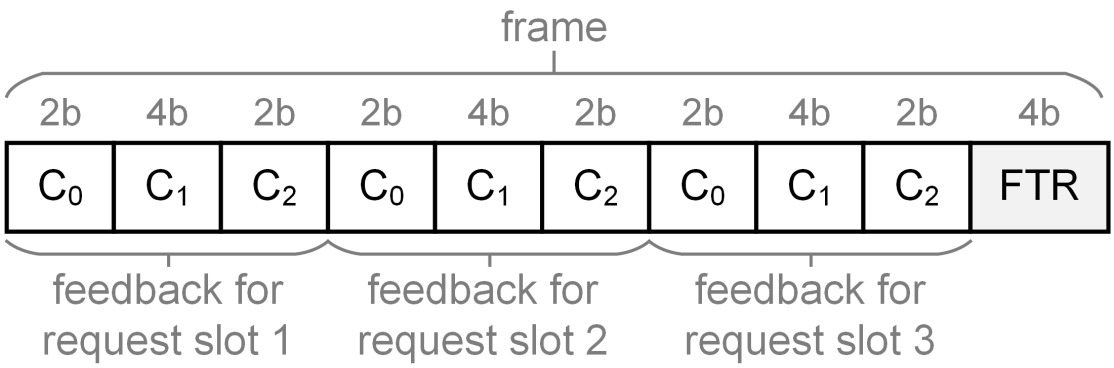
The first stage is when the *node senses the physical phenomenon.* This data is interrogated to identify if a significant event has occurred. It is transmitted if a significant event is identified. The node sleeps until the next sensing period if no significant event is identified.

Upon the identification of an event, the node synchronises to the start of the request period. The node updates its local device clock to the start time of a request-reply message (RRM, shown in Figure 5.3 (b)) received from the gateway. The gateway sends RRMs periodically. Synchronisation is required to correct the clock drift caused after the sensor node resumes from power-saving deep sleep mode. The sensor node also learns the number of request slots k, the duration of each request slot t_{slot} from the RRM.

The second stage is when the *node transmits its data transmission request.* Once the node is synchronised to the request period, it randomly chooses which slot to transmit in (shown in Figure 5.3 (a)). The node chooses using a uniform random distribution of all available request slots instead of a fixed slot or following a round-robin slot scheduling scheme. This is done to manage the case with more nodes than available



(a)



(b)

Figure 5.3: (a) Channel allocation and frame structure in request and data channels (b) The RRM structure for $k = 3$, $M_D = 3$ which implies that $size(C_2)$=$\log_2 M_D$, $l = 16$.

request slots where no fixed schedule is possible. The node makes a data transmission request by transmitting its ID during its chosen time slot. If the node is the only one requesting a transmission slot, its request will be successful; otherwise, there will be a collision (a result of slot contention), and the request for both sensors will fail.

The node is informed about the success or failure of its data transmission request in the next RRM. An example of an RRM for a system with $k = 3$ transmission request slots and $l = 16$ data slots is shown in Figure 5.3. The RRM contains three fields for each transmission request slot. These are:

1. The state of the request slot, $C_0 \in \{0, 1, 2\}$, where $C_0 = 0$ indicates that no node sent a request in this request slot, $C_0 = 1$ indicates that there was no contention in this request slot, and $C_0 = 2$ indicates that a collision occurred in this request slot.

2. The data slot counter, $C_1 \in \{1, \ldots, l\}$, where l is the number of data slots.

3. The data channel counter, $C_2 \in \{1, \ldots, M_D\}$, where $M_D = M - 2$ is the number of data channels, and M is the overall number of channels.

The last value of an RRM is called the FTR. FTR contains the cumulative sum of all contentions that have not been resolved until now. This includes the current failed and previously failed transmission requests. The FTR is decremented by 1 after each RRM.

A node checks the value of C_0 in the request slot that it used to transmit its data transmission request. If $C_0 = 1$, that node can proceed to stage three and send data. If $C_0 = 2$, the data transmission request fails. The node must return to stage two and send another transmission request at the time defined by the value of FTR. If $FTR > 1$, the node counts all of the other request slots with C_0 equal to 2. This value is denoted as r. The node determines the position of its request slot compared to other $r - 1$ request slots. This value is denoted as p. The node waits for the $(FTR + r - p)$th RRM to retransmit its data transmission request choosing the slot k randomly.

In stage three, the *node transmits data*. The RRM contains information to partition data transmissions into $(M - 2)$ channels and l data slots. The gateway allocates channel/slot pairs to requesting sensors on a first-come-first-served basis. When a node receives an RRM with $C_0 = 1$, it changes to the channel in C_2 and transmits in the slot given by C_1. If the node senses new data when requesting a transmission slot, it will send the new data and discard the old. This is because the controller requires the newest information to keep the system stable.

Phase 2: Control Update and Actuation. The second phase of *Ctrl-MAC* is the control and actuation phase. After the controller receives sampled information from the node, it uses the data to calculate the control action to be sent downstream to actuator nodes. This constitutes one of the challenges met by our co-design approach. LoRa prioritises upstream communication only and is therefore ill-suited to meet the timing requirements of control systems with feedback loops. Instead, *Ctrl-MAC*

adds the bidirectional capability to the current LoRa protocol where the downstream communication is as important as the upstream communication.

The gateway sends actuation information to the actuator nodes in the downlink message. These messages are sent periodically, so no DC regulations are violated. The size of each downlink message can change depending on how many actuator nodes need to be updated. The sending time of this message depends on the size of the previous downlink message, so it complies with the DC regulation.

Next, we give the details of the *Ctrl-MAC* implementation and design choices.

5.3.2 Ctrl-MAC Implementation

Ctrl-MAC is implemented on top of LoRa. The reference implementation of Ctrl-MAC has room for significant performance improvement. It purposefully does not exploit any LoRa performance-enhancing features, such as using different spreading factors, or coding rates. Further, in all simulations and experiments *Ctrl-MAC* maintains a constant spreading factor of 7, which gives low transmission time and achieves distances of less than 2km[104] in urban areas.

Ctrl-MAC assumes that there are four channels available. There are three uplink channels of 125kHz with 1% DC and one downlink channel of 250kHz with 10% DC. The current LoRaWAN implementation uses only half (125kHz) of the allocated downlink bandwidth for the acknowledgements. *Ctrl-MAC* exploits this feature and uses the remaining 125kHz to send the actuation signals to actuator nodes. The actuation signals are periodically transmitted while complying with the 10% DC regulation. The downlink message's size depends on the number of actuators and how many bytes of information are used per actuator. For example, for a network with 10 actuator nodes and 2Bytes of information with the 10% DC regulation, the actuation messages can be transmitted every 0.5 seconds.

Our implementation uses only a single channel for data request transmissions (with 10% DC) and the remaining three channels for data transmission. The results in Figure 5.4(a) show that when only a single channel is used for data request transmissions (with 10% DC) and the remaining three channels for data transmission, *Ctrl-MAC* achieves lower delays and higher PDRs than when all four channels are used for both. This happens because the data transmission requests and actual data transmissions

operate concurrently, unlike when the data transmission requests and actual data transmissions happen on the same channel one after the other. The PDR reduces as the number of request slots increase. This happens as the sensors have resolved their contention and are waiting in the data transmission queue for their slot, during which time they drop any old packets. Our implementation allows more requests to be sent, the contention is resolved faster, and data load is distributed evenly over 3 data channels, and therefore more data can be sent.

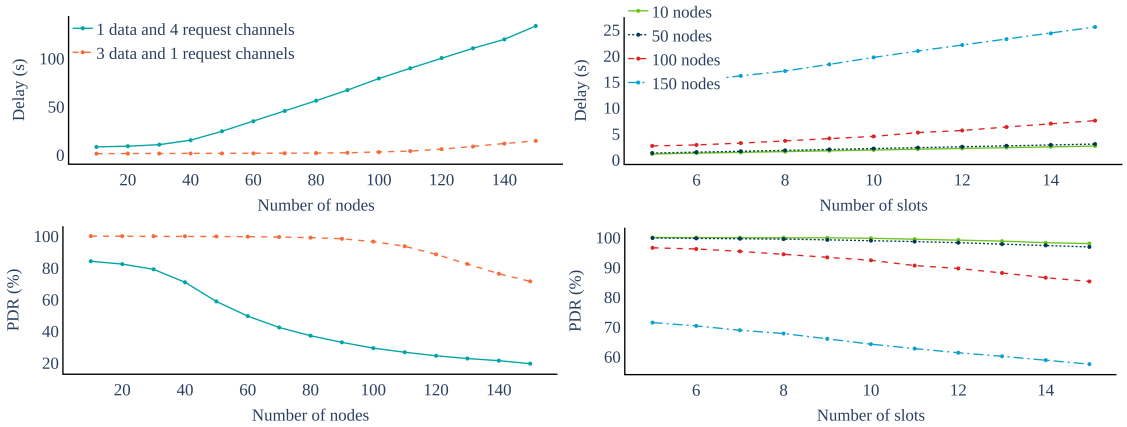


Figure 5.4: *Ctrl-MAC* (a) Channel disciplines comparison, (b) Request slots comparison.

Ctrl-MAC uses $k = 5$ request slots with the duration of $t_{slot} = 0.1$ seconds. $t_{slot} = 0.1$ seconds with $SF = 7$ implies that every sensor can transmit upto a 50 byte packet in one slot. As sensors only transmit header information(13 bytes) and one sensor value (4bytes), there is sufficient time available for sensors to transmit data. The value of k cannot be smaller than 5, and the request slots can not be lower to account for the 10% DC regulation. The results in Figure 5.4(b) show the degradation of the communication delays and PDRs for all k larger than five and all populations of nodes.

Next,*Ctrl-MAC* is modelled to analyse its performance in terms of the guarantees and achieved communication delays.

5.3.3 *Ctrl-MAC* Delay Analysis

This section analyses the communication delays identified by the co-design constraint $\mathbf{C_1}$ from Section 5.2.2, using a queue based approach. The communication delays of *Ctrl-MAC* due to $\{t_{ca}, t_t, DC\}$, are bounded and do not violate the maximum allowable transmission delay derived from the control system. The maximum allowable trans-

mission delay for each subsystem i is $\tau_{di} = \tau_d$, which is from 0 to 15 seconds. The analysis obtains the bounds of $\{t_{ca}, t_t\}$ by deriving the delays of different stages of *Ctrl-MAC*, $\{t_{sync}, t_{req}, t_{send}, t_{update}\}$, where $t_{ca} = t_{sync} + t_{req}$ and $t_t = t_{send} + t_{update}$ while complying with the DC regulation. *Ctrl-MAC* can operate within the communication constraints and keep the total delays less than the maximum allowable transmission delay (i.e. $t_{sync} + t_{req} + t_{send} + t_{update} < \tau_d$).

Sensing and Data Transmission Phase Delay. The delays incurred in the three stages of the Sensing and Data Transmission Phase are modelled, and their maximum bounds are derived.

The first stage of the first phase is the synchronisation process. This occurs when a node has data to send and uses information in the RRM received from the gateway. The RRM message is sent periodically and of a fixed size such that the time on air of the message and its sending rate never violate the DC regulation. Therefore, the synchronisation process includes a fixed delay, denoted as t_{sync}, which is in the range $[0, t_{slot} \times k]$. In our system, $t_{sync} \in [0, 0.5]$ seconds (based on the implementation in Section 5.3.2 that uses $k = 5$ request slots of $t_{slot} = 0.1$ seconds).

After the delay of t_{sync}, the node needing to transmit enters the next two stages of phase one. Delays of the next two stages are denoted as t_{req} and t_{send}. Their bounds are analysed by modelling the two stages as two queues:

- **M/M/1 queue, or queue one** - The second stage of the first phase is the data transmission request process. This process can be modelled as an M/M/1 queue. A queue is used to analyse the delay, t_{req}, and its bounds as it allows one to reason about the occurrence of sensor events and the waiting time for a sensor node to receive a transmit slot for data communication. We define t_{req} as the mean time for a sensor node to request successfully and receive a data slot and is estimated by $1/(\mu_{req} - \lambda)$ where λ is the system load, less than or equal to the number of sensor nodes, and $\mu_{req} = ln(1/(1 - e^{(-\lambda/k)}))$ is the service rate. We assume that the inter-arrival of sensor events and time to get a transmit slot are exponentially distributed (these assumptions are based on the traffic pattern of an event-based control system as presented in Section 5.4.4). For the system to be stable $\lambda/\mu_{req} < 1$. The delay t_{req} is non-deterministic. In the best-case scenario, when the sensor node immediately successfully requests and receives a data slot $t_{req} = 0.2$ seconds. Otherwise, we can analyse that the probability of t_{req} is less

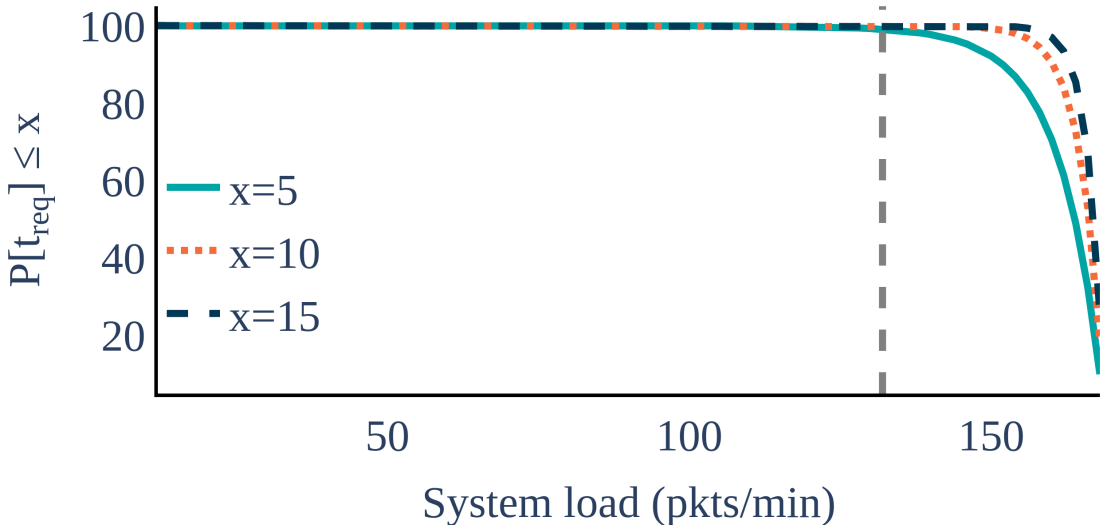


Figure 5.5: The probability of achieving $t_{req} \leq x$ for M/M/1 queue.

than a given delay which we denote as x. We show through numerical simulation the probability $P[t_{req} \leq x] = 1 - e^{(-(\mu-\lambda)x)}$ for various data loads. Results are presented in Figure 5.5. Our results show a very high probability (more than 99%) of meeting the delay bounds of 5 seconds for a communication capacity of up to 136 packets per minute and up to 150 packets per minute of meeting the delay bounds of 10 seconds. The packet sizes and time on air of packets at this rate do not violate the DC regulation.

- **M/D/n queue, or queue two** - The third stage of phase one is the data transmission stage. We model the delay of this stage, t_{send}, as an M/D/n queue. The term n is the number of data transmission channels available to *Ctrl-MAC*. We define t_{send} as the time for a sensor node to send its data to the gateway. The queue is deterministic because a sensor node receives the data slot and channel exclusively reserved for its transmission from the gateway during the data transmission request stage. The exclusive data slot and channel reservation ensure that the DC regulation is met and that the time to send is bounded. The delay t_{send} is equal to $1/n + (\lambda_{dt}/(2n \times (n - \lambda_{dt})))$ where λ_{dt} is the output of the M/M/1 queue given above and the input to the M/D/n queue described here. Based on the input rate, λ_{dt}, is in the range of $[12, 150]$ packets/minute based on Figure 5.5, the time to completion is $t_{send} \in [0.3, 0.45]$.

Control Update and Actuation Phase Delay. The final delay that we need to consider is the time for phase two, t_{update} when the controller sends the control input to the actuator node. The gateway transmits on a separate downlink channel reserved for communication from the gateway to the actuator. The use of a reserved channel ensures that the DC regulation is met and that the time of t_{update} is deterministic. The delay is dependant upon the size of the last actuation update sent. In our implementation, the maximum packet size that the controller can send is 222 bytes. Each actuation update is 2 bytes per actuator node, 1 byte for control input and 1 byte for addressing. The maximum number of actuator nodes that can be updated with a single packet is 111. The delay t_{update} is in the range of $[0.4, 3.6]$ seconds depending on the size of the last actuation update.

Therefore, the total delay introduced by *Ctrl-MAC*, t_{MAC} is:

$$t_{MAC} = t_{sync} + t_{req} + t_{send} + t_{update}. \tag{5.3}$$

According to the co-design constraint $\mathbf{C_1}$ from Section 5.2.2, the system stability is dependant upon the condition that $\tau_d > t_{ca} + t_t$ where $t_{ca} = t_{sync} + t_{req}$ and $t_t = t_{send} + t_{update}$. We derived the individual bounds for t_{sync}, t_{req}, t_{send}, and t_{update} while taking the DC regulation into account. A simple sum of all of our bounds shows that the total delay, t_{MAC} is in the range of $[0.9, 14.55]$ seconds with a very high probability of more than 99%. The communication delays of *Ctrl-MAC* do not violate the maximum allowable transmission delay of the system (i.e. $\tau_d > t_{MAC}^{Upper}$).

5.3.4 *Ctrl-MAC* and LoRaWAN

In this section, we motivate the need for *Ctrl-MAC* and explain the differences between *Ctrl-MAC* and LoRaWAN. We also show how LoRaWAN cannot meet communication requirements for control systems that are reliable communication, two-way communication and bounded communication delays. We evaluate *Ctrl-MAC* against two versions of LoRaWAN:

- *Version one: LoRaWAN* - This is a baseline version that represents the typical use case for LPWAN, such as data collection from sensors or smart meters. It exploits a pure ALOHA channel access scheme. The sensor nodes are the Class A devices that use unconfirmed (unacknowledged) messages, i.e. the sensor nodes do not

request an acknowledgement for their uplink transmissions. Hence, there is no guarantee for the successful delivery of the message. The actuator nodes are modelled as Class C devices. Our actuation messages are sent on this downlink channel to the actuators.

- *Version two: LoRaWAN++* - This is an improved version of LoRaWAN that enables reliable data communication to the gateway. It uses the ALOHA channel access scheme with an acknowledgement message from the gateway for every uplink transmission. If no acknowledgement is received, the transmitted data is resent up to 8 times, after which it is dropped. Acknowledgements are sent on the LoRaWAN downlink channel. In this version, we also add a dedicated downlink channel for actuation to make this scheme comparable to *Ctrl-MAC*.

We evaluate MAC protocol suitability for control applications with three metrics:

1. *End-to-End Packet Delivery Ratio* (E2E PDR in %) - E2E PDR is the ratio of the number of unique actuation updates received by the actuator node over the number of unique events generated by the corresponding sensor node. This metric indicates the suitability of the MAC protocols for control applications in terms of two-way communication. Downstream communication is as important as upstream communication.

2. *End-to-End Delay* (E2E Delay in seconds) - E2E Delay is defined as the difference between the time that an actuator node receives the actuation update and the time that an event is generated at the sensor node. We assume the computation delays to be zero when calculating this metric. This metric indicates the suitability of the MAC protocols for control applications in terms of bounded communication delays.

3. *Uplink Reliability* (UL Reliability in %) - UL Reliability is the ratio of the number of events successfully acknowledged by the gateway over the number of events successfully delivered at the actuator nodes. This metric gives the guarantees that the controller has complete and up-to-date information to maintain system stability. For example, a 10% uplink reliability would indicate that only 10% of the delivered messages from sensor nodes were reliably acknowledged. The sensor nodes may or may not have information about the remaining 90%, which introduces uncertainties in the system.

	Ctrl-MAC					LoRaWAN					LoRaWAN++				
Size	**10**	**50**	**100**	**150**	**200**	**10**	**50**	**100**	**150**	**200**	**10**	**50**	**100**	**150**	**200**
P(10s)	99.99	67.72	33.62	22.34	16.86	99.98	64.32	39.82	30.26	23.22	80.85	40.31	31.39	25.89	20.24
P(30s)	99.99	99.99	98.23	67.20	50.77	99.98	80.23	72.64	57.6	48.14	99.96	86.71	40.11	27.56	21.04
P(50s)	99.99	99.99	99.99	99.99	83.74	99.99	93.13	85.56	65.18	62.95	99.99	93.06	63.37	45.31	34.50
E(10s)	99.98	97.59	52.73	35.27	26.43	84.42	66.44	51.32	41.45	33.44	86.31	38.54	21.34	14.66	10.94
E(30s)	99.99	96.01	91.79	79.84	60.38	93.22	83.52	72.71	64.08	55.95	98.31	72.95	45.80	32.22	24.46
E(50s)	99.99	97.76	97.31	95.67	93.00	95.81	88.58	81.13	73.96	68.23	99.18	89.28	64.86	47.77	37.46

P=periodical, E=exponential

Table 5.1: E2E PDR (%) of *Ctrl-MAC*/LoRaWAN/LoRaWAN++ under different data loads and data distributions.

	Ctrl-MAC					LoRaWAN++				
Size	**10**	**50**	**100**	**150**	**200**	**10**	**50**	**100**	**150**	**200**
P(10s)	1.38	9.17	23.37	37.66	52.77	4.15	14.39	15.68	15.56	14.84
P(30s)	1.28	1.48	5.85	26.31	36.02	3.32	11.92	15.26	15.60	15.40
P(50s)	1.27	1.41	1.66	2.8	25.54	0.33	8.54	14.52	15.57	15.26
E(10s)	1.35	2.81	17.29	31.96	46.65	3.58	13.41	15.52	15.63	15.03
E(30s)	1.30	1.48	2.91	14.99	27.03	1.63	9.51	14.33	15.56	15.38
E(50s)	1.29	1.37	1.59	2.45	7.73	1.16	6.36	12.28	14.62	15.46

LoRaWAN E2E Delays are 0.15 seconds for all simulation scenarios.

P=periodical, E=exponential

Table 5.2: E2E Delays (s) of *Ctrl-MAC*/LoRaWAN/LoRaWAN++ under different data loads and distributions

We compare the E2E PDR, E2E Delay and UL Reliability of *Ctrl-MAC*, LoRaWAN and LoRaWAN++ assuming four channels, a varied number of nodes $(10, 50, 100, 150, 200)$ and two transmission patterns. First, we chose constant periodic transmissions to represent the worst-case scenario of the event-triggered control (i.e. at every sampling period, the transmission is triggered - in reality, there would be periods of no events). Each sensor periodically sends data with an interval of $10, 30, 50$ seconds. Second, we chose exponential transmissions with a mean of $10, 30, 50$ seconds to represent aperiodic and bursty traffic patterns specific to the event-triggered control. This range of sending rates (inter-arrival times) corresponds to the sampling interval of slow-loop control systems, as is shown in Section 5.4.4. Results are presented in Tables 5.1, 5.2 and 5.3.

	P(10s)		P(30s)		P(50s)		E(10s)		E(30s)		E(50s)	
Protocol	**10**	**200**	**10**	**200**	**10**	**200**	**10**	**200**	**10**	**200**	**10**	**200**
LoRaWAN	UL Reliability is 0% for all simulation scenarios.											
LoRaWAN++	98.76	34.62	99.92	37.81	99.99	39.41	99.46	35.36	99.96	41.91	99.99	50.81
Ctrl-MAC	UL Reliability is 100% for all simulation scenarios.											

P=periodical, E=exponential

Table 5.3: UL Reliability (%) of *Ctrl-MAC*/LoRaWAN/LoRaWAN++ for 10 and 200 nodes

The results demonstrate that LoRaWAN is not suitable for control systems as it stands; it was not designed for this. It achieves very low E2E PDRs (less than 90%), except in the case of tiny network sizes and low datarates. This implies failing to support two-way communication. LoRaWAN does not support acknowledgement messages which prevents reliable data transmission to the gateway. The E2E Delays are constant and low as LoRaWAN uses pure ALOHA. This implies that any contention-free transmission will have the same E2E delays, which would be a sum of the uplink and downlink time on airs which equals 0.15 seconds; however, the percentages of contention-free transmissions are very low as per Table 5.1. Stable controlled system operation that we derive in Section 5.4.3 can not be guaranteed with LoRaWAN.

Similar to LoRaWAN, LoRaWAN++ is also unable to meet the communication requirements for wide-area control systems. LoRaWAN++ achieves good E2E PDRs (more than 90%) for small network sizes and low datarates. However, for larger network sizes and higher datarates, there is a network performance deterioration; E2E PDRs are even lower than in the case of LoRaWAN. These results align with the results in [83] which show that if 100% of the nodes request an acknowledgement, the network can barely operate at 15% of its capacity compared to the scenario when no one requests an acknowledgement. LoRaWAN++ E2E Delays are always less than 20 seconds as it allows 8 retransmissions, after which the packet is dropped. This means that LoRaWAN++ can provide guarantees of bounded communication delays. However, it fails to guarantee reliable and two-way communication. Not all delivered messages are acknowledged (based on the results in Table 5.3) and this discrepancy increases with the number of nodes adding uncertainty to a closed-loop system.

Finally, the results show that *Ctrl-MAC* is able to meet all communication requirements for control systems for the networks of up to 150 nodes with a maximum datarate

of 1 packet per 50 seconds. Higher rates can be supported for the networks with less than 150 nodes (see Table 5.1). To increase a single-hop network's scale, one would include more gateways to have gateway diversity rather than handle more nodes per gateway. *Ctrl-MAC* achieves high E2E PDRs for a much larger number of scenarios when compared to LoRaWAN and LoRaWAN++. It achieves bounded E2E Delays and can guarantee UL reliability of 100% for all simulated scenarios. This UL reliability can be guaranteed because all data transmissions are scheduled; hence, there is no possibility of data collisions.

5.4 Control Model for a WDN

This section presents the system model for our water distribution network. We demonstrate how we use our C^3 approach to influence the controller design to meet the co-design constraint $\mathbf{C_2}$ in Section 5.2.2 and enable safe and reliable system operation. The communication requirements of the controller (based on the choice of $\{h_i, \tau_{di}, \sigma_i, \rho_i\}$) cannot be larger than the maximum communication network capacity. At the same time, the parameters $\{h_i, \tau_{di}, \sigma_i, \rho_i\}$ have to be chosen such they guarantee the system stability.

5.4.1 System Model & Controller Design

The water distribution network model considered in this work is shown in Figure 5.1. It consists of several DMAs with three or four water tanks. The system stability is defined as stabilising the tank water level to a pre-defined set point. Each DMA is instrumented with n_i sensor nodes (measuring the water level in the tanks) and m_i valves for actuation. Each DMA is modelled by the linear time-invariant model in Equation 5.1. In the state vector $\xi_i(t)$, each state is the difference between the current water level and setpoint reference water level for a tank. The control input vector $v_i(t)$ represents the degree to which the in-valves are open to fill each tank.

The system states are measured with water level nodes for each DMA supply tank. The states are communicated to the linear input-to-state controller given in Eq. 5.2. The values in $\hat{\xi}_i(t)$ contain either the previous measured values or new measured values of $\xi_i(t)$. A new measured value for a tank is used if the measured value violates the event condition (defined in the next section). Otherwise, the previous measurement of the

water level in the DMA tanks is used. The combination of previous and new values creates an estimation of $\xi_i(t)$. The error, or difference, between the estimated levels and the measured levels is

$$\varepsilon_i(t) = \hat{\xi}_i(t) - \xi_i(t). \tag{5.4}$$

From a control system point of view, the water tank levels in $\xi_i(t)$ can be viewed as the ones on nodes, and the values in $\hat{\xi}_i(t)$ as the values on the controller.

It is the task of *Ctrl-MAC* to send the new values from the sensors to the controller when required. This process is referred to as the sample-and-update of $\xi_i(t)$. The speed at which updates occur determines the system stability and the congestion of the communication network, which relates to the co-design constraint $\mathbf{C_2}$. The speed is regulated by the update mechanism given next.

5.4.2 Updating the Control Input

The traditional way of choosing when to update $\hat{\xi}_i(t)$ is periodic or Time-Triggered Control (TTC). In TTC, all of the nodes send updates at a fixed interval. This method wastes energy when no changes to the water levels warrant sending and can cause network congestion. Instead, this work focuses on Decentralised Periodic Event-Triggered Control (DPETC)[153]. DPETC reduces both the communication from the nodes to the controllers and the energy consumption of the nodes[149].

DPETC from [153] is summarised herein with modifications for our input-to-state feedback control system (to accommodate the co-design constraint $\mathbf{C_2}$). The equation:

$$\mathcal{T}_k := \{t_k | t_k := kh, k \in \mathbb{N}\} \tag{5.5}$$

denotes the periodic sampling sequence of the nodes in time. The term $h > 0$ is the time between samples and k is the sample number. The sensor s_{ij} of DMA i, where $j \in \{1, \ldots, n_i\}$ updates its state, $\hat{\xi}_{ij}(t)$, on the controller as:

$$\hat{\xi}_{ij}(t) = \begin{cases} \xi_{ij}(t_k), \text{ when } \mathcal{C}_{ij}(\xi_{ij}(t_k), \hat{\xi}_{ij}(t_k)) > 0 \\ \hat{\xi}_{ij}(t_k), \text{ when } \mathcal{C}_{ij}(\xi_{ij}(t_k), \hat{\xi}_{ij}(t_k)) \leq 0 \end{cases} \tag{5.6}$$

for times $t \in (t_k, t_{k+1}]$. The quadratic event-trigger condition $\mathcal{C}_{ij}(\xi_{ij}(t), \hat{\xi}_{ij}(t))$ defines when there is an event and a sensor needs to send its value ($\xi_{ij}(t)$) to update $\hat{\xi}_{ij}(t)$. The event condition $\mathcal{C}_{ij}$ is

$$\mathcal{C}_{ij}(\xi_{ij}(t), \hat{\xi}_{ij}(t)) = |\hat{\xi}_{ij}(t) - \xi_{ij}(t)| - \sigma_{ij}|\xi_{ij}(t)|. \tag{5.7}$$

Based on Equation 5.7, small σ_{ij} triggers new events faster, which means that the load to the communication network will be higher. With this in mind, a σ_{ij} is chosen that triggers the least number of events but still guarantees stable system operation as per the co-design constraint $\mathbf{C_2}$. In doing so, we perform stability analysis for our DPETC based on Borgers, Dolk, and Heemels[154]. The goal of the analysis is to define the maximum allowable delay τ_d and $\sigma_{ij} = \sigma$ for all i, such that the control system is stable. The stability analysis is presented next.

For our stability analysis, we define $\mathcal{J}$ to be a set that contains the states transmitted at time t_k and $\mathcal{J}_c$ to be a set of the states not transmitted at time t_k. We define Γ_j as a diagonal matrix where the jth diagonal element is 1, and the rest of the diagonal elements are 0.

5.4.3 Stability Analysis

The following stability analysis determines the maximum allowable delay τ_d for the chosen values of h, σ and ρ that renders the controlled system stable. Please note that we dropped denoting subsystems by i for the sake of clarity. However, the analysis refers to an individual subsystem (or DMA) given in Equation 5.1 and its controller in Equation 5.2 are valid for all other subsystems.

The condition for having a globally exponentially stable (GES) system is that there exists scalars $c > 0$ and $\rho > 0$ such that for all $\xi(0) \in \mathbb{R}^{n_s}$ for all $t \in \mathbb{R}^+$ the following holds: $|\xi(t)| \leq ce^{-\rho t}|\xi(0)|$. The stability proof follows from the Corollary III.3, and Theorem V.2 of [153], and Theorem IV.1 of [154], and is summarised in the following corollary.

Corollary 1. *Consider plant in Equation 5.1, controller in Equation 5.2, sampling sequence in Equation 5.5, event-triggered mechanism in Equation 5.6, and event-triggered condition in Equation 5.7. Let τ_d be a given maximum allowable delay and ρ be a given convergence rate. Assume there exist matrices $P_{0h} \succ 0$ and $P_{1d} \succ 0$, and scalars $\acute{\mu}^j_{\mathcal{J}} \geq 0$, $\acute{\mu}^j_{\mathcal{J}_c} \geq 0$,*

$\tilde{\mu}^j_{\mathcal{J}} \geq 0, \tilde{\mu}^j_{\mathcal{J}_c} \geq 0, \hat{\mu}^j_{\mathcal{J}} \geq 0, \hat{\mu}^j_{\mathcal{J}_c} \geq 0$, *where* $j \in \{1, \cdots, n_s\}$, *such that*

$$\begin{bmatrix} e^{-2\rho\tau_d}P_{0h} + G_1 & \acute{J}^{\mathrm{T}}_{\mathcal{J}} e^{\bar{A}^{\mathrm{T}}\tau_d} P_{1d} \\ (\acute{J}^{\mathrm{T}}_{\mathcal{J}} e^{\bar{A}^{\mathrm{T}}\tau_d} P_{1d})^{\mathrm{T}} & P_{1d} \end{bmatrix} \succ 0$$
$$\begin{bmatrix} e^{-2\rho(h-\tau_d)}P_{1d} + G_2 & \tilde{J}^{\mathrm{T}}_{\mathcal{J}} e^{\bar{A}^{\mathrm{T}}(h-\tau_d)} P_{0h} \\ (\tilde{J}^{\mathrm{T}}_{\mathcal{J}} e^{\bar{A}^{\mathrm{T}}(h-\tau_d)} P_{0h})^{\mathrm{T}} & P_{0h} \end{bmatrix} \succ 0 \tag{5.8}$$

where $G_1 = \sum_{j\in\{1,\cdots,n_s\}}(-\acute{\mu}^j_{\mathcal{J}}\acute{Q}_j + \acute{\mu}^j_{\mathcal{J}_c}\acute{Q}_j - \tilde{\mu}^j_{\mathcal{J}}\tilde{Q}_j + \tilde{\mu}^j_{\mathcal{J}_c}\tilde{Q}_j), G_2 = \sum_{j\in\{1,\cdots,n_s\}}(-\hat{\mu}^j_{\mathcal{J}}\hat{Q}_j + \hat{\mu}^j_{\mathcal{J}_c}\hat{Q}_j)$ *and*

$$\bar{A} = \begin{bmatrix} A & BK & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \acute{Q}_j = \begin{bmatrix} (1-\sigma^2)\Gamma_j & 0 & -\Gamma_j \\ 0 & 0 & 0 \\ -\Gamma_j & 0 & \Gamma_j \end{bmatrix}$$
$$\tilde{Q}_j = \begin{bmatrix} (1-\sigma^2)\Gamma_j & -\Gamma_j & 0 \\ -\Gamma_j & \Gamma_j & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{Q}_j = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Gamma_j & -\Gamma_j \\ 0 & -\Gamma_j & (1-\sigma^2)\Gamma_j \end{bmatrix}$$
$$\acute{J}_{\mathcal{J}} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \Gamma_{\mathcal{J}} & 0 & I-\Gamma_{\mathcal{J}} \end{bmatrix} \tilde{J}_{\mathcal{J}} = \begin{bmatrix} I & 0 & 0 \\ 0 & I-\Gamma_{\mathcal{J}} & \Gamma_{\mathcal{J}} \\ 0 & 0 & I \end{bmatrix}$$

Then, the system is GES with decay rate ρ.

The proof of Corollary 1 is given in Appendix A.2. Based on Corollary 1 and its proof, the feasible solution of Equation 5.8 for our water distribution network can be found when $h \in [1, 50]$ seconds, $\tau_d \in [0, 15]$ seconds and $\sigma \in [0.01, 0.3]$. The set of parameters are chosen that guarantee stable system operation and also meets the co-design constraint $\mathbf{C_2}$. It ensures that the network load is less than the maximum network capacity which is shown next.

5.4.4 System Operation

The start and steady-state (or the system's response) of our model water distribution network are demonstrated in Figure 5.6(a).

The model is implemented and simulated using Matlab/Simulink, the Simscape toolbox and the Fluids toolbox. The parameters of the physical components of the DMA

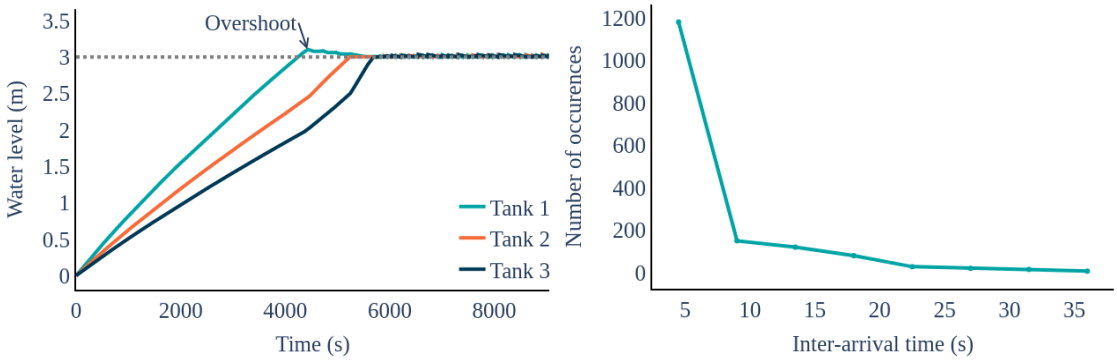


Figure 5.6: (a) System response to constant customer demand, (b) Distribution of inter-arrival times of the events.

(pipes diameters, valve types, tank sizes) are taken from the specification sheets of the same make and model valves, pumps, and tanks as those used by water companies.

An ideal case scenario is simulated with one DMA (with three tanks) and no packet drops or delays. The control system has a sampling time of $h = 4.5$ seconds and an event trigger parameter of $\sigma = 0.01$. σ regulates the rate at which events occur. For example, $\sigma = 0.01$ leads to an average of 22.36 events per minute and increasing it to 0.1 reduces the event rate to 13.87 events per minute. The event rate for both cases (which is the load to the communication network) is much less than the network capacity of 136 packets/minute, which is needed to ensure that the delays are bounded within a given threshold. This is presented in detail in Section 5.3.3. Therefore, the control design meets the co-design constraint $\mathbf{C_2}$.

Next, the number of occurrences of the inter-arrival times are shown in Figure 5.6(b). The inter-arrival times can be approximated to an exponential distribution with a mean of 4.5 seconds (equivalent to sampling time) using the curve-fitting toolbox. This supports our claim in choosing the M/M/1 queue with exponentially distributed inter-arrival times to model the input operation of *Ctrl-MAC*. Additionally, multiple exponentially distributed event traces (as in the case of more than one DMA) models aperiodic and bursty traffic[155] which is the reality of event-based systems. The following section evaluates the performance of the controller and communication protocol, or system, that was designed using our C^3 approach.

5.5 Implementation and Evaluation

This section evaluates a system designed using our C^3 approach that consists of a controller and communication protocol. *Ctrl-MAC* and its associated controller are together referred as the C^3 *system*. The evaluation is based on a co-simulation environment and real-world experiments. Four system parameters are measured to evaluate the success of C^3 system with respect to the co-design constraints defined in Section 5.2.2. The parameters are:

- *End-to-End Delay* (E2E Delay in seconds) - The time that starts when a sensor node senses an event and ends when the actuation update is successfully delivered to the actuator. This metric measures the success of the constraint $\mathbf{C_1}$.
- *End-to-End Packet Delivery Ratio* (E2E PDR in %) - The number of event messages received at the actuator over the number of messages generated by the sensor. For *Ctrl-MAC* this metric measures the success of the contention resolution and successful actuation updates, and this relates to the constraint $\mathbf{C_1}$.
- *System overshoot percentage* (OvSh in %) - The percentage of the overshoot above the reference value. It is calculated as the difference between the water level's absolute value and reference value divided by the reference value. We use this metric to describe the stability of our system that has to hold in both constraints, $\mathbf{C_1}$ and $\mathbf{C_2}$.
- *Average number of events per minute* (Events/minute) - The average number of events per minute measures the system load per minute. The more events, the more communication *Ctrl-MAC* needs to handle. This metric measures the success of the constraint $\mathbf{C_2}$.

These parameters are measured for various sampling intervals, h, and different event trigger parameters σ as defined in Section 5.2.1. The size of the system is varied, including different numbers of DMAs and nodes. A co-simulation is conducted, and the results are compared with real-world experiments to verify the success of the C^3 system. The following sections present the experiments and discuss the results.

5.5.1 Evaluation via Co-Simulation

The first stage of the evaluation of the C^3 system are performed on a custom co-simulation environment. This environment consists of a physical system simulator using MATLAB/Simulink and a communication simulator using the OMNeT++ network simulator. Matlab and OMNeT++ are both run on the same computer. Inter-process communication is used to enable communication between Matlab and OMNeT++ using UDP. We use FLORA[69] for our LoRaWAN simulator.

We compare our approach to a wired centralised network and LoRaWAN++ in the co-simulation environment. The wired network represents the best-case scenario of a periodic centralised controller with no packet drops or transmission delays. LoRaWAN++ uses the ALOHA contention scheme with acknowledgement messages and a retry limit of 8 messages. This is the only configuration of LoRaWAN that can provide partial guarantees in message delivery required by control applications, as mentioned in Section 5.3.4. A spreading factor of 7 and the default 4 channels specified in the LoRaWAN specification are used. The network is simulated with 3 DMAs and ten sensor and ten actuator nodes. The goal of the control system is to maintain the DMA tank water levels to a reference point of 3 metres as time evolves. With the definition of overshoot percentage given above, a 4.5m high water tank with a 3m reference point overflows, or critically fails, with an overshoot above 50%. The customer demand is expressed as the degree that the out-valve of the tank is open. The maximum is 100%, and the minimum is 0% when the valve is closed.

Study 1: Matching Constant Customer Demand. In this co-simulation, we keep the customer demand constant at 100%. All experiments were run for 9000 seconds.

The results in Table 5.4 show that the C^3 system and LoRaWAN++ both maintain the stability of the system with similar overshoots. It is because both *Ctrl-MAC* and LoRaWAN++ always send the latest reading and discard any messages that were not sent successfully. However, the E2E PDR of *Ctrl-MAC* is on average 30% higher and more consistent than the E2E PDR of LoRaWAN++. The E2E Delays was bounded in all cases below 2.2 seconds for Ctrl-MAC. The LoRaWAN++ E2E Delays are, on average greater than 6 seconds. With the C^3 system the events/minute are always less than 136 packets/minute, indicating that the co-design constraint $\mathbf{C_2}$ from Section 5.2.2 is being met, and the network load remains within the network capacity.

C^3 system	Events/minute	OvSh (%)	E2E PDR (%)	E2E Delay (s)
$h = 1$s, $\sigma = 0.1$	113.15	20.59	77.33	1.849
$h = 4.5$s, $\sigma = 0.01$	61.17	32.11	87.03	2.161
$h = 4.5$s, $\sigma = 0.1$	40.32	3.1	88.74	2.194
$h = 4.5$s, $\sigma = 0.3$	22.56	4.39	88.01	1.861
$h = 10$s, $\sigma = 0.1$	14.91	16.25	89.27	1.919
Wired Control	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delay (s)**
$h = -1$s*	inf	2.73	100	0
$h = 1$s	600	2.73	100	0
$h = 4.5$s	133.33	2.75	100	0
$h = 10$s	60	2.75	100	0
LoRaWAN++	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delay (s)**
$h = 1$s, $\sigma = 0.1$	115.8	3.15	31.4	6.143
$h = 4.5$s, $\sigma = 0.01$	63.35	29.04	51.04	9.859
$h = 4.5$s, $\sigma = 0.1$	35.34	6.38	52.4	9.725
$h = 4.5$s, $\sigma = 0.3$	20.26	5.37	59.01	8.089
$h = 10$s, $\sigma = 0.1$	15.03	16.63	81.22	8.759

*the system is continuous in time

Table 5.4: The evaluation of the C^3 system for constant customer demand in comparison with an idealistic wired control and LoRaWAN++ for control.

Study 2: Matching Dynamical Customer Demand. In this co-simulation, we model customer demand from actual traces of daily domestic water consumption inspired by [156] and [157]. Figure 5.7 shows the typical demand of a water network over 24 hours in an urban environment. It is a tri-modal curve in which the demands are highest during the mornings and evenings and a slight bump during the daytime with the lowest demand at night. This demand is simulated by controlling the degree to which the out-valve is open.

The results in Table 5.5 show that the C^3 system and LoRaWAN++ can both maintain systems stability through keeping small overshoots. The events per minute for the C^3 system are half of those for LoRaWAN++ with $\sigma = 0.3$ and are within the maximum load that the system can handle as described in Section 5.3.3. The C^3 system also has nearly twice the E2E PDR and 75% smaller E2E Delays than LoRaWAN++. The C^3 system's E2E delays are bounded by a maximum of 2.1 seconds. The E2E delays for LoRaWAN++ are 8 seconds on average. The increase in E2E Delays and reduction of

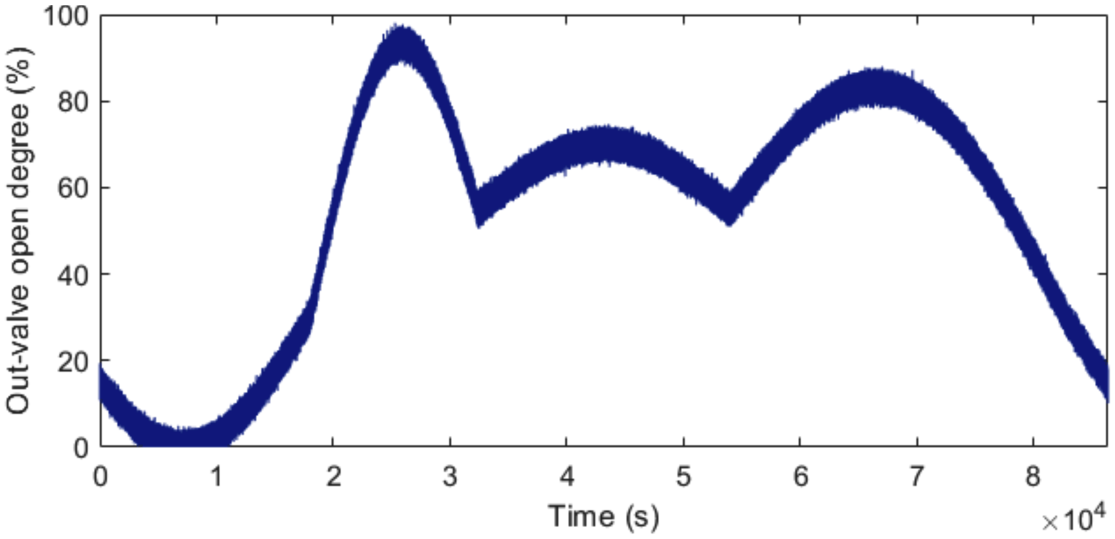


Figure 5.7: Daily customer demand (24 hours)

C^3 system	Events/minute	OvSh (%)	E2E PDR (%)	E2E Delays (s)
$h = 4.5$s, $\sigma = 0.1$	50.33	6.67	87.3	2.043
$h = 4.5$s, $\sigma = 0.3$	34.80	6.0039	88.88	1.825
Wired Control	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delays (s)**
$h = 4.5$s	133.33	3.44	100	0
LoRaWAN++	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delays (s)**
$h = 4.5$s, $\sigma = 0.1$	69.85	6.463	46.13	8.546
$h = 4.5$s, $\sigma = 0.3$	58.71	7.868	55.2	8.268

Table 5.5: Evaluation of the C^3 system for dynamic customer demand compared to idealistic wired control and LoRaWAN++ for control.

E2E PDR shows that a change in customer demand causes the LoRaWAN++ network load to approach its capacity and struggles to meet the co-design criteria $\mathbf{C_2}$.

Study 3: Fault Detection. An essential requirement of a WDN is fault detection when pipes burst due to sub-zero temperatures or sudden pressure changes. A pipe burst causes water leakage, which is another source of demand on a system. In this scenario, we assume that the sensors can take readings from the pipes (flow or vibrations) and determine a leakage.

The leak detection mechanism identifies a fault and communicates this to enable leakage mitigation. The controller closes the valves that provide water to the affected

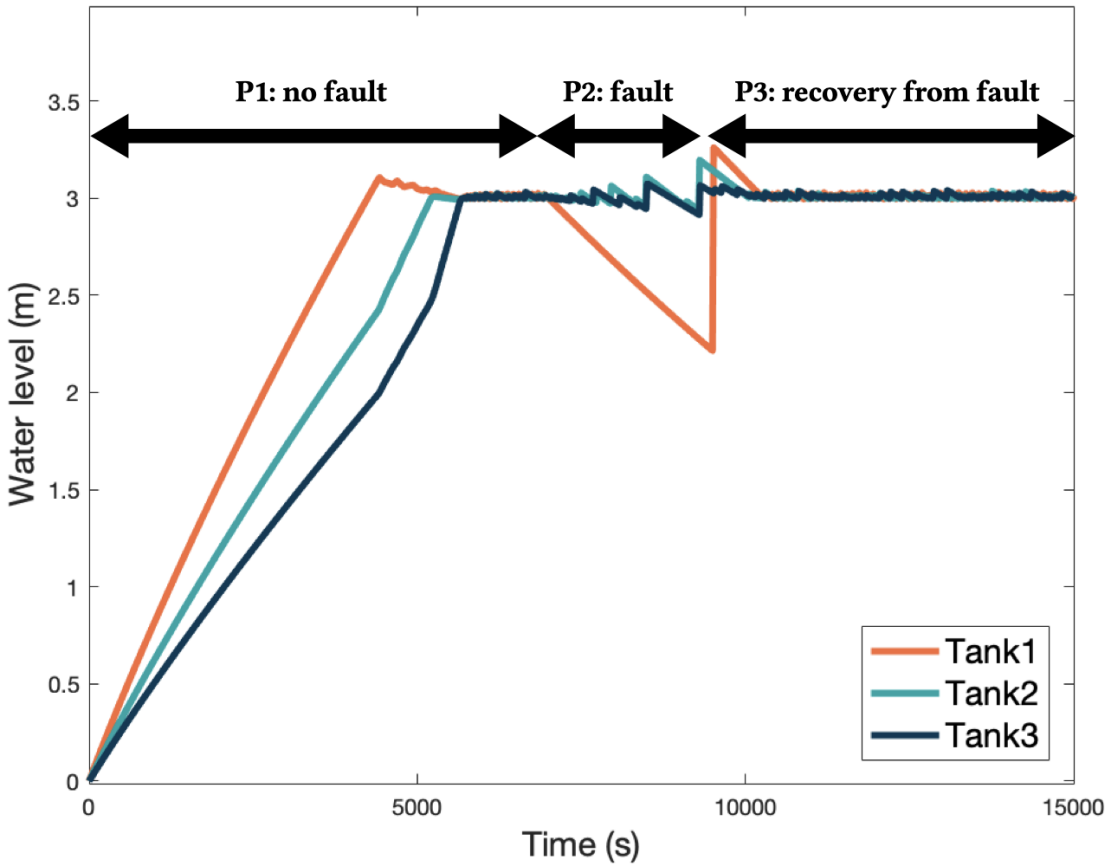


Figure 5.8: System response to a fault in the tank 1.

area, re-routing water and preventing waste. Kartakis et al.[158] provide an example of this with leakage localisation to 50cm [159].

The system state response is depicted in Figure 5.8 and has three phases: no-fault (P1), fault (P2) and recovery from fault (P3). Our results are presented in Table 5.6. The overshoots are comparable, and stability is maintained. The E2E PDR of the C^3 system is almost twice that of LoRaWAN++. The E2E Delays are less than 2.1 seconds. The number of events/minute is within the maximum network load of 136 packets/minute and achieve high reliability.

The results in Table 5.5 show that LoRaWAN++ maintains system stability not by design but by accident. The E2E Delays of LoRaWAN++ are four times higher than those of the C^3 system. Analysis of the simulation traces reveal that LoRaWAN++ succeeds in sending uplink messages to the gateway, but gateway acknowledgements are lost

C^3 system	Events/minute	OvSh (%)	E2E PDR (%)	E2E Delays (s)
$h = 4.5$s, $\sigma = 0.1$	**P1:** 25.03 **P2:** 64.34 **P3:** 58.98	3.3087 16.059 35.663	91.35	2.029
$h = 4.5$s, $\sigma = 0.3$	**P1:** 14.07 **P2:** 41.87 **P3:** 39.18	4.4741 6.3206 13.260	92.25	1.789
Wired Control	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delays (s)**
$h = 4.5$s	**P1:** 133.33 **P2:** 122.65 **P3:** 133.32	2.75 5.9129 10.23	100	0
LoRaWAN++	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delays (s)**
$h = 4.5$s, $\sigma = 0.1$	**P1:** 25.52 **P2:** 69.35 **P3:** 72.50	3.0926 19.29 17.62	50.35	8.465
$h = 4.5$s, $\sigma = 0.3$	**P1:** 15.08 **P2:** 38.25 **P3:** 38.34	4.1367 7.4541 4.2372	60.94	6.875

P1 - no fault, P2 - fault, P3 - recovery from fault

Table 5.6: The evaluation of the C^3 system for constant customer demand with fault in comparison with an idealistic wired control and LoRaWAN for control.

due to duty-cycle, DC, constraints. The loss of acknowledgements triggers a resend. The gateway forwards the received, unacknowledged sensor value to the controller, regardless of whether the sensor receives an acknowledgement. The sensor value leads to an actuation update and a stable system. The traces show that LoRaWAN++ is incapable of meeting our co-design constraint $\mathbf{C_2}$ of maintaining the network load within its capacity. The freshness property of the sent data also enables LoRaWAN++ to maintain system stability. When a node fails to receive an acknowledgement before an event causes the new data sampling, the new data replaces the old in the re-transmitted packet. In lost acknowledgements and re-transmissions, the controller always receives the newest data and maintains system stability. The subsequent study investigates whether the stability by accident of LoRaWAN++ scales to a large-scale system.

Study 4: Large-Scale System. In this co-simulation, we model a more extensive system with 10 DMAs consisting of 32 sensor and 32 actuator nodes. We use constant

C^3 System	Events/minute	OvSh (%)	E2E PDR (%)	E2E Delays (s)
$h = 1$s	253.93	32.44	63.35	3.36
$h = 4.5$s	115.20	22.39	88.16	4.01
LoRaWAN++	**Events/minute**	**OvSh (%)**	**E2E PDR (%)**	**E2E Delays (s)**
$h = 1$s	238.77	46.80	26.9	11.28
$h = 4.5$s	174.26	63.03	45.2	14.9

Table 5.7: The evaluation the C^3 system for 10 DMAs (32 nodes) in comparison with LoRaWAN++ for control.

customer demand, two sampling intervals (h) of 1 and 4.5 seconds, and we run the experiment for 10000 seconds.

The results in Table 5.7 show that at this scale, LoRaWAN++ is unable to maintain overshoots within the system boundaries, and the controller is unable to maintain system stability. The C^3 system can maintain stability while achieving overshoots of 22% and a good performance in E2E PDR and E2E Delays, consistent with the three previous studies. 115 events/minute are generated with $h = 4.5$ seconds. *Ctrl-MAC* can provide a high degree of reliability with upto 136 minutes/minute. Given that with the same parameters and 3 DMAs (study 1), 40 events/minute were generated. A single gateway should be able to handle between 12-13 DMAs while keeping the delays and reliability within the limits needed for C^3s stability.

To conclude, our co-simulation results show that when the scale of the system increases by a factor of 3, LoRaWAN++ can no longer keep the network load within the network's capacity and maintain system stability. This confirms our statement in Section 5.3.4 that the ALOHA channel access scheme is unable to provide the communication requirements for control and our motivation for the creation of the two co-design constraints for our C^3 approach and the *Ctrl-MAC*.

5.5.2 Real World Evaluation

This section looks at the performance of the C^3 system designed using the C^3 approach on real hardware. For completeness, we apply a hybrid approach that combines a physical LoRa communication network with a simulated water supply network to create a hardware-in-the-loop experiment and a hardware-in-the-park experiment.

This stage of the evaluation validates the findings from the simulations showing that *Ctrl-MAC* and the controller can function correctly when tested with the uncertainty of real-world radio behaviours.

Real-world WDNs have mixed environments where they are deployed indoors and outdoors. The hardware-in-the-loop experiment is carried out to characterise the behaviour of the C^3 system in an indoor environment. In this experiment, there is no line of sight between nodes. They are separated by plaster, brick and metal walls at distances on the order of tens of meters. Ten nodes are divided into three groups, each representing a single DMA.

The hardware-in-the-park experiment is done in a $2 \times 2\text{km}^2$ urban park. This experiment has buildings, trees and large ornamental structures blocking line-of-sight. The distances of the hardware-in-the-park experiments are in the order of hundreds of meters. In this experiment, eight nodes are divided up into two DMAs.

Ctrl-MAC is implemented on all nodes. Each node consists of an Adafruit Feather M0 RFM95 LoRa board mounted on a Raspberry Pi Model 3 with large capacity batteries to power both. The gateway for the LoRa network is a MultiConnect Conduit Gateway connected to a desktop via Ethernet. The gateway uses an 868MHz +3dbi whip antenna. The packet forwarder software is altered to provide a timestamp to the upper layers of the gateway stack running on the desktop. The desktop runs Gateway-Bridge to receive communication from the gateway. The Gateway-Bridge then sends the messages to LoRa Server for decryption. Following this, the LoRa Server sends the decrypted messages to an App Server. The entire software stack on the desktop is provided by ChripStack[2].

Study 1: Hardware-in-the-Loop. Experimental setup is given in Figure 5.9 (a).

The nodes are provided with a constant demand workload from the physical system simulator, as seen in the first co-simulation experiment. E2E delays and overshoot are recorded to measure the system's ability to satisfy τ_d and assess our approach's ability to maintain system stability.

The results are presented in Table. 5.8. There is a higher delay in the hardware-in-the-loop experiments compared to the simulations due to network effects. These are due to issues with the hardware implementation at the gateway, and unidentified latencies in the LoRa server software stack, as the current LoRaWAN stack, is not

[2] https://github.com/brocaar/chirpstack-network-server

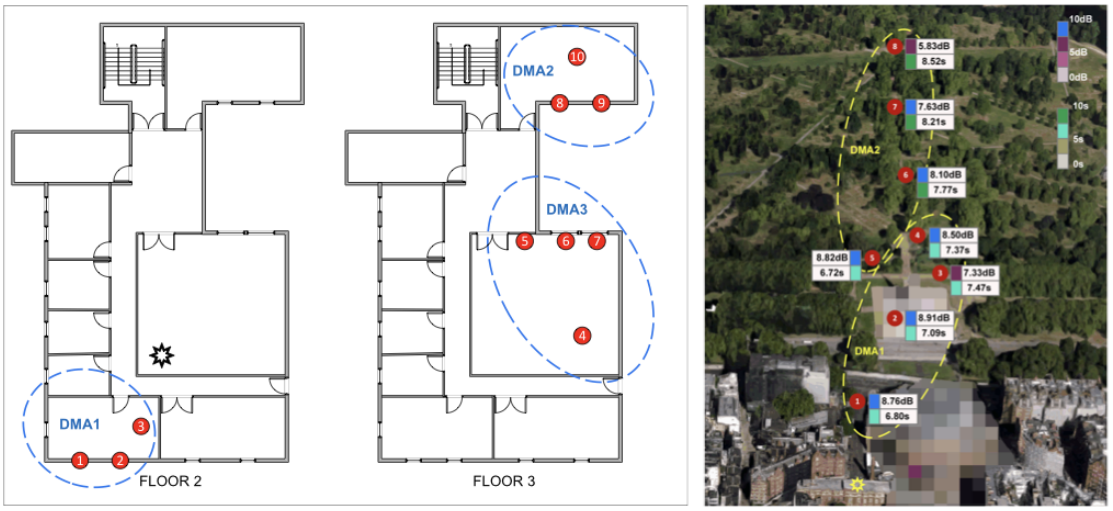


Figure 5.9: (a) Hardware-in-the-Loop experimental setup, (b) Hardware-in-the-Park experimental setup with up-stream delay and SNR values as seen at the gateway (the gateway is marked by the star, individual nodes locations by red circles).

designed for real-time operation. Despite these added delays, the overshoots are still small and show that our C^3 approach can maintain system stability.

Node ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Round Trip Time (s)	4.8	4.7	4	4.85	5.78	5.67	4.32	5.21	3.97	5.7
Overshoot (%)	4.98	17.25	3.8	8.4	12.36	7.22	9.64	4.98	16.6	13.3

Table 5.8: Round Trip Times and Overshoot for Hardware-in-the-loop experiment.

Study 2: Hardware-in-the-Park. The experimental setup is shown in Fig. 5.9 (b). This setup is chosen to provide realistic radio noise and effects such as the signal strength drop caused by distance and random interferences. Real-world disturbances can be complicated to categorise and model in simulation. It is important to note that the small +3dbi whip antenna used on the gateway limits the achievable communication range. The gateway placement was also far from ideal and reduced our overall range.

This experiment differs from the previous hardware-in-the-loop because it does not use input directly from the simulator. Instead, this experiment uses event traces recorded from the simulator on every node. The Round Trip Times (RTTs) and Signal-to-Noise Ratio (SNR) are measured at the gateway for each node in this experiment. RTTs are defined as the difference between the packet's generation and acknowledgement

times.

The results are depicted in Figure 5.9(b). It can be seen that the average delay between two successful events is small, very similar to the previous hardware-in-the-loop experiments. The SNR shows that there is a correlation between distance, SNR and delays. This is an indication of higher packet loss and re-transmissions. The results show that even in the presence of an unstable radio link environment, the C^3 system keeps the delays bounded. The bounded delay times observed in both of our real-world experiments validate the co-simulation results. The results show that our C^3 approach is helpful to design control and communication protocols that can maintain the stability of WDNs.

5.5.3 Evaluation Conclusion

The results of the evaluation clearly show that the use of our C^3 approach enabled the co-design of *Ctrl-MAC* and a controller that can maintain the E2E communication delay bounds required to keep a WA-CPS physical system stable. The results in Table 5.6 show that LoRaWAN++ and the C^3 system can maintain system stability on a 3 DMA system, although LoRaWAN++ has an E2E PDR of 50% to 60% compared to 91-100% for our C^3 system. At this scale, LoRaWAN++ enables stability for small scale control systems by accident as described in Section 5.5.1. As can be seen from Study 4, this situation does not scale well, and LoRaWAN++ fails when the system scale increases. Table 5.7 shows that with 10 DMAs and 32 sensors, LoRaWAN++ allows the overshoots to get as large as 63%. The C^3 system keeps the overshoots down to 22% for the same parameters. The E2E PDR falls to 26% for LoRaWAN++, while the C^3 system has an E2E PDR of 63% at a sample interval of 1 second. Thus, the results of the evaluation show benefits of the C^3 system over LoRaWAN++.

5.6 Discussion, Limitations and Future Work

While this chapter aimed to demonstrate that LoRa can indeed support control of WDNs, there are some limitations to our exposition that are reserved for further research.

Limitations of the approach. Ctrl-MAC and its associated event-triggered controller has an overshoot of 22% and delays of 4 seconds, with a sampling time of 4.5 seconds.

The stability analysis presented in Section 5.4.3, showed that with a sampling time of 4.5 seconds and delays of less than 15 seconds, the system stability can be maintained. However, the analysis accounted for only one DMA (with 3 tanks), and analysis for multiple DMAs operating simultaneously was not done. A further stability analysis needs to be conducted, to find a feasible solution for a several DMAs operating simultaneously. Based on the results of the control system stability, the communication system may have to be changed to handle the requirements of the control system.

Energy Savings. Energy usage and savings are not explicitly discussed. It is implicit from the large volumes of communication messages saved by using an event-triggered controller. An assumption was made that actuation is relatively power-hungry (e.g. a pump), and so they are either wired or use energy harvesting systems. So actuators are modelled as listening to the channel at all times; therefore, no energy costs are explicitly presented. For future work, we can look at minimising the listening time of the actuators to save energy.

Maximum number of nodes. A maximum of 150 nodes per gateway was possible with Ctrl-Mac. In real-world deployments of LoRa, 1000s of nodes are in every gateway's communication range. We have seen how we handle a large number of nodes with a multi-gateway setup in Chapters 3 and 4. However, it is unclear how these nodes could impact Ctrl-Mac's performance or how Ctrl-Mac should work in a multi-gateway environment. Multi-gateway environments could expose scheduling problems to manage feedback messages or slot allocations; therefore, work on slot allocation to cope with overlapping communications is needed.

Future Applications. Robust analysis of a communications protocol in a control scenario dictates its analysis in terms of a canonical physical system, in our case, the water distribution network. This system was chosen as it represents the dynamics of many multi-kilometre pipeline systems. Ctrl-Mac can be further extended for other applications like monitoring oil and gas pipelines, smart meters or any other wide-area slow-loop feedback systems. These works would demonstrate the approach's generality and open its use in other areas.

5.7 Related Work

Much work has been done in recent times to study wide-area systems with fast-feedback loops. This work has been done on energy grids as shown[160, 161, 162]. In these works, the authors propose control strategies that use sparse communication schemes. Dibaji et al. [162] propose communications with analytical guarantees and in [161] they propose distributed control with large communication delays. These systems have higher sampling rates (6-60 samples/sec) than a slow-loop feedback system (13 samples/minute), as studied in this chapter, and the latency requirements are in the order of milliseconds compared to seconds in our work.

Control schemes. A complete survey for Networked Control Systems(NCS) can be found is done by authors in [163]. There are several aspects of NCS like security, sampled-data control, event-triggered control and network control. In this chapter, we were mainly focussed on network and event-triggered control. Two major types of control schemes can be used for the control of physical plants: time-triggered control and event-triggered control. Time-triggered control has been studied a lot in literature, and a lot of the work in the field of wireless CPSs use time-triggered control schemes. A novel and interesting work proposed in 2007 is in the field of event-triggered control[164, 165, 166]. In these works, the authors examine the effects and bounds required to perform event-triggered control. They use canonical control systems that do not reflect the complexity of real-world practical physical systems. Mazo and Tabuada[164] ignore the complexity of disturbances, and all sensors communicate local measurements synchronously. In [166], the event-trigger mechanism and controller are decentralised, making local computation complex (may not be suitable for low-powered sensor devices) and expensive. These works have shown how event-triggered controller reduces the communication bandwidth required to ensure closed-loop stability which makes it highly beneficial for wireless systems. Another survey on distributed event-triggered estimation over sensors networks is conducted in [167]. The survey discusses the difficulties of implementing event-triggered systems in low constrained devices, network self-organization, communication delays and security. The work in this chapter specifically targets wireless NCS that use event-triggered control. The chapter also shows how low bandwidth communication system like LoRa with long airtimes (leading to large latencies) can maintain system stability by choosing appropriate sampling and

triggering parameters. LoRa as a communication backbone may not work for all NCS applications but it is suitable for applications that have a slower system dynamic.

Wireless control. Wireless CPS are an extension of the traditional wireless sensor actuator networks with the addition of taking into account the influence of the chosen action on a physical plant. The research in this domain is growing. Park et al.[18] have performed a detailed analysis on wireless network design on control systems. The survey does a detailed comparison on time-triggered control, event-triggered control, energy consumption, message delays, message losses, effect of packet delays and losses.

Initial work in the field of control and communication focuses on applications that initiate an actuation based on thresholds, and examples are: green house climate monitoring[168],adaptive traffic light control [169, 170] and many other. However, these works are relatively simplistic and do not formally consider control complexities or reason about the system's stability or computational bounds that each physical system requires.

Industrial standard. To allow for rapid adoption of wireless systems in industry, two industrial standards were created. They are WirelessHart[23] and ISA100.11a[171]. These standard have gained favour for industrial control processes, using 802.15.4 2.4GHz ISM band and TDMA-based scheduling approaches, and more recently IEEE 802.15.4e TSCH [172] using Time-Slotted Channel Hopping. All these standards have been designed to work for short-ranges, and do not take the underlying physical system into account. This has led to a slow-adoption of these systems in real-world deployments[25].

Short range control systems. Short-range control systems can be segregated based on several parameters like system dynamics, network architecture, wireless protocols, triggering mechanisms. Many works in literature propose control over short-range protocols like 802.15.4, Wi-Fi. These works can be divided into four categories based on the system dynamics and the network architecture. The physical system dynamics can be slow or fast, and the network topology can be single-hop or multi-hop. This is shown in Table 5.9.

Schindler et al.[172] have experimented with duty-cycling and radio behaviours to implement a control loop that maintains latencies well below a critical delay to support the problem of a single pendulum. This work does not implement a distributed system or wide-area control nor uses low-powered devices. Mager et al.[177] derived

	multi-hop	single-hop
slow-loop	Adaptive lighting [173] Data center management [175]	Double tank system [174] **WDNs long-range**
fast-loop	Inverted pendulum [177] Batch Reactor [152] Water-irrigation system[181]	Dryer plants [176] Gait rehabilitation system [178] Inverted pendulum [179, 180] Water distribution network [149]

Table 5.9: Wireless CPS categories with hardware implementations

schemes with guaranteed closed-loop stability for linear dynamic systems over low-power multi-hop networks. This end-to-end design used a communication system based on constructive interference for control message flooding. It assumed a large concentration of devices in the same region performing similar tasks to ensure that the control messages reach all physical systems in a short amount of time. Because the system runs on a multi-hop network, the energy consumption of the nodes is high, and they are awake for 15-40% time based on the sampling interval. This work is over LAN scaled areas and not the wide-area multi-sub-system distributed control of this chapter.

Zimmerling et al.[182] propose an adaptive communication protocol which provides end-to-end guarantees in a multi-hop network. They propose, Blink, a protocol that can dynamically adapt to the needs of a network and provide hard guarantees in order of 100ms in a four hop network. This protocol runs on top of Glossy[183] which is a MAC layer based on constructive interference that provides the required redundancy. Lu et al. [152] propose real-time scheduling over WirelessHart. In this work, the authors performed experiments on a testbed with an 802.15.4 communication scheme and use it to control a two-tank system. Kartakis et al. [149] built on top of this work and propose centralised and decentralised event-triggered control of water systems. It assumed small LAN scale systems using Wi-Fi for communication on a small-scale testbed. In [174] a TDMA protocol with reserved slots is proposed. The system has double tanks in a small area, uses Zigbee for communication, and takes 20 seconds to reach equilibrium. A TDMA protocol will not scale well to a large amount of low-powered nodes. These systems are also relatively faster when compared to a WA-CPS.

Several protocols are designed on top of 802.15.4 and Wi-Fi for control of slow and fast systems. These systems do not scale well for a large-scale system, so LoRa and event-triggered control are ideal combinations to design for WA-CPS. Trobinger et al.[181] use event-triggered controller with concurrent transmissions to have multi-hop fast-feedback loops for control of a Water-irrigation system. They also leveraged the hardware-in-loop approach to create a virtual simulation environment as it is difficult to recreate the actual application.

5.8 Summary

This chapter targeted control of WDNs, a wide-area CPS application for which LoRa is a suitable communication protocol. LoRaWAN has reliability issues because of the ALOHA channel access and duty-cycle limitations which make it unsuitable for applications that need bounded delays. To deal with these issues, this chapter presented *Ctrl-MAC* and its associated controller that balances the control requirements with the communication limitations imposed by LoRaWAN's design. The protocol is a product of a Control Communication Co-design (C^3) approach. The C^3 approach specifies a set of control and communication parameters and the constraints that couple them to create a controller and communication protocol that can maintain system stability.

The results show that Ctrl-MAC and its controller, referred to as the C^3 system, can maintain system stability under real-world conditions. The results of the co-simulation evaluation showed that neither LoRaWAN nor LoRaWAN++ can provide data reception guarantees to the sensors or the control system. The combination of guarantees and low latencies makes Ctrl-MAC the only LoRa system suitable for the long range communication and control of wide-area applications.

In the last three chapters, we studied how leveraging node-to-node and gateway-to-gateway communication can meet the shortcomings of LoRaWAN. The protocols were able to meet requirements for networks with a relatively low network bit flux. However, data handover schemes to increase reliability would not meet the requirements for a wireless control system with a high network bit flux, so a communication and control system was co-designed to meet their requirements. We also studied how increasing the transmission rates, reducing the number of gateways and nodes, and deployment has an impact on the packet delivery ratios and end-to-end delays, as it increases the amount of

traffic any gateway has to handle that in-turn affects the ability of the protocols to meet the application requirements. This thesis showed that LoRa is currently under-utilised and can be used for CPS applications previously unthought. The final chapter will conclude the thesis, show how the research questions have been answered, a short discussion on metrics and other wireless systems and future developments.

6

Conclusion and Future Work

This PhD began with the aim of defending the following: **Low datarate, long-range communication systems like LoRa can be used for wide-area CPSs by using distributed algorithms to appropriately exploit the communication links between the cyber and physical layers**.

To defend this aim, we first explored the WA-CPSs where LoRa's long-range and low-datarate capabilities can bring some benefits. We identified two main applications: critical infrastructures and vehicular networks. We also wanted to study the problems where multiple overlapping LoRaWAN networks co-exist, sharing the same wireless spectrum with several WA-CPS applications running at the same time. After understanding the applications and their requirements, we studied which of LoRa's under-utilized features can be leveraged to create new distributed algorithms to meet the application's requirements. This led to the three main contributions of the thesis.

Chapter 3 focused on delay-tolerant vehicular networks. The mobile nodes suffered from unreachable gateways due to a low deployment density of gateways. LoRa's symmetrical modulation scheme allows for node-to-node communication that is under-utilised in LoRaWAN and can help deal with unreachable gateways. This chapter proposed a node-to-node data handover scheme where nodes made decisions about which node and how much data to handover. To achieve this, a distributed real-time metric, RCA-ETX was proposed. RCA-ETX modelled the link conditions between nodes and gateways. RCA-ETX was further combined with queue sizes and ROBC was proposed. In ROBC, instead of handing over all the data to its neighbour, nodes made decisions about how much data to handover depending on neighbour's queue size. This significantly increased throughput while maintaining low latencies achieved

by RCA-ETX. The work showed that LoRa's node-to-node capabilities could be used for mobile WA-CPSs to reduce latencies.

Chapter 4 focused on multiple overlapping LoRaWAN networks sharing the wireless resource. As multiple networks start sharing the same resource, the networks' goodput reduces as the retransmissions increases to get the data across. This chapter proposed a fully distributed and autonomous wireless overlay network that facilitates gateway-to-gateway communication between gateways belonging to different stakeholders. The chapter proposes two learning-based algorithms, RMIP and InterPred that learn the macro trends about a node's transmission patterns and communication channel and are used to make real-time decisions about lost messages and the channel to use in IRONWAN. The work showed that LoRaWAN's ad-hoc deployment model can be leveraged to benefit all networks with overlapping communication ranges and meet the requirements of WA-CPSs in areas with multi-stakeholder networks.

Chapter 5 focused on critical infrastructures like water distribution networks. This application had strict requirements of *high reliability*and *bounded delays*. To meet this application's requirements, the chapter leveraged a communication-control co-design scheme to design Ctrl-Mac and its associated event-triggered controller. The system decided when to communicate data (based on the event-triggered controller) and the communication system aimed to meet the latency requirement by scheduling the communication between a single gateway and multiple nodes. The work showed that LoRa can enable closed-loop control of slow-loop feedback systems like water distribution networks, which were previously unthought.

The above contributions demonstrate that by using distributed algorithms, low-datarate and long-range wireless systems like LoRa can be used for WA-CPS applications. These algorithms can make decisions about what, when, to whom and how much data to transmit within the limitations of the low-datarate system and keeping in mind the impact of long-range communication. These contributions show that there are several aspects of LoRa that are still under-utilized and they can be leveraged along with the standard LoRa parameters to achieve a reliable long-range communication system that targets WA-CPS applications.

6.1 Discussion

After defending the thesis, the next question is what are the tradeoffs between reliability, latency and density when it comes to the application's this thesis targets, and the similarities between the ideas presented with Age of Information, a new metric in the CPS space. The work also leads to a natural comparison of the work with Cellular and NB-IoT.

6.1.1 Metrics

Age of Information. Age of Information (AoI) is a new metric that has emerged in CPS research. The metric represents the age of information at the gateways taking into account the sampling intervals, queuing delays and transmission delays. Specifying requirements in terms of AoI is not common, so we used the classical metrics of PDR and latencies. However, the idea of AoI is inherent in RCA-ETX and C^3. RCA-ETX measures the link conditions between the node and gateways. It increases linearly in time and resets when a message is successfully delivered, similar to how AoI operates. In C^3, a new event-triggered when a node is queuing leads to the old message being replaced with the new data. This does not impact AoI directly but ensures that the data available at the gateways are as fresh as possible, which in turn promotes a smaller average AoI[184]. The benefits of AoI are that it allows for an easy-metric for communication and control communities to ensure system stability. AoI needs to be studied in more detail to see how they can be leveraged for WA-CPSs.

Reliability, Latency and Density tradeoff. During our studies, we observed an interesting tradeoff between reliability, latency and density. LoRaWAN has been designed for a large number of nodes with minimal reliability requirements. Applications requiring higher reliabilities, like the ones tackled in this thesis, cannot use LoRaWAN directly. We proposed node-to-node and gateway-to-gateway communication schemes to increase the reliabilities and reduce delays. However, we started using the already existing gateway to node bandwidth for other communications to increase reliabilities by adding these extra communication capabilities. Depending on the retransmission mechanisms and how our schemes used the wireless spectrum, we may reduce the available uplink bandwidths, eventually reducing the number of nodes that can be

handled. It is a future idea to explore how our schemes affect the bandwidth and whether we can further optimise them. In Ctrl-Mac, we took this one step further by regulating access to the channel for all nodes. As we have seen in the results, one single gateway could only handle up to 136 packets per minute, which is lower than what LoRaWAN handles typically. This is an intriguing latency, reliability and density tradeoff that we need to consider while designing the solutions for the applications.

6.1.2 Cellular and LoRa

Cellular communication like 5G and 6G is an obvious choice for WA-CPSs; they provide high reliabilities and widespread coverage. With the promise of ultra-reliable low latency communication, massive machine-to-machine communication, they can meet WA-CPS requirements. However, a few reasons make LoRa better suited for these applications compared to cellular communication. They are low power consumption, longer communication ranges, higher link budgets and ad-hoc deployments. High-bandwidth cellular technologies are not targetting the same use-cases as LPWANs, and they can be used at the gateways for the backhaul networks.

NB-IoT and LTE-M are also two standards within the cellular umbrella that target similar applications as LoRa and this thesis. A comparison between NB-IoT and LoRa was done in Chapter 2. Two benefits of LoRa compared to NB-IoT are their ad-hoc deployment model and cost. NB-IoT is expensive compared to LoRa and requires a subscription. Ad-hoc deployments allow users to deploy their private networks without relying on network providers. This facilitates rapid deployments and crowd-sourced networks, and a flexible network model which can be used for a wide-array of applications. LTE-M is designed for mobile applications, however it is significantly more expensive and power-hungry compared to LoRa, so it may not be well-suited for the mobile applications of Chapter 3. Cellular also implements means of handing over data between multiple operators to deal with areas of no or bad coverage. Chapter 4 proposes a similar data handover mechanism to bring together multi-network systems. Finally, Chapter 5 proposes applications where sensors and actuators are deployed in hard-to-reach places. LoRa has a higher penetration and resilience compared to NB-IoT and is naturally more suited for such applications. Also, the ad-hoc deployment models mean that application users can deploy their own gateways closer to where the sensors and actuators will be, increasing the chances for successful data delivery.

6.2 Future Developments

There are still several advancements that need to be made for LoRa-CPSs to achieve widespread usage. Some of the future developments are discussed herein.

Real-world adoption. To achieve real-world adoption, there are still several issues that need to be handled. One of the reasons for the large-scale adoption of LoRa and LoRaWAN is the standardisation of the protocol. For works proposed in the thesis to be used, they need to be part of the LoRaWAN standard. However, CPSs have stronger requirements, so the protocols need to be further analysed before the standardisation process. Further analysis of the protocols can be facilitated by benchmarking and rigorous experimentation. This is in line with the ideas proposed in [26, 27, 186], where the authors are creating benchmarks and benchmarking tools for short-range communication systems. We have also taken this approach and proposed SNIAPE[40], a benchmarking methodology for LPWANs. Benchmarking and standardisation would increase the trust of industries in the technology and facilitate commercial adoption.

Spectrum and duty-cycling usage. Several devices operate in the unlicensed spectrum. Applications operating using this spectrum are subject to duty-cycle regulations to ensure fair usage. However, as these are unregulated, there is no accountability of usage. We could leverage ideas from RMIP to study the node's arrival patterns and spot any duty-cycle violations. Additionally, one of the biggest bottlenecks of LoRaWAN for use in CPSs is their downlink channel on gateways. The downlink channel is subject to duty-cycle like all other nodes, and as seen in Chapter 5, the messages were reliably delivered, but the acknowledgements failed, triggering retransmissions. There have been works to do bulk acknowledgements or multi-channel acknowledgements which show good reliability. However, a straightforward regulatory change could be to increase the duty-cycle limits for gateway-class devices. A higher duty cycle would allow gateways to acknowledge more packets and regulate channel access better.

CPSs and machine learning. Learning-based schemes in wireless communications for CPSs can severely optimise the channels. The schemes can help design channel access schemes, choose transmission parameters, and decide when to transmit or for error recovery. All of these would increase the reliabilities of wireless communication allowing it to be used more in CPSs.

JIT vs one-shot. Much research in LoRaWAN is around static parameter allocation or slow-adapting ADR. Wireless environment, especially unlicensed spectrum, is dynamic. We need to learn macro-trends but make decisions just-in-time (JIT) for transmissions to have high performance. This JIT transmission parameter adaption is used in cellular communication, which helps it provide high reliability and handle large densities. One-shot parameter allocation or slow-adapting algorithms cannot provide the reliability levels needed for CPSs, so JIT schemes need to be explored further. RMIP and InterPred designed in Chapter 4 are both examples of JIT systems.

Gateway deployments. Network topology has a severe impact on the performance of wireless systems. We need to design gateway placement algorithms that can take into account the topography and environmental conditions. This technique is already used while deploying 5G networks; however, it is not quite common in the LoRaWAN world. Compared to cellular deployments, where the deployments are carrier-controlled, an ad-hoc deployment model of LoRaWAN could lead to a more concentrated deployment in one area, whereas no coverage in others. We could study the already existing deployments and decide on appropriate gateway placements. We could also blend this idea with monetisation for users to have a higher reward if gateways are deployed in areas with minimal coverage and lower reward otherwise. This could lead to an increase in coverage.

Finally, this thesis focussed on studying the limitations of LoRa, and addressing them to expand the use-cases of LoRa/LoRaWAN. Long-range, low-datarate communication schemes target several applications that the current wireless systems may not be in the position to meet. LPWANs and particularly LoRa are carving out a niche in the IoT space, and this thesis is trying to show how they can also do the same in the WA-CPS space. With the contributions of this thesis, along with the future work proposed above, LoRa can become one of the go-to solutions for any low-datarate, long-range application in the IoT and WA-CPS space.

Bibliography

[1] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. "Low power wide area networks: An overview". In: *IEEE Communications Surveys and Tutorials* 19.2 (2017), pp. 855–873.

[2] Lora Alliance Technical Committee. *LoRaWAN 1.1 Specification*. https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/. [Online; accessed 20-June-2021]. 2018.

[3] *Sigfox*. https://www.sigfox.com.

[4] *NB-IoT*. https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf.

[5] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. "Cyber-physical Systems: The Next Computing Revolution". In: *Proc. of the 47th Design Autom. Conf.* 2010, pp. 731–736.

[6] Edward Ashford Lee and Sanjit A Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. Mit Press, 2017.

[7] I. Lee et al. "Challenges and Research Directions in Medical Cyber–Physical Systems". In: *Proceedings of the IEEE* 100.1 (2012), pp. 75–90. DOI: 10.1109/JPROC.2011.2165270.

[8] Alessandra Bagnato, Regina Krisztina Biro, Dario Bonino, Claudio Pastrone, Wilfried Elmenreich, René Reiners, Melanie Schranz, and Edin Arnautovic. "Designing Swarms of Cyber-Physical Systems: The H2020 CPSwarm Project: Invited Paper". In: *Proceedings of the Computing Frontiers Conference*. CF'17. Siena, Italy: Association for Computing Machinery, 2017, pp. 305–312. ISBN:

9781450344876. DOI: 10.1145/3075564.3077628. URL: https://doi.org/10.1145/3075564.3077628.

[9] X. Yu and Y. Xue. "Smart Grids: A Cyber–Physical Systems Perspective". In: *Proceedings of the IEEE* 104.5 (2016), pp. 1058–1070. DOI: 10.1109/JPROC.2015.2503119.

[10] Naiqian Zhang, Maohua Wang, and Ning Wang. "Precision agriculture—a worldwide overview". In: *Computers and electronics in agriculture* 36.2-3 (2002), pp. 113–132.

[11] Walter Cinibulk. *Aircraft electrical wire: wire manufacturers perspective*. Tech. rep. TE Connectivity, 2010. URL: https://www.mitrecaasd.org/atsrac/FAA_PI-Engineer_Workshop/2001/aircraft_electrical_wire.pdf.

[12] Aranya Chakrabortty. "Wide-Area Control of Power Systems: Employing Data-Driven, Hierarchical Reinforcement Learning". In: *IEEE Electrification Magazine* 9.1 (2021), pp. 45–52. DOI: 10.1109/MELE.2020.3047165.

[13] Don Winter and Boeing Phantom Works. *Cyber physical systems-an aerospace industry perspective*. https://labs.ece.uw.edu/nsl/aar-cps/winterrev4.pdf.

[14] Palak P Parikh, Mitalkumar G Kanabar, and Tarlochan S Sidhu. "Opportunities and challenges of wireless communication technologies for smart grid applications". In: *IEEE PES General Meeting*. IEEE. 2010, pp. 1–7.

[15] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, and Li Zhao. "Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G". In: *IEEE Communications Standards Magazine* 1.2 (2017), pp. 70–76.

[16] Aline Baggio. "Wireless sensor networks in precision agriculture". In: *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden*. Vol. 20. Citeseer. 2005, pp. 1567–1576.

[17] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. "Farmbeats: An iot platform for data-driven agriculture". In: *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 2017, pp. 515–529.

[18] Pangun Park, Sinem Coleri Ergen, Carlo Fischione, Chenyang Lu, and Karl Henrik Johansson. "Wireless Network Design for Control Systems: A Survey". In: *IEEE Communications Surveys and Tutorials* 20.2 (2018), pp. 978–1013. ISSN: 1553877X. DOI: 10.1109/COMST.2017.2780114. arXiv: 1708.07353.

[19] Pangun Park, Piergiuseppe Di Marco, Junghyo Nah, and Carlo Fischione. *Wireless Avionics Intra-Communications: A Survey of Benefits, Challenges, and Solutions*. 2020. arXiv: 2006.12060 [cs.NI].

[20] O. Elgezabal Gómez. "Fly-by-wireless: Benefits, risks and technical challenges". In: *CANEUS Fly by Wireless Workshop 2010*. 2010, pp. 14–15. DOI: 10.1109/FBW.2010.5613788.

[21] Fifth Generation Communication Automotive Research and innovation. *5GCAR Scenarios, Use Cases,Requirements and KPIs*. Tech. rep. 5GPPP, 2017. URL: https://5gcar.eu/wp-content/uploads/2017/05/5GCAR_D2.1_v1.0.pdf.

[22] John B Kenney. "Dedicated short-range communications (DSRC) standards in the United States". In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182.

[23] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. "WirelessHART: Applying wireless technology in real-time industrial process control". In: *IEEE Real-Time and Embedded Tech. and Appl. Symp.* 2008.

[24] ISA standards committee member. *ISA100, Wireless Systems for Automation*. https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa100. [Online; accessed 15-December-2020]. 2020.

[25] Xavier Vilajosana, Cristina Cano, Borja Martinez, Pere Tuset-Peiró, and Ferran Adelantado. "The Wireless Technology Landscape in the Manufacturing Industry : A Reality Check". In: (2017), pp. 1–5.

[26] C. A. Boano et al. "IoTBench: Towards a Benchmark for Low-Power Wireless Networking". In: *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*. 2018, pp. 36–41. DOI: 10.1109/CPSBench.2018.00013.

[27] Romain Jacob, Carlo Alberto Boano, Usman Raza, Marco Zimmerling, and Lothar Thiele. "Towards a Methodology for Experimental Evaluation in Low-Power Wireless Networking". In: *Proceedings of the 2nd Workshop on Benchmarking Cyber-Physical Systems and Internet of Things*. CPS-IoTBench '19. Montreal, Quebec, Canada: Association for Computing Machinery, 2019, pp. 18–23. ISBN: 9781450366939. DOI: 10.1145/3312480.3313173. URL: https://doi.org/10.1145/3312480.3313173.

[28] Xavier Vilajosana, Thomas Watteyne, Mališa Vučinić, Tengfei Chang, and Kristofer SJ Pister. "6tisch: Industrial performance for ipv6 internet-of-things networks". In: *Proceedings of the IEEE* 107.6 (2019), pp. 1153–1165.

[29] Atis Elsts, Seohyang Kim, Hyung-Sin Kim, and Chongkwon Kim. "An Empirical Survey of Autonomous Scheduling Methods for TSCH". In: *IEEE Access* 8 (2020), pp. 67147–67165. DOI: 10.1109/ACCESS.2020.2980119.

[30] Venkata Prashant Modekurthy, Abusayeed Saifullah, and Sanjay Madria. "DistributedHART: A Distributed Real-Time Scheduling System for WirelessHART Networks". In: *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 2019, pp. 216–227. DOI: 10.1109/RTAS.2019.00026.

[31] Seohyang Kim, Hyung-Sin Kim, and Chongkwon Kim. "ALICE: Autonomous link-based cell scheduling for TSCH". In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. 2019, pp. 121–132.

[32] Kazushi Muraoka, Thomas Watteyne, Nicola Accettura, Xavier Vilajosana, and Kristofer S. J. Pister. "Simple Distributed Scheduling With Collision Detection in TSCH Networks". In: *IEEE Sensors Journal* 16.15 (2016), pp. 5848–5849. DOI: 10.1109/JSEN.2016.2572961.

[33] Shengyang Li, Usman Raza, and Aftab Khan. "How agile is the adaptive data rate mechanism of LoRaWAN?" In: *arXiv* (2018). ISSN: 23318422. arXiv: 1808.09286.

[34] H. Wang, G. Zhou, L. Bhatia, Z. Zhu, W. Li, and J. A. McCann. "Energy-Neutral and QoS-Aware Protocol in Wireless Sensor Networks for Health Monitoring of Hoisting Systems". In: *IEEE Transactions on Industrial Informatics* 16.8 (2020), pp. 5543–5553. DOI: 10.1109/TII.2020.2969218.

[35] I. Tomić, M. J. Breza, G. Jackson, L. Bhatia, and J. A. McCann. "Design and Evaluation of Jamming Resilient Cyber-Physical Systems". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 687–694. DOI: 10.1109/Cybermatics_2018.2018.00138.

[36] Laksh Bhatia, David Boyle, and Julie A. McCann. "Poster Abstract: Aerial Interactions with Wireless Sensors". In: *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. SenSys '18. Shenzhen, China: Association for Computing Machinery, 2018, pp. 373–374. ISBN: 9781450359528. DOI: 10.1145/3274783.3275189.

[37] Michael Breza, Laksh Bhatia, Ivana Tomic, Anqi Fu, Waqas Ikram, Valentinos Kongezos, and Julie A McCann. "The Separator, a Two-Phase Oil and Water Gravity CPS Separator Testbed". In: *arXiv preprint arXiv:2002.00945* (2020).

[38] Mohammad Heggo, Laksh Bhatia, and Julie A McCann. "RFTacho: An OAM approach for non-intrusive RF monitoring of rotating machines". In: *Proceedings of the 21th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2022)*. IPSN '22. New York, NY, USA: Association for Computing Machinery, 2022.

[39] Mohammad Heggo, Laksh Bhatia, and Julie A. McCann. "Poster Abstract: Cognisense: A Contactless Rotation Speed Measurement System". In: *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. SenSys '21. Coimbra, Portugal: Association for Computing Machinery, 2021, pp. 347–348. ISBN: 9781450390972. DOI: 10.1145/3485730.3492879. URL: https://doi.org/10.1145/3485730.3492879.

[40] Laksh Bhatia, Boon Wong, Yathreb Bouazizi, Michael Breza, and Julie A McCann. "SNIAPE: Sensor Network and IoT Application Performance Evaluation Benchmark". In: *(under review)* (2021).

[41] Laksh Bhatia, Ivana Tomić, Anqi Fu, Michael Breza, and Julie A. Mccann. "Control Communication Co-Design for Wide Area Cyber-Physical Systems". In: *ACM Transactions on Cyber-Physical Systems* 5.2 (Jan. 2021). ISSN: 2378-962X. DOI: 10.1145/3418528.

[42] P. -Y. Chen, L. Bhatia, R. Kolcun, D. Boyle, and J. A. McCann. "Contact-Aware Opportunistic Data Forwarding in Disconnected LoRaWAN Mobile Networks". In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. 2020, pp. 574–583. DOI: 10.1109/ICDCS47774.2020.00110.

[43] Laksh Bhatia, Po-Yu Chen, Michael Breza, Cong Zhao, and Julie A. McCann. "IRONWAN: Increasing Reliability of Overlapping Networks in LoRaWAN". In: *IEEE Internet of Things Journal* (2021), pp. 1–1. DOI: 10.1109/JIOT.2021.3125842.

[44] Ivana Tomić, Laksh Bhatia, Michael J Breza, and Julie A McCann. "The Limits of LoRaWAN in Event-Triggered Wireless Networked Control Systems". In: *2018 UKACC 12th International Conference on Control (CONTROL)* (2018), pp. 101–106.

[45] Laksh Bhatia, Michael Breza, Ramona Marfievici, and Julie A. McCann. "LoED: The LoRaWAN at the Edge Dataset: Dataset". In: *Proceedings of the Third Workshop on Data: Acquisition To Analysis*. DATA '20. Virtual Event, Japan: Association for Computing Machinery, 2020, pp. 7–8. ISBN: 9781450381369. DOI: 10.1145/3419016.3431491.

[46] Laksh Bhatia, Ivana Tomić, and Julie A. McCann. "Poster Abstract: LPWA-MAC: A Low Power Wide Area Network MAC Protocol for Cyber-Physical Systems". In: *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. SenSys '18. Shenzhen, China: Association for Computing Machinery, 2018, pp. 361–362. ISBN: 9781450359528. URL: https://doi.org/10.1145/3274783.3275183.

[47] Michael Baddeley. *Software Defined Networking for the Industrial Internet of Things*. 2020. URL: https://www.researchgate.net/profile/Michael%7B%5C_%7DBaddeley/publication/341680023%7B%5C_%7DSoftware%7B%5C_%7DDefined%7B%5C_%7DNetworking%7B%5C_%7Dfor%7B%5C_%7Dthe%7B%5C_%7DIndustrial%7B%5C_%7DInternet%7B%5C_%7Dof%7B%5C_%7DThings/links/5ecf753b92851c9c5e630530/Software-Defined-Networking-for-the-Industrial-Internet-of-Things.pdf.

[48] Kamil Staniec and Michał Kowal. "LoRa performance under variable interference and heavy-multipath conditions". In: *Wireless communications and mobile computing* 2018 (2018).

[49] *LoRa Alliance*. https://lora-alliance.org.

[50] *Weightless*. http://www.weightless.org. 2021.

[51] *Extended Coverage – GSM – Internet of Things (EC-GSM-IoT)*. https://www.gsma.com/iot/extended-coverage-gsm-internet-of-things-ec-gsm-iot/. 2021.

[52] "IEEE Standard for Low-Rate Wireless Networks - Amendment 5: Enabling/Updating the Use of Regional Sub-GHz Bands". In: *IEEE Std 802.15.4v-2017 (Amendment to IEEE Std 802.15.4-2015, as amended by IEEE Std 802.15.4n-2016, IEEE Std 802.15.4q-2016, IEEE Std 802.15.4u-2016, and IEEE Std 802.15.4t-2017)* (June 2017), pp. 1–35. DOI: 10.1109/IEEESTD.2017.7964803.

[53] *Dash7*. http://www.dash7-alliance.org. 2021.

[54] "IEEE Standard for Information technology–Telecommunications and information exchange between systems - Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation". In: *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)* (2017), pp. 1–594. DOI: 10.1109/IEEESTD.2017.7920364.

[55] Augustine Ikpehai, Bamidele Adebisi, Khaled M. Rabie, Kelvin Anoh, Ruth E. Ande, Mohammad Hammoudeh, Haris Gacanin, and Uche M. Mbanaso. "Low-power wide area network technologies for internet-of-things: A comparative review". In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 2225–2240. ISSN: 23274662. DOI: 10.1109/JIOT.2018.2883728.

[56] Joseph Finnegan and Stephen Brown. *A Comparative Survey of LPWA Networking*. 2018. eprint: arXiv:1802.04222.

[57] Sokratis Kartakis, Babu D. Choudhary, Alexander D. Gluhak, Lambros Lambrinos, and Julie A. McCann. "Demystifying Low-Power Wide-Area Communications for City IoT Applications". In: *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*. WiNTECH '16. New York City, New York: Association for Computing Machinery, 2016, pp. 2–8. ISBN: 9781450342520. DOI: 10.1145/2980159.2980162. URL: https://doi.org/10.1145/2980159.2980162.

[58] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. "A comparative study of LPWAN technologies for large-scale IoT deployment". In: *ICT express* 5.1 (2019), pp. 1–7.

[59] Mikhail Afanasov et al. "Battery-Less Zero-Maintenance Embedded Sensing at the MithræUm of Circus Maximus". In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 368–381. ISBN: 9781450375900. URL: https://doi.org/10.1145/3384419.3430722.

[60] Semtech. *SX1301 Datasheet*. https://www.semtech.com/products/wireless-rf/lora-gateways/sx1301.

[61] Marco Chiani and Ahmed Elzanaty. "On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8463–8470. DOI: 10.1109/JIOT.2019.2919151.

[62] R. W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[63] *Short Range Devices operating in the 863 - 870 MHz frequency band*. https://www.ofcom.org.uk/__data/assets/pdf_file/0025/38095/final_report.pdf. 2010.

[64] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo. "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology". In: *2015 14th International Conference on ITS Telecommunications (ITST)*. 2015, pp. 55–59. DOI: 10.1109/ITST.2015.7377400.

[65] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. "Do LoRa low-power wide-area networks scale?" In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. November. 2016, pp. 59–67. ISBN: 9781450345026. DOI: 10.1145/2988287.2989163. URL: http://dl.acm.org/citation.cfm?doid=2988287.2989163.

[66] Orestis Georgiou and Usman Raza. "Low power wide area network analysis: Can LoRa scale?" In: *IEEE Wireless Communications Letters* 6.2 (2017), pp. 162–165.

[67] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke. "LoRa scalability: A simulation model based on interference measurements". In: *Sensors* 17.6 (2017), p. 1193. ISSN: 14248220. DOI: 10.3390/s17061193.

[68] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. "Mitigating inter-network interference in LoRa networks". In: *arXiv preprint arXiv:1611.00688*. EWSN '17 (2016), pp. 323–328.

[69] Mariusz Slabicki, Gopika Premsankar, Mario Di Francesco, and Mario Di Francesco. "Adaptive Configuration of LoRa Networks for Dense IoT Deployments". In: *16th IEEE/IFIP Network Operations and Management Symposium (NOMS 2018)*. 2018, pp. 1–9. URL: http://flora.aalto.fi/download/slabicki2018adaptive.pdf.

[70] P. Jörke, S. Böcker, F. Liedmann, and C. Wietfeld. "Urban channel models for smart city IoT-networks based on empirical measurements of LoRa-links at 433 and 868 MHz". In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–6. DOI: 10.1109/PIMRC.2017.8292708.

[71] Hendrik Linka, Michael Rademacher, Osianoh Glenn Aliu, and Karl Jonas. "Path loss models for low-power wide-area networks: Experimental results using lora". In: (2018).

[72] W. Xu, J. Y. Kim, W. Huang, S. S. Kanhere, S. K. Jha, and W. Hu. "Measurement, Characterization, and Modeling of LoRa Technology in Multifloor Buildings". In: *IEEE Internet of Things Journal* 7.1 (2020), pp. 298–310. DOI: 10.1109/JIOT.2019.2946900.

[73] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. "Empowering Low-Power Wide Area Networks in Urban Settings". In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '17. Los Angeles, CA, USA: Association for Computing Machinery, 2017, pp. 309–321. ISBN: 9781450346535. DOI: 10.1145/3098822.3098845. URL: https://doi.org/10.1145/3098822.3098845.

[74] Zhenqiang Xu, Shuai Tong, Pengjin Xie, and Jiliang Wang. "FlipLoRa: Resolving Collisions with Up-Down Quasi-Orthogonality". In: *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 2020, pp. 1–9. DOI: 10.1109/SECON48991.2020.9158432.

[75] Xianjin Xia, Yuanqing Zheng, and Tao Gu. "FTrack: Parallel Decoding for LoRa Transmissions". In: *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. SenSys '19. New York, New York: Association for Computing Machinery, 2019. ISBN: 9781450369503. DOI: 10.1145/3356250.3360024. URL: https://doi.org/10.1145/3356250.3360024.

[76] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. "MLoRa: A multi-packet reception protocol in LoRa networks". In: *Proceedings - International Conference on Network Protocols, ICNP* 2019-Octob (2019). ISSN: 10921648. DOI: 10.1109/ICNP.2019.8888038.

[77] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. "CoLoRa: Enabling Multi-Packet Reception in LoRa". In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 2020, pp. 2303–2311. DOI: 10.1109/INFOCOM41043.2020.9155509.

[78] Vinay Gupta, Sendil Kumar Devar, N. Hari Kumar, and Kala Praveen Bagadi. "Modelling of IoT Traffic and Its Impact on LoRaWAN". In: *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings* 2018-January (2017), pp. 1–6. DOI: 10.1109/GLOCOM.2017.8254512.

[79] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. "Mathematical model of LoRaWAN channel access with capture effect". In: *arXiv* 14 (2020). ISSN: 23318422.

[80] Yathreb Bouazizi, Fatma Benkhelifa, and Julie McCann. "Spatiotemporal Modelling of Multi-Gateway LoRa Networks with Imperfect SF Orthogonality". In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 2020, pp. 1–7. DOI: 10.1109/GLOBECOM42002.2020.9322640.

[81] Martin Heusse and Sophia Antipolis. "Capacity of a LoRaWAN Cell". In: *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (Nov. 2020), pp. 131–140. DOI: 10.1145/3416010.3423228. URL: https://dl.acm.org/doi/10.1145/3416010.3423228.

[82] Verónica Toro-Betancur, Gopika Premsankar, Mariusz Slabicki, and Mario Di Francesco. "Modeling Communication Reliability in LoRa Networks with Device-level Accuracy". In: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*. 2021, pp. 1–10. DOI: 10.1109/INFOCOM42981.2021.9488783.

[83] Alexandru-Ioan Ioan Pop, Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?" In: *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings* 2018-Janua (Dec. 2018), pp. 1–6. DOI: 10.1109/GLOCOM.2017.8254509. arXiv: 1704.04174. URL: http://ieeexplore.ieee.org/document/8254509/.

[84] M. Capuzzo, D. Magrin, and A. Zanella. "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures". In: *Med-Hoc-Net*. June 2018, pp. 1–7. DOI: 10.23919/MedHocNet.2018.8407095.

[85] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. "Charm: Exploiting Geographical Diversity through Coherent Combining in Low-Power Wide-Area Networks". In: *Proceedings - 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2018* (2018), pp. 60–71. DOI: 10.1109/IPSN.2018.00013.

[86] Jun Liu, Weitao Xu, Sanjay Jha, and Wen Hu. "Nephalai: Towards LPWAN C-RAN with physical layer compression". In: *arXiv* (2020). ISSN: 23318422. DOI: 10.1145/3372224.3419193. arXiv: 2008.02599.

[87] Artur Balanuta, Nuno Pereira, Swarun Kumar, and Anthony Rowe. "A Cloud-Optimized Link Layer for Low-Power Wide-Area Networks". In: *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. MobiSys '20. Toronto, Ontario, Canada: Association for Computing Machinery, 2020. ISBN: 9781450379540. DOI: 10.1145/3386901.3388915. URL: https://doi.org/10.1145/3386901.3388915.

[88] Stéphane Delbruel, Nicolas Small, Emekcan Aras, Jonathan Oostvogels, and Danny Hughes. "Tackling Contention Through Cooperation: A Distributed Federation in LoRaWAN Space". In: *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks on Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*. EWSN '20. Lyon, France: Junction Publishing, 2020, pp. 13–24. ISBN: 9780994988645.

[89] Jeferson Rodrigues Cotrim and João Henrique Kleinschmidt. "LoRaWAN Mesh networks: A review and classification of multihop communication". In: *Sensors (Switzerland)* 20.15 (2020), pp. 1–21. ISSN: 14248220. DOI: 10.3390/s20154273.

[90] Muhammad Shehryar Aslam, Alishba Khan, Abeera Atif, Syed Ali Hassan, Aamir Mahmood, Hassaan Khaliq Qureshi, and Mikael Gidlund. "Exploring Multi-Hop LoRa for Green Smart Cities". In: *IEEE Network* 34.2 (2020), pp. 225–231. ISSN: 1558156X. DOI: 10.1109/MNET.001.1900269.

[91] Martin Haubro, Charalampos Orfanidis, George Oikonomou, and Xenofon Fafoutis. "TSCH-over-LoRA : long range and reliable IPv6 multi-hop networks for the internet of things". In: *Internet Technology Letters* 3.4 (2020), pp. 1–6. ISSN: 2476-1508. DOI: 10.1002/itl2.165.

[92] *Contiki-NG*. http://contiki-ng.org. 2021.

[93] Dion Tanjung, Seunggyu Byeon, Dong Hyun Kim, and Jong Deok Kim. "OODC: An Opportunistic and On-Demand Forwarding Mechanism for LPWA Networks". In: *International Conference on Information Networking* 2020-January (2020), pp. 301–306. ISSN: 19767684. DOI: 10.1109/ICOIN48656.2020.9016475.

[94] Toni Adame Vázquez. "HARE: Supporting Efficient Uplink Multi-Hop Communications in Self-Organizing LPWANs". In: *Sensors* 18.1 (2018), p. 115. ISSN: 1424-8220. DOI: 10.3390/s18010115. URL: http://www.mdpi.com/1424-8220/18/1/115.

[95] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks". In: *Proceedings of the 4th international conference on Embedded networked sensor systems*. 2006, pp. 307–320.

[96] Made Harta Dwijaksara, Wha Sook Jeon, and Dong Geun Jeong. "Multihop gateway-to-gateway communication protocol for lora networks". In: *Proceedings of the IEEE International Conference on Industrial Technology* 2019-Febru (2019), pp. 949–954. DOI: 10.1109/ICIT.2019.8755136.

[97] Konstantinos Kousias, Giuseppe Caso, Özgü Alay, and Filip Lemic. "Empirical analysis of LoRaWAN adaptive data rate for mobile internet of things applications". In: *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*. S3'19 (2019), pp. 9–11. DOI: 10.1145/3349621.3355727. URL: https://doi.org/10.1145/3349621.3355727.

[98] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti. "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage". In: *International Journal of Distributed Sensor Networks* 13.3 (2017), p. 1550147717699412.

[99] P. J. Marcelis, V. S. Rao, and R. V. Prasad. "DaRe: Data Recovery through Application Layer Coding for LoRaWAN". In: *Proceedings IoTDI 2017*. 2017, pp. 97–108.

[100] Yosra Zguira, Hervé Rivano, and Aref Meddeb. "Internet of bikes: A DTN protocol with data aggregation for urban data collection". In: *Sensors (Switzerland)* 18.9 (Aug. 2018), p. 2819. ISSN: 14248220. DOI: 10.3390/s18092819. URL: http://www.mdpi.com/1424-8220/18/9/2819.

[101] Arshad Farhad, Dae Ho Kim, Santosh Subedi, and Jae Young Pyun. "Enhanced LoRaWAN adaptive data rate for mobile internet of things devices". In: *Sensors (Switzerland)* 20.22 (2020), pp. 1–21. ISSN: 14248220. DOI: 10.3390/s20226466.

[102] Arshad Farhad, Dae-Ho Kim, Beom-Hun Kim, Alaelddin Fuad Yousif Mohammed, and Jae-Young Pyun. "Mobility-Aware Resource Assignment to IoT Applications in Long-Range Wide Area Networks". In: *IEEE Access* 8 (2020), pp. 186111–186124. ISSN: 2169-3536. DOI: 10.1109/access.2020.3029575.

[103] Branden Ghena, Joshua Adkins, Longfei Shangguan, Kyle Jamieson, Philip Levis, and Prabal Dutta. "Challenge: Unlicensed LPWANs Are Not Yet the Path to Ubiquitous Connectivity". In: *The 25th Annual International Conference on Mobile Computing and Networking*. 2019, pp. 1–12.

[104] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. "Understanding the Limits of LoRaWAN". In: *IEEE Communications Magazine* 55.9 (Sept. 2017), pp. 34–40. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600613.

[105] S. Yang, U. Adeel, Y. Tahir, and J. A. McCann. "Practical opportunistic data collection in wireless sensor networks with mobile sinks". In: *IEEE Transactions on Mobile Computing* 16.5 (2017), pp. 1420–1433.

[106] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. "A high-throughput path metric for multi-hop wireless routing". In: *Wireless networks* 11.4 (2005), pp. 419–434.

[107] A. Dunkels, B. Gronvall, and T. Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors". In: *Proc. IEEE LCN 2004.* 2004, pp. 455–462.

[108] Y. Tahir, S. Yang, and J. McCann. "BRPL: Backpressure RPL for High-throughput and Mobile IoTs". In: *IEEE Transactions on Mobile Computing* (2017).

[109] Leandros Tassiulas and Anthony Ephremides. "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks". In: *29th IEEE Conference on Decision and Control.* IEEE. 1990, pp. 2130–2132.

[110] Scott Moeller, Avinash Sridharan, Bhaskar Krishnamachari, and Omprakash Gnawali. "Routing without routes: The backpressure collection protocol". In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks.* ACM. 2010, pp. 279–290.

[111] Michael J. Neely and Rahul Urgaonkar. "Optimal backpressure routing for wireless networks with multi-receiver diversity". In: *Ad Hoc Networks* 7.5 (2009), pp. 862–881. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2008.07.009. URL: https://www.sciencedirect.com/science/article/pii/S1570870508001157.

[112] S Malathy, V Porkodi, A Sampathkumar, MHD Hindia, Kaharudin Dimyati, Valmik Tilwari, Faizan Qamar, and Iraj Sadegh Amiri. "An optimal network coding based backpressure routing approach for massive IoT network". In: *Wireless Networks* 26.5 (2020), pp. 3657–3674.

[113] TFL. *Transport for London Open Data.* tfl.gov.uk/info-for/open-data-users/. 2019.

[114] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wiessner. "Microscopic Traffic Simulation using SUMO". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC).* 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

[115] Christoph Sommer. *Veins, the open source vehicular simulation framework.* https://veins.car2x.org. [Online; accessed 28-May-2022]. 2022.

[116] Norbert Blenn and Fernando Kuipers. *LoRaWAN in the Wild: Measurements from The Things Network*. 2017. arXiv: 1706.03086 [cs.NI].

[117] Y. Cao and Z. Sun. "Routing in Delay/Disruption Tolerant Networks: A Taxonomy, Survey and Challenges". In: *IEEE Communications Surveys and Tutorials* 15.2 (2013), pp. 654–677.

[118] M. Conti and S. Giordano. "Mobile ad hoc networking: milestones, challenges, and new research directions". In: *IEEE Communications Magazine* 52.1 (2014), pp. 85–96.

[119] B. Kusy, H. Lee, M. Wicke, N. Milosavljevic, and L. Guibas. "Predictive QoS routing to mobile sinks in wireless sensor networks". In: *Proc. Int. Conf. Inf. Process. Sensor Netw.* 2009, pp. 109–120.

[120] Z. Li, M. Li, J. Wang, and Z. Cao. "Ubiquitous data collection for mobile users in wireless sensor networks". In: *2011 Proceedings IEEE INFOCOM*. 2011, pp. 2246–2254. DOI: 10.1109/INFCOM.2011.5935040.

[121] Jung Woo Lee, Branislav Kusy, Tahir Azim, Basem Shihada, and Philip Levis. "Whirlpool Routing for Mobility". In: *Proceedings of the Eleventh ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '10. Chicago, Illinois, USA: Association for Computing Machinery, 2010, pp. 131–140. ISBN: 9781450301831. DOI: 10.1145/1860093.1860112. URL: https://doi.org/10.1145/1860093.1860112.

[122] Amin Vahdat, David Becker, et al. *Epidemic routing for partially connected ad hoc networks*. 2000.

[123] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. "Efficient routing in intermittently connected mobile networks: The multiple-copy case". In: *IEEE/ACM transactions on networking* 16.1 (2008), pp. 77–90.

[124] Anders Lindgren, Avri Doria, and Olov Schelen. "Probabilistic routing in intermittently connected networks". In: *International Workshop on Service Assurance with Partial and Intermittent Resources*. Springer. 2004, pp. 239–254.

[125] L. Pelusi, A. Passarella, and M. Conti. "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks". In: *IEEE Communications Magazine* 44.11 (2006), pp. 134–141.

[126] P. Yuan, H. Ma, and H. Fu. "Hotspot-entropy based data forwarding in opportunistic social networks". In: *Pervasive and Mobile Computing* 16 (2015), pp. 136–154.

[127] S. M. Tornell, C. T. Calafate, J. Cano, and P. Manzoni. "DTN Protocols for Vehicular Networks: An Application Oriented Overview". In: *IEEE Communications Surveys and Tutorials* 17.2 (2015), pp. 868–887.

[128] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and X. Sherman Shen. "ZOOM: Scaling the mobility for fast opportunistic forwarding in vehicular networks". In: *Proceedings of IEEE INFOCOM 2013*. 2013, pp. 2832–2840.

[129] K. Zhu, W. Li, and X. Fu. "SMART: A Social- and Mobile-Aware Routing Strategy for Disruption-Tolerant Networks". In: *IEEE Transactions on Vehicular Technology* 63.7 (2014), pp. 3423–3434.

[130] H. Zhou, V. C. M. Leung, C. Zhu, S. Xu, and J. Fan. "Predicting Temporal Social Contact Patterns for Data Forwarding in Opportunistic Mobile Networks". In: *IEEE Transactions on Vehicular Technology* (2017).

[131] H. Zhou, J. Chen, H. Zhao, W. Gao, and P. Cheng. "On Exploiting Contact Patterns for Data Forwarding in Duty-Cycle Opportunistic Mobile Networks". In: *IEEE Transactions on Vehicular Technology* 62.9 (Nov. 2013), pp. 4629–4642.

[132] Ramona Marfievici, Amy L Murphy, Gian Pietro Picco, Federico Ossi, and Francesca Cagnacci. "How environmental factors impact outdoor wireless sensor networks: a case study". In: *2013 IEEE 10th international conference on mobile ad-hoc and sensor systems*. IEEE. 2013, pp. 565–573.

[133] Vasileios Kotsiou, Georgios Z Papadopoulos, Periklis Chatzimisios, and Fabrice Tholeyre. "Is local blacklisting relevant in slow channel hopping low-power wireless networks?" In: *2017 IEEE International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6.

[134] Pedro Henrique Gomes, Thomas Watteyne, and Bhaskar Krishnamachari. "MABO-TSCH: multihop and blacklist-based optimized time synchronized channel hopping". In: *Transactions on Emerging Telecommunications Technologies* 29.7 (2018), e3223.

[135] V. Di Vincenzo, M. Heusse, and B. Tourancheau. "Improving Downlink Scalability in LoRaWAN". In: *IEEE ICC 2019*. 2019.

[136] Michiel Aernouts, Rafael Berkvens, Koen Van Vlaenderen, and Maarten Weyn. *Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas*. July 2019. DOI: 10.5281/zenodo.3342253.

[137] Rachel Cardell-Oliver, Christof Hübner, Matthias Leopold, and Jason Beringer. "Dataset: LoRa Underground Farm Sensor Network". In: *Proc. of the 2nd Workshop on Data Acquisition To Analysis (DATA)*. 2019.

[138] Laksh Bhatia, Michael Breza, Ramona Marfievici, and Julie A. McCann. *LoED: The LoRaWAN at the Edge Dataset*. Sept. 2020. DOI: 10.5281/zenodo.4121430.

[139] Orion Afisiadis, Student Member, Matthieu Cotting, Andreas Burg, and Alexios Balatsoukas-stimming. "On the Error Rate of the LoRa Modulation with Interference". In: (2019), pp. 1–11. arXiv: arXiv:1905.11252v1.

[140] Kwon Nung Choi, Harini Kolamunna, Akila Uyanwatta, Kanchana Thilakarathna, Suranga Seneviratne, Ralph Holz, Mahbub Hassan, and Albert Y Zomaya. "LoRadar: LoRa Sensor Network Monitoring through Passive Packet Sniffing". In: *SIGCOMM Computer Communication Review* 50.4 (2020), pp. 10–24. ISSN: 0146-4833. URL: https://doi.org/10.1145/3431832.3431835.

[141] Joost CF De Winter. "Using the Student's t-test with extremely small sample sizes". In: *Practical Assessment, Research, and Evaluation* (2013).

[142] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[143] The Things Network. *PacketBroker*. https://www.packetbroker.org. [Online; accessed 20-June-2021]. 2019.

[144] M. Slabicki, G. Premsankar, and M. Di Francesco. "Adaptive configuration of lora networks for dense IoT deployments". In: *IEEE/IFIP NOMS*. 2018.

[145] Olivier Seller. *Predicting LoRaWAN Capacity*. https://lora-developers.semtech.com/library/tech-papers-and-guides/predicting-lorawan-capacity. [Online; accessed 20-June-2021]. 2020.

[146] Kathleen McLaughlin. *Gaps in 4G Network Hinder High-tech Agriculture: FCC Prepares to Release 500 Million to Improve Coverage*. 2016.

[147] Shangxing Wang, Hanpeng Liu, Pedro Henrique Gomes, and Bhaskar Krishnamachari. "Deep reinforcement learning for dynamic multichannel access in wireless networks". In: *IEEE Transactions on Cognitive Communications and Networking* 4.2 (2018), pp. 257–265.

[148] Ruben M Sandoval, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. "Optimizing and updating lora communication parameters: A machine learning approach". In: *IEEE Transactions on Network and Service Management* 16.3 (2019), pp. 884–895.

[149] S. Kartakis, A. Fu, M. Mazo, and J. A. McCann. "Communication Schemes for Centralized and Decentralized Event-Triggered Control Systems". In: *IEEE Transactions on Control Systems Technology* 26.6 (2018), pp. 2035–2048.

[150] Belimo. *NRQ24A-SR Datasheet*. https://www.belimo.ch/pdf/e/NRQ24A-SR_datasheet_en-gb.pdf. [Online; accessed 20-June-2021]. 2019.

[151] James Dicken, Paul D Mitcheson, Ivan Stoianov, and Eric M Yeatman. "Power-extraction circuits for piezoelectric energy harvesters in miniature and low-power applications". In: *IEEE Transactions on Power Electronics* 27.11 (2012), pp. 4514–4529.

[152] Chenyang Lu, Abusayeed Saifullah, Bo Li, Mo Sha, Humberto Gonzalez, Dolvara Gunatilaka, Chengjie Wu, Lanshun Nie, and Yixin Chen. "Real-time wireless sensor-actuator networks for industrial cyber-physical systems". In: *Proceedings of the IEEE* 104.5 (2016), pp. 1013–1024.

[153] WPM Heemels Heemels, MCF Donkers, and Andrew R Teel. "Periodic event-triggered control for linear systems". In: *IEEE Transactions on Automatic Control* 58.4 (2013), pp. 847–861.

[154] Dominicus Paulus Borgers, Victor Sebastiaan Dolk, and WPMH Heemels. "Riccati-based design of event-triggered controllers for linear systems with delays". In: *IEEE Transactions on Automatic Control* 63.1 (2018), pp. 174–188.

[155] Balakrishnan Chandrasekaran. *Survey of Network Traffic Models*. https://www.cse.wustl.edu/~jain/cse567-06/ftp/traffic_models3.pdf. [Online; accessed 20-June-2021]. 2006.

[156] Thulo Ram Gurung, Rodney A Stewart, Cara D Beal, and Ashok K Sharma. "Smart meter enabled water end-use demand data: platform for the enhanced infrastructure planning of contemporary urban water supply networks". In: *Journal of Cleaner Production* 87 (2015), pp. 642–654.

[157] Alain Moreau. "Control strategy for domestic water heaters during peak periods and its impact on the demand for electricity". In: *Energy Procedia* 12 (2011), pp. 1074–1082.

[158] Sokratis Kartakis, Weiren Yu, Reza Akhavan, and Julie A McCann. "Adaptive edge analytics for distributed networked control of water systems". In: *2016 IEEE First Int. Conf. on Internet-of-Things Design and Impl.* IEEE. IEEE, 2016, pp. 72–82.

[159] Weiren Yu and Julie A. McCann. "Effectively Positioning Water Loss Event in Smart Water Networks". In: *The 2nd International Electronic Conference on Sensors and Applications (IECSA)* (2015), pp. 1–10.

[160] Aranya Chakrabortty and Pramod P. Khargonekar. "Introduction to wide-area control of power systems". In: *2013 American Control Conference*. 2013, pp. 6758–6770. DOI: 10.1109/ACC.2013.6580901.

[161] Seyed Mehran Dibaji, Anuradha Annaswamy, Aranya Chakrabortty, and Alefiya Hussain. "Sparse and Distributed Control of Wide-Area Power Systems with Large Communication Delays". In: *Proceedings of the American Control Conference* 2018-June (2018), pp. 3822–3827. ISSN: 07431619. DOI: 10.23919/ACC.2018.8431309.

[162] Seyed Mehran Dibaji et al. "Delay-Aware Wide-Area Control of Power Systems over Sparse Communications with Analytical Guarantees". In: *American Control Conference (ACC)*. Institute of Electrical and Electronics Engineers (IEEE). 2018.

[163] Xian-Ming Zhang, Qing-Long Han, Xiaohua Ge, Derui Ding, Lei Ding, Dong Yue, and Chen Peng. "Networked control systems: a survey of trends and techniques". In: *IEEE/CAA Journal of Automatica Sinica* 7.1 (2020), pp. 1–17. DOI: 10.1109/JAS.2019.1911651.

[164] Manuel Mazo and Paulo Tabuada. "Decentralized event-triggered control over wireless sensor/actuator networks". In: *IEEE Transactions on Automatic Control* 56.10 (2011), pp. 2456–2461.

[165] Manuel Mazo Jr and Anqi Fu. "Decentralized event-triggered controller implementations". In: *Event-Based Control and Signal Processing* (2015), p. 121.

[166] Xiaofeng Wang and Michael D Lemmon. "Event-triggering in distributed networked control systems". In: *IEEE Transactions on Automatic Control* 56.3 (2011), p. 586.

[167] Xiaohua Ge, Qing-Long Han, Xian-Ming Zhang, Lei Ding, and Fuwen Yang. "Distributed Event-Triggered Estimation Over Sensor Networks: A Survey". In: *IEEE Transactions on Cybernetics* 50.3 (2020), pp. 1306–1320. DOI: 10.1109/TCYB.2019.2917179.

[168] Andrzej Pawlowski, Jose Luis Guzman, Francisco Rodriguez, Manuel Berenguel, José Sánchez, and Sebastián Dormido. "Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control". In: *Sensors* 9.1 (2009), pp. 232–252.

[169] Malik Tubaishat, Yi Shang, and Hongchi Shi. "Adaptive traffic light control with wireless sensor networks". In: *Consumer Commun. and Networking Conf., 2007. CCNC 2007. 4th IEEE*. IEEE. 2007, pp. 187–191.

[170] Anna Pellegrino, Valerio RM Lo Verso, Laura Blaso, Andrea Acquaviva, Edoardo Patti, and Anna Osello. "Lighting control and monitoring for energy efficiency: a case study focused on the interoperability of building management systems". In: *IEEE Transactions on Industry Applications* 52.3 (2016), pp. 2627–2637.

[171] Stig Petersen and Simon Carlsen. "WirelessHART versus ISA100. 11a: The format war hits the factory floor". In: *IEEE Industrial Electronics Magazine* 5.4 (2011), pp. 23–34.

[172] Craig B Schindler, Thomas Watteyne, Xavier Vilajosana, and Kristofer SJ Pister. "Implementation and characterization of a multi-hop 6TiSCH network for experimental feedback control of an inverted pendulum". In: *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2017 15th Int. Symp. on*. IEEE. 2017, pp. 1–8.

[173] M. Ceriotti et al. "Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels". In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. Apr. 2011, pp. 187–198.

[174] José Araújo, Manuel Mazo, Adolfo Anta, Paulo Tabuada, and Karl H Johansson. "System architectures, protocols and algorithms for aperiodic wireless control systems". In: *IEEE Transactions on Industrial Informatics* 10.1 (2014), pp. 175–184.

[175] A. Saifullah, S. Sankar, J. Liu, C. Lu, R. Chandra, and B. Priyantha. "CapNet: A Real-Time Wireless Management Network for Data Center Power Capping". In: *2014 IEEE Real-Time Systems Symposium*. Dec. 2014, pp. 334–345. DOI: 10.1109/RTSS.2014.35.

[176] G. C. Walsh and L. G. Bushnell. "Real-time mixed-traffic wireless networks". In: *IEEE Transactions on Industrial Electronics* 48.5 (Oct. 2001), pp. 883–890. ISSN: 0278-0046. DOI: 10.1109/41.954551.

[177] Fabian Mager, Dominik Baumann, Romain Jacob, Lothar Thiele, Sebastian Trimpe, and Marco Zimmerling. "Feedback Control Goes Wireless: Guaranteed Stability over Low-power Multi-hop Networks". In: *Proc. of the 10th ACM/IEEE Int. Conf. on Cyber-Physical Sys.* ICCPS '19. Montreal, Quebec, Canada: ACM, 2019, pp. 97–108. ISBN: 978-1-4503-6285-6.

[178] Y. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka. "RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications". In: *2013 IEEE 34th Real-Time Systems Symposium*. Dec. 2013, pp. 140–149. DOI: 10.1109/RTSS.2013.22.

[179] N. W. Bauer, S. J. L. M. B. van Loon, N. van de Wouw, and W. P. M. H. M. Heemels. "Exploring the Boundaries of Robust Stability Under Uncertain Communication: An NCS Toolbox Applied to a Wireless Control Setup". In: *IEEE Control Systems Magazine* 34.4 (Aug. 2014), pp. 65–86. ISSN: 1066-033X. DOI: 10.1109/MCS.2014.2320394.

[180] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne. "Closed-loop control over wireless networks". In: *IEEE Control Systems Magazine* 24.3 (June 2004), pp. 58–71. ISSN: 1066-033X. DOI: 10.1109/MCS.2004.1299533.

[181] Matteo Trobinger, Gabriel de Albuquerque Gleizer, Timofei Istomin, Manuel Mazo, Amy L. Murphy, and Gian Pietro Picco. "The Wireless Control Bus: Enabling Efficient Multi-Hop Event-Triggered Control with Concurrent Transmissions". In: *ACM Transactions on Cyber-Physical Systems* 6.1 (Nov. 2021). ISSN: 2378-962X. DOI: 10.1145/3485467. URL: https://doi.org/10.1145/3485467.

[182] Marco Zimmerling, Luca Mottola, Pratyush Kumar, Federico Ferrari, and Lothar Thiele. "Adaptive Real-Time Communication for Wireless Cyber-Physical Systems". In: *ACM Transactions on Cyber-Physical Systems* 1.2 (Feb. 2017), 8:1–8:29. ISSN: 2378-962X. DOI: 10.1145/3012005. URL: http://doi.acm.org/10.1145/3012005.

[183] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. "Efficient network flooding and time synchronization with Glossy". In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. Apr. 2011, pp. 73–84.

[184] Maice Costa, Marian Codreanu, and Anthony Ephremides. "On the Age of Information in Status Update Systems With Packet Management". In: *IEEE Transactions on Information Theory* 62.4 (2016), pp. 1897–1910. DOI: 10.1109/TIT.2016.2533395.

[185] Michael Koziol. *Swarm Takes LoRa Sky-High*. https://spectrum.ieee.org/tech-talk/telecom/wireless/swarm-takes-lora-skyhigh. 2021.

[186] Romain Jacob. "Leveraging Synchronous Transmissions for the Design of Real-time Wireless Cyber-Physical Systems". en. PhD thesis. Zurich: ETH Zurich, 2020. DOI: 10.3929/ethz-b-000401717.

[187] Danda B Rawat, Chandra Bajracharya, and Gongjun Yan. "Towards intelligent transportation cyber-physical systems: Real-time computing and communications perspectives". In: *SoutheastCon 2015*. IEEE. 2015, pp. 1–6.

[188] Junping Zhang, Fei Yue Wang, Kunfeng Wang, Wei Hua Lin, Xin Xu, and Cheng Chen. "Data-driven intelligent transportation systems: A survey". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (2011), pp. 1624–1639. ISSN: 15249050. DOI: 10.1109/TITS.2011.2158001.

[189] *Critical National Infrastructure*. https://www.cpni.gov.uk/critical-national-infrastructure-0. 2021.

[190] *Communications requirements of smart grid technologies*. Tech. rep. Department of Energy, 2010. URL: https://www.energy.gov/sites/default/files/gcprod/documents/Smart_Grid_Communications_Requirements_Report_10-05-2010.pdf.

[191] Sokratis Kartakis. “Next generation cyber-physical water distribution systems”. PhD thesis. 2017.

[192] Panagiotis Tsakalides, Athanasia Panousopoulou, Grigorios Tsagkatakis, and Luis Montestruque. *Smart Water Grids: A Cyber-Physical Systems Approach*. CRC Press, 2018.

[193] *Water Facts*. http://www.waterintelligence.co.uk/water-facts/. 2011.

[194] Jack Loughran. *Water leakage from UK pipes rises to over three billion litres a day*. https://eandt.theiet.org/content/articles/2017/12/water-leakage-from-uk-pipes-rises-to-over-three-billion-litres-a-day/. Dec. 2017.

[195] Eduardo D. Sontag. “Input to State Stability: Basic Concepts and Results”. In: *Nonlinear and Optimal Control Theory: Lectures given at the C.I.M.E. Summer School held in Cetraro, Italy June 19–29, 2004*. Ed. by Paolo Nistri and Gianna Stefani. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 163–220. ISBN: 978-3-540-77653-6. DOI: 10.1007/978-3-540-77653-6_3. URL: https://doi.org/10.1007/978-3-540-77653-6_3.

Glossary

CPS Cyber-Physical Systems are feedback systems that are networked and distributed, adaptive to changes in the physical and cyber systems, and intelligent.. 16

ETX ETX measures the number of (re)transmissions needed to deliver a packet to its destination.. 56

Hurwitz matrix A structured real square matrix constructed with coefficients of a real polynomial.. 123

IoT Internet of things describes physical objects with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks.. 17

Lyapunov function A Lyapunov function is a scalar function V(y) defined on a region D that is continuous, positive definite, V(y)>0 for all y!=0), and has continuous first-order partial derivatives at every point of D. [195].. 193

Riccati differential equation A Riccati equation in the narrowest sense is any first-order ordinary differential equation that is quadratic in the unknown function.. 193

S-procedure The S-procedure is a mathematical result that gives conditions under which a linear matrix inequalities is a consequence of another inequality.. 194

Signal-to-Noise Ratio (SNR) SNR is to the ratio between the power of the desired signal and the background noise.. 45, 85, 92, 152

A

Appendix

A.1 CPS Industries

Several applications can be characterised as CPSs. The industries driving the CPS research are aerospace, building control, critical infrastructure, factory automation, healthcare, transportation and agriculture. The industries help lay down the CPS' requirements for researchers and engineers to design and tailor their solutions to specific applications.

Most CPS requirements designed by ETSI, ITU have been for applications that require ultra-low latency or short range. This can be commonly found in requirements for aerospace, factory automation, building control and are shown in Appendix A.1. This section discusses two application areas, vehicular networks and critical infrastructure that require long-range communication. A few of those applications are are slow-loop, delay-tolerant, which is the main focus of this thesis.

A.1.1 Vehicular Networks

Vehicular networks or Transportation CPS [187, 188] is an area of CPS gaining the most traction in the past few years. Vehicular networks are focussed around intelligent vehicles, public transportation networks, road management and communication between all these components. This application is used to improve the experience of users of transportation systems. Components like bikes, cars, buses, trains, pedestrians, traffic signals all form part of the vehicular networks.

Wireless communication for these applications can be categorised as intra-vehicle, vehicle to vehicle (V2V), vehicle to infrastructure (V2I) and vehicle to everything

(V2X). The intra-vehicle includes connection of the electronic control units (ECU), sensors for surrounding detection, and actuators like braking system, lights. The V2V includes emergency brake light, emergency vehicle approaching, slow or stationary vehicles, traffic jam ahead warning, cooperative collision risk warning and approaching indication[21]. The V2I, V2X includes hazardous location notification, road works warnings, weather conditions, in-vehicle speed limits, parking information and wrong-way driving[21]. Table A.1 highlights the applications and their requirements.

	Cooperative manoeuvre	**Cooperative perception**	**Safety**	**Autonomous navigation**	**Infotainment**
Latency (ms)	<30ms	50	<60	<30	< 100
Reliability (%)	99.9	99	99-99.99	99-99.99	70-80
Datarate (Mbps)	0.35-7	10-29	0.128	0.96-1.9	> 10
Device Mobility (km/h)	0-150	0-30	0-100	0-250	0-100
Comm. Range (m)	> 350	300-500	> 70	> 1000	Several kms

Table A.1: Vehicular networks wireless requirements[21]

A.1.2 Critical infrastructures

Critical Infrastructures[189] is another industry of CPS that has gained much traction. Some of the applications under critical infrastructures are gas and water pipeline systems, smart energy grids, public transportation and telecommunication networks. These applications are vital to a country, so real-time management and control are needed to ensure a country's safety, economy, and public health. These critical infrastructures are generally distributed on a city or national scale.

Wireless control of critical infrastructures has become an important research topic

as they were traditionally manually controlled or open-loop systems. However, wireless closed-loop control will allow for remote management, deal with real-time changes in demand and handle failures. Wireless communication is being actively researched in the smart grid space, including applications like advanced metering information, demand response, wide-area situational awareness, and distributed energy resource and storage. These applications deal with metering, managing demand, handling sudden changes and dealing with distributed generation. These applications and their requirements have been well-defined by the Department of Energy[190]. The requirements have been summarised in Table A.2.

An application area whose requirements are currently undefined is smart water grids or water distribution networks. It is an active area of research, with a re-imagination of the current system to make it resilient to faults or changing demands[191, 192]. Water distribution network (WDN) is a system of water and waste producers and consumers connected by a network of pipes, pumps, valves, and tanks. WDNs are an essential resource and span millions of kilometres carrying trillions of water per day. Without remote management and water-routing capabilities, a lot of water is wasted, leading to monetary losses to companies. UK water companies face several issues with pipe bursts and leaks. These water companies have 700,000kms of mains and sewers[193]. A UK resident uses 150 litres of water per day, and industries use 1300 million cubic meters per year. This means that these pipes supply an average of 13 trillion cubic meters of water per day in the UK. Between April 2017 and March 2018, there were 54611 pipe bursts which led to an average loss of 3183 million litres of water per day[194]. These pipe bursts can lead from minor problems like loss of water to a property to significant consequences like the loss of life, severe disruptions in service, damage to infrastructure and million-pound repair costs for the companies. There is a move by the government to levy fines on companies that fail to detect leaks in adequate time or cause water wastage because of inadequate maintenance of their infrastructure. By augmenting a WDN with wireless communication systems, we envision a future inter-connected

	Advanced metering information	Demand response	Wide-area situational awareness	Distributed energy resource and storage
Latency (ms)	2000-15000	500-minutes	<200	20
Reliability	99-99.99	99-99.99	99.99995	99-99.99
Comm. Range (m)	<100(in-home), kms(outdoor)	kms	kms	kms
Datarate (Kbps)	10-100	14-100	600-1500	9.6-56

Table A.2: Critical infrastructure wireless requirements[190]

water pipeline network that can reduce resource wastage and save millions of pounds.

A.2 Proof of Corollary 1 (By Anqi Fu (co-author in [41]))

Proof. With Corollary 1, we are finding the right set of linear matrix inequalities (LMIs) to verify whether our chosen parameters of sampling time, tolerable delay and rho will keep the system GES. The proof is inspired by the proofs of Heemels, Donkers, and Teel[153] and Borgers, Dolk, and Heemels[154]. In [153] (Theorem III.2), the authors show that a PETC system can remain stable if certain conditions are met. In Theorem III.3, they simplify Theorem III.2 to the case with no disturbances. This work is further extended by [154]. In [154] (Theorem IV.1), the authors add delays to the above system with disturbances, and prove the conditions for system stability. Our proof is closest to [154] (Theorem IV.1) without disturbances. The proof consists of three main steps which are presented next.

In the first step, we want to show that our chosen Lyapunov function is proper. To do so, it needs to satisfy the condition in the proof with c_1 and c_2, which is based of standard definition of input to state stability[195]. We consider the following Lyapunov

function:

$$V(\xi,\tau)=\begin{cases}\xi^{\mathrm{T}}P_1(\tau)\xi,\ \tau\in[0,\tau_d]\\ \xi^{\mathrm{T}}P_0(\tau)\xi,\ \tau\in[\tau_d,h]\end{cases} \tag{A.1}$$

where

$$\frac{\mathrm{d}}{\mathrm{d}\tau}P_i=-\bar{A}^{\mathrm{T}}P_i-P_i\bar{A}-2\rho P_i \tag{A.2}$$

with $i\in\{1,0\}$. The matrices P_{0h} and P_{1d} are defined in Corollary 1 as $P_0(h)=P_{0h}$ and $P_1(\tau_d)=P_{1d}$. Additionally, we define $P_{10}:=P_1(0)$, and $P_{0d}:=P_0(\tau_d)$.

We also consider two scalars c_1 and c_2 that satisfy $0<c_1\leq c_2$ and are given as:

$$\begin{aligned}c_1&:=\min\{\min_{\tau\in[0,\tau_d]}\lambda_{\min}(P_1(\tau)),\min_{\tau\in[0,h]}\lambda_{\min}(P_0(\tau))\}\\ c_2&:=\max\{\max_{\tau\in[0,\tau_d]}\lambda_{\max}(P_1(\tau)),\max_{\tau\in[0,h]}\lambda_{\max}(P_0(\tau))\}\end{aligned}$$

where $\lambda_{\min}(P_i)$ and $\lambda_{\max}(P_i)$ are the minimum and maximum eigenvalues of matrix P_i, respectively. According to [154] and [153], it can be shown that the function V satisfies the following inequalities:

$$c_1|\xi|^2\leq V(\xi,\tau)\leq c_2|\xi|^2.$$

In the second step, we analyse V converges during the flow of the system. This depends on our choice of P_i, which is similar to Eq.19 in [153]. Using Eq. A.1 and Eq. A.2, and derivating V along τ (V: $\tau\in[0,\tau_d]$ and $\tau\in[\tau_d,h]$) we get:

$$\frac{\mathrm{d}}{\mathrm{d}t}V(\xi,\tau)=-2\rho V(\xi,\tau).$$

The inequality shows that, the Lyapunov function converges exponentially with a decay rate 2ρ during flow. This inequality is a Riccati differential equation.

In the third step, we need to show that V does not increase during jumps. We have two jumps in our system. First, from the system state to storage function (because of sampling), and second from storage function to controller (because of transmission delays). According to [153], it holds that

$$\begin{aligned}P_{10}&=e^{2\rho\tau_d}e^{\bar{A}^{\mathrm{T}}\tau_d}P_{1d}e^{\bar{A}\tau_d}\\ P_{0d}&=e^{2\rho(h-\tau_d)}e^{\bar{A}^{\mathrm{T}}(h-\tau_d)}P_{0h}e^{\bar{A}(h-\tau_d)}.\end{aligned} \tag{A.3}$$

By applying Schur complement and S-procedure, it follows from Equation 5.8 that:

$$\xi^{+\mathrm{T}} P_{10} \xi^{+} = \xi^{\mathrm{T}} \acute{J}_{\mathcal{J}}^{\mathrm{T}} P_{10} \acute{J}_{\mathcal{J}} \xi \leq \xi^{\mathrm{T}} P_{0h} \xi$$
$$\xi^{+\mathrm{T}} P_{0d} \xi^{+} = \xi^{\mathrm{T}} \tilde{J}_{\mathcal{J}}^{\mathrm{T}} P_{0d} \tilde{J}_{\mathcal{J}} \xi \leq \xi^{\mathrm{T}} P_{1d} \xi.$$

The results of this proof guarantee that V is a proper Lyapunov function and the system is GES with convergence rate ρ which concludes the proof. □

A.3 Attributions

- Bus by alimasykurm from the Noun Project
- Motor by ic2icon from the Noun Project
- Wifi router by JunGSa from the Noun Project
- Internet server by Rank Sol from the Noun Project
- Sensor by Carolina Cani from the Noun Project
- Cover Photo artwork designed by macrovector from Freepik

