

Imperial College London
Department of Computing

Next Generation Cyber-Physical Water Distribution Systems

Sokratis Kartakis

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London and
the Diploma of Imperial College, February 2017

Abstract

Over the last decade, there has been a trend where water utility companies aim to make water distribution networks more intelligent in order to improve their quality of service, reduce water and energy waste, minimize maintenance costs etc., by incorporating Information and Communications Technologies (ICT). Current state of the art solutions use expensive power hungry deployments to monitor and transmit water network states periodically in order to detect anomalous behaviors, such as water leakage and bursts, and control water network assets. However, more than 97% of water network assets are found in remote areas, away from power and are often in geographically remote underpopulated areas; facts that make current approaches unsuitable for next generation more dynamic adaptive water networks. Battery-driven wireless sensor/actuator-based solutions are theoretically the perfect choice to support next generation cyber-physical water distribution systems.

In this context, this thesis answers the question: "*How can the communication be optimized to achieve sustainable Cyber-Physical Systems (CPS) deployed in such harsh environments exploiting limited resources by combining Information, Control, and Communication theory (I2C)?*"

In order to efficiently utilize underground wireless sensor and actuator network infrastructures, the concepts of edge data processing, anomaly detection and localization, based on compression, stream analyses and graph theory, are introduced. Furthermore, energy optimization and network sustainability by exploiting data-rate and communication scheduling adaptation, based on Lyapunov optimization, is proposed; while the benefits of aperiodic communication are investigated by accommodating event-triggered control technique into smart water networks. In addition to simulations based on real data, WaterBox and BentoBox evaluation platforms were developed to evaluate the proposed algorithms and prove the benefits of event-triggered control and Low Power Wide Area (LPWA) communication technologies against the state-of-the-art solutions. Through theoretical analysis, simulations, and real testbed experiments, the proposed algorithms and systems are shown to outperform contemporary solutions by achieving communication and actuation optimization, data reliability enhancement, while ensuring the sustainable operation of smart water networks.

The work presented in this thesis should be of interest to researchers in the emerging areas Cyber-Physical Systems (CPS), Internet of Things (IoT), and Information and Communications Technology (ICT) for smart sustainable cities.

Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives license. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the license terms of this work.

Acknowledgements

I gratefully acknowledge the funding received towards my PhD from the NEC Corporation, Japan. I would like to express my sincere gratitude to my advisor, Prof. Julie A. McCann, for the continuous support of research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

I would like to express deepest thanks to my friend Shusen Yang, who helped in my personal and academic development. Shusen is a mentor and friend, from whom I have learned the important skill of disciplined critical thinking.

I would also like to thank Usman Adeel, Yad Tahir, Babu D. Choudhary, and Greg Jackson who, as good friends, were always willing to help and give their best suggestions. I graciously thank my colleagues for providing me valuable feedback on my work.

I also thank Dr. Manuel Mazo Jr and his PhD student, Anqi Fu, for modeling my hardware evaluation platforms and providing the theory of decentralized event-triggered control. In addition, it is always a pleasure to collaborate with Dr. George Tzagkarakis under the context of compressive sensing and signal processing.

Most importantly, my PhD would not be possible without my family. My wife, Dionysia Aravantinou, and my son Christos have all my love and gratitude. During the stressful times, they had provided me a sweet home full of understanding, patience, encouragement, and happiness. I own a particular debt of gratitude to my father, Christos, and mother, Ourania, who are missing their son for so long, for their endless love and support from the first second of my life. I will never be able to thank my brothers, George and Angelos, enough for their unconditional support which allowed me to fearlessly chase my dreams knowing they always look after me.

I dedicate this thesis to my beloved wife, Dionysia and son, Christos.

Ἡ πιο ιερή μορφή της θεωρίας είναι η πράξη.’

Νίκος Καζαντζάκης

Contents

Abstract	i
Copyright	iii
Acknowledgements	v
1 Introduction	1
1.1 Motivation and Objectives	2
1.2 Contributions	6
1.3 Statement of Originality	8
1.4 Publications	8
2 Background Theory	11
2.1 Water Network Architecture	11
2.2 Low Power Wide Area Communication Technologies	14
2.3 Data Reduction Techniques	19
2.3.1 Lossless & Lossy Compression	21
2.3.2 Compressive Sensing	24
2.4 Lyapunov Drift-Plus-Penalty	29

2.5	Event-triggered Control	32
2.5.1	Centralized Event-triggered Control	34
2.5.2	Decentralized Event-triggered Control	39
2.6	Summary	42
3	Evaluation Platforms	43
3.1	WaterBox: Water Network Monitoring & Control Testbed	45
3.1.1	Design Overview	46
3.1.2	Hardware Infrastructure	47
3.1.3	Software Infrastructure	54
3.2	BentoBox: Low Power Wide Area Communication Evaluation Platform	56
3.2.1	Prior Limited LPWA Evaluation Studies	57
3.2.2	BentoBox Hardware Infrastructure & LPWA Experimentation	58
3.2.3	LPWA Communication in Smart Water Networks	66
3.3	Summary	67
4	Data Reduction & Edge Processing	69
4.1	Adaptive Thresholding-based Real-time Anomaly Detection exploiting Compression Rates	71
4.1.1	Compression Rate and Raw Data Trend Correlation	72
4.1.2	Adaptive Thresholding-based Anomaly Detection Algorithm	74
4.1.3	Optimal Self-Tuning via Active Learning	77
4.1.4	Extended Evaluation	80

4.2	Incorporating Edge Analytics and Off-line Graph-based Burst Localization Algorithm	81
4.2.1	Prior Related Work	82
4.2.2	Edge Anomaly Detection based on Pipe Vibration Data	84
4.2.3	Graph-based Water Burst Event Localization	85
4.2.4	Evaluation	93
4.3	Applying Compressive Sensing to Water Networks	97
4.3.1	Methodology	97
4.3.2	Evaluation	99
4.4	Energy-based Adaptive Compression in Water Network Control Systems	102
4.4.1	System Overview and Preliminaries	102
4.4.2	Optimal Compression Technique Selection Formulation	105
4.4.3	Compression Rate Selection Algorithm	106
4.4.4	Evaluation	107
4.5	Summary	110
5	Sustainable & Reliable Scheduling for Smart Water Networks	113
5.1	Related Work	117
5.2	Preliminary	118
5.3	System Overview and Modeling	119
5.3.1	Sensor Nodes	120
5.3.2	Data Processing Center	123
5.4	Anomaly Definition & Detection	126

5.5	Sustainable Scheduling for Raw Data Transmission	128
5.5.1	Monotonically Non-Decreasing Estimation Reliability	128
5.5.2	Stochastic Scheduling Optimization	129
5.5.3	Data Transmission Scheduling (DTS)	130
5.5.4	Sustainability & Optimality Proof	131
5.5.5	NP Completeness of the DTS	132
5.5.6	FAST-DTS A Linear Approximation of DTS	133
5.6	Data Stream Estimation	134
5.7	Evaluation	135
5.7.1	Evaluation Set-up	135
5.7.2	A Case Study	136
5.7.3	System Adaptation in Anomalies	138
5.7.4	The Impacts of System Parameter Settings	140
5.7.5	Scalability Study	141
5.8	Exploiting Distributed Compressive Sensing	141
5.9	Summary	144
6	Event Trigger Control & Communication Framework	146
6.1	Event-Trigger Control Techniques	149
6.1.1	Periodic control	149
6.1.2	Periodic centralized event-triggered control	150
6.1.3	Periodic synchronous decentralized event-triggered control	151
6.1.4	Periodic asynchronous decentralized event-triggered control	153

6.2	Incorporating ETC with the MAC Layer	156
6.2.1	Simplistic TDMA Protocol	156
6.2.2	Control-TDMA Protocol	157
6.2.3	Synchronous Decentralized Control-TDMA Protocol	159
6.2.4	Asynchronous Decentralized Control-TDMA Protocol	160
6.3	WaterBox Infrastructure & Modelling	161
6.3.1	WaterBox infrastructure	162
6.3.2	System Identification & Modelling	164
6.4	Switched Controller Design	164
6.5	Evaluation	167
6.5.1	Evaluation Setup	167
6.5.2	Experimental Results	169
6.6	Summary	174
7	Conclusion	176
7.1	Summary of Thesis Achievements	176
7.1.1	How much information to transmit?	177
7.1.2	Which node will transmit?	179
7.1.3	When to transmit?	180
7.2	Future Work	182
7.2.1	Smart Scheduling and Distributed Compressive Sensing	182
7.2.2	Incorporate Validation, Security and Event-Triggered Control	183
7.2.3	Large Scale Implementation	183

A Proof in Chapter 2	185
A.1 Proof of Rate Stability	185
B Proofs in Chapter 5	187
B.1 Proof of Sustainable Operation	187
B.2 Proof of Asymptotic Optimality	188
C WaterBox dynamical model in Chapter 6	191
C.1 Hydraulic component definitions	191
C.2 WaterBox Model	193
C.2.1 Dynamics at different points	193
C.2.2 System model	195
Bibliography	196

List of Tables

1.1	Technical and mechanical needs/challenges.	3
3.1	Technical parameters LPWA modules	60
4.1	Algorithm input parameters	77
4.2	Edge Processing Evaluation Results	94
4.3	Energy parameters	108
4.4	Genetic algorithm parameters	108
4.5	Savings compared to periodic communication (a) without compression and (b) with only lossless compression	110
5.1	Frequently used symbols	126
5.2	Evaluation results of the case study.	137
5.3	Average results of 80 sensor nodes for 30 days ($B_{\text{exp}} = 150$ and $\text{rlb}_{\text{min}} = 0.98$). . .	141
6.1	Communication parameter evaluation setup.	167
6.2	Parameters of triggering strategies.	168
6.3	Savings compared to TTC (%) total experiment time, i.e. $[0, t_{\text{end}}]$, with $\sigma =$ $0.2, \rho = 85$, and $\mu = 0.95$	173

6.4	Savings compared to TTC (%) until mode switching time, i.e. $[0, t_{sm}]$, with $\sigma = 0.2, \rho = 85$, and $\mu = 0.95$	174
C.1	Definition	192

List of Figures

1.1	Smart water network as a cyber-physical system.	2
1.2	Next generation CPSs require the fusion of I2C.	5
1.3	Thesis contributions.	6
2.1	DMA-based water network architecture.	12
2.2	An illustration of a smart water system and sensor node hardware.	14
2.3	IoT Applications and LPWA Communication Scenarios.	15
2.4	Data fusion terms.	20
2.5	Aggregation routing approaches.	20
2.6	Lossless compression methods, libraries, and applications.	21
2.7	Lossless compression algorithms evaluation process.	22
2.8	Standard compression algorithms memory usage.	23
2.9	Lossy Compression - Raw data segmentation.	23
2.10	Compressive Sensing process.	25
2.11	Distributed Compressive Sensing process.	27
2.12	Sensor node operation example for Lyapunov Drift-Plus-Penalty method.	30
2.13	Ideal smart water network.	32

2.14	Input to State Stability bounds.	36
2.15	Control flow diagram in embedded systems.	37
2.16	Event-triggered threshold vioaltion.	39
2.17	Centralized to decentralized event-triggered control systems.	40
2.18	Background knowledge and chapter assignment.	42
3.1	Chapter 3: Evaluation Platforms and Supportive Testbed Infrastructures.	43
3.2	Testbed design overview.	46
3.3	Testbed DMA reconfiguration scenarios.	47
3.4	WaterBox hardware infrastructure.	48
3.5	WaterBox sensor node and power analysis.	49
3.6	Pumps and controller infrastructure.	51
3.7	Smart water shield.	52
3.8	Sensor/ actuator node software infrastructure.	54
3.9	Experiment visualization software infrastructure.	55
3.10	BentoBox: Hybrid LPWA Communication Box.	59
3.11	Base station (Receiver - Rx) and moving node (Transmitter - Tx) installation.	60
3.12	Experimental Environments	61
3.13	Data rate, power and energy consumption comparison.	63
3.14	Success Rate chart for different modules/modes and range.	64
4.1	Chapter 3: Data Reduction & Edge Processing.	69
4.2	Compression algorithms evaluation.	72

4.3	Compression rate & raw data comparison of Dataset A, B, and C.	73
4.4	Apply Kalman Filter to Dataset A and drop detection.	74
4.5	Dataset C - (a) "Cold Start", Large Variation and unelastic outliers detection ($l = 1.5$), and (b) fixed algorithm results ($l = 3$).	76
4.6	Algorithm results ($l=3$) for (a) Dataset A and (b) Dataset B.	77
4.7	Sensor node re-configuration process.	78
4.8	Cross evaluation distance calculation results - Dataset A.	79
4.9	Applying edge anomaly detection algorithm in temperature data.	80
4.10	System architecture: sensor nodes, communication, and server side system. . . .	81
4.11	Applying edge anomaly detection algorithm in vibration data.	84
4.12	Modelling a water network (left) as a weighted graph (right)	85
4.13	Lab-based pipe rig floor plan and infrastructure.	93
4.14	Anomaly detection from three sensor nodes (A, B, and C) which are located 2m, 8.7m, and 13.7m from the burst respectively.	94
4.15	Anomaly detection periods.	95
4.16	± 0.5 m accuracy in localizing the burst event	96
4.17	Compressive sensing information transformation flow diagram	97
4.18	Bristol area testbed and pressure data from 4 sensor nodes	99
4.19	Average RMSRE per CS sampling rate for the four nodes.	101
4.20	Original and reconstructed segments for sensor node 2 using NESTA algorithm .	101
4.21	Reconstruction error ($RMSRE$) and compression rate	104
4.22	Optimal compression selection evaluation results	109

5.1	Chapter 5: Sustainable & Reliable Scheduling for Smart Water Networks.	113
5.2	Pressure wave propagation delay and attenuation of sequential nodes	119
5.3	Illustration of the System architecture.	120
5.4	Dynamic energy generated by water flow.	122
5.5	The time-varying correlation graph topology at two intervals t_1 and t_2	124
5.6	Observed correlation between pressure fluctuation and compression rates.	127
5.7	results of linear regression for 1 day training set and combination size	129
5.8	Reliability-Greedy and FAST-DTS comparison based on reliability and battery level per node	138
5.9	Testbed modification for anomaly emulation.	138
5.10	Correlation adaptation based on anomaly variance.	139
5.11	Reliability, energy waste, and transmission gaps with different $rlb_{\min}$ and $B_{\exp}$. .	140
6.1	Chapter 6: Event Trigger Control & Communication Framework.	146
6.2	Simplistic TDMA MAC Protocol.	157
6.3	CTDMA protocol.	158
6.4	SDC-TDMA Protocol for PSDETC.	159
6.5	ADC-TDMA Protocol for PADETC.	160
6.6	The schematic structure of the WaterBox.	161
6.7	Hybrid controller state automaton.	166
6.8	Experimental results. Note that, for SETC, the maximum event-interval of small tank 2 is 66 sec.	169

6.9	Impact of ETC parameters (σ, μ, ρ) in: (a) water level overshoot, (b) switching time, (c) sleep time, (d) discharge, (e) actuations, (f) valve movement, (g) violation, and (h) state transmissions.	172
6.10	Impact of interval length T in: (a) water level overshoot, (b) switching time, (c) sleep time, (d) discharge, (e) actuations, (f) valve movement, (g) violation, and (h) state transmissions.	173
7.1	Summary of Thesis Achievements.	176
7.2	Thesis achievements as a system architecture.	181
C.1	WaterBox hydraulic components: indexes and positions.	191

Chapter 1

Introduction

Next generation smart cities, smart buildings, driverless vehicles, precision agriculture etc. will go beyond sophisticated telemetry and will enable automatic distributed control. Cyber-Physical Systems (CPS) will move beyond merely understanding people, places and things via sensing and analysis to being able to close the loop and provide more automation. High precision sensing from a heterogeneous collection of sensor devices to establish an accurate and timely understanding of that systems' state constitutes the main requirement of such systems. The extreme complexity of analytics required to process these volumes of data limits what can be understood about the system in real-time. Yet, these real-time analytics is what is required to enable automatic control. The only way to overcome this problem is to reduce the dimensionality by reducing the data and therefore the challenge is to do this without losing information.

The overarching aim of this thesis is to better understand cyber-physical systems such as next generation water distribution networks and smart cities and to better control them through the development of integrated self-adaptive protocols that support distributed and collaborative analysis, prediction, and control using remote terminal units and servers and which provides guarantees pertaining to reliability, stability, convergence, and sustainability.

1.1 Motivation and Objectives

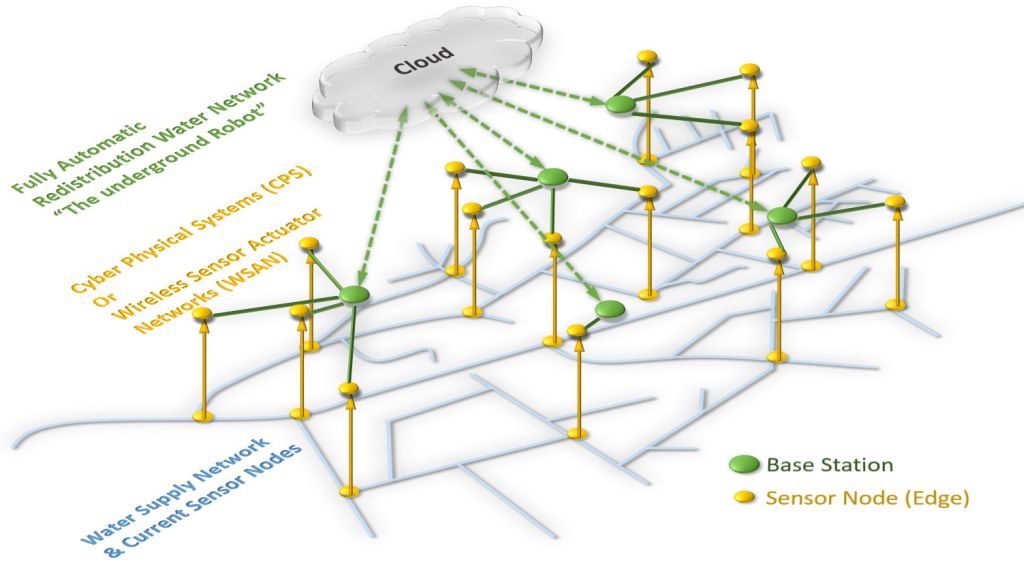


Figure 1.1: Smart water network as a cyber-physical system.

Over the last decade, there has been a trend where water utility companies aim to make water distribution networks more intelligent in order to improve their quality of service, reduce water waste, and minimize maintenance costs etc., by incorporating ICT technologies (see Figure 1.1). Current state of the art solutions use expensive power-hungry deployments to monitor and transmit water network states periodically in order to detect anomalous behaviours such as water leakage and bursts. However, more than 97% of water network assets are away from power and are often in geographically remote underpopulated areas; facts that make current approaches unsuitable for next generation more dynamic adaptive water networks due to their energy and maintenance cost requirements. Table 1.1 aggregates the technical and mechanical challenges of smart water network infrastructures as have been present in [1] by British Water.

Water security is currently a hot topic [2, 3]. Water demands are not being met in regions of the world, both developed and underdeveloped, where climate change and economic water scarcity are two issues that have the largest impact. The former sea areas of the planet less able to generate enough water for their inhabitants, whereas in economic scarcity the principality is unable to build or maintain a water distribution network to continuously meet demands. Drought prone areas such as California in the USA have had severe water restrictions in place for some time. Wet countries, such as the UK, have been experiencing what has been termed

Table 1.1: Technical and mechanical needs/challenges.

Technical Challenges	Mechanical Challenges
Long Battery Life	No digging - no wires
Regular reporting	Installable without modification to manhole or boundary box cover
Cyber-attack proof	Ideally can be installed from the surface to eliminate confined space entry
Secure	Tolerate total submersion IP68+
Deployable anywhere	Physically small antenna by non-obstructing sewer entry or connecting cables to covers
Compatible with existing equipment	Rapid fit
Fault tolerant - self testing	-

wettest droughts over the past few years. Notwithstanding the 7.5bn investment in UK water distribution networks, 3.3bn litres of water were lost per day in 2010 [4].

Furthermore, according to the Guardian [5] and Ofwat [6] data, more than half of the water companies have not reduced their leakages at all until 2015; fact that increased the billing cost. More specifically:

- During 1998-2011 there was only a 5% cut in leaks by the 21 water companies serving England and Wales.
- 10% - the target for the next 25 years according to Ofwat.
- 11 of the 21 water companies have 0% leak targets for years by 2017.
- Southern water, which missed its leakage target (again) in 2010-11 by 16%, was allowed to "increase" leakage by 6% by 2014-15.
- 3.36bn litres of water a day are currently being leaked, and aiming for 3.24bn by end of 2016 - roughly 25% of all water.
- Average bills have risen by £64 in last decade to £376 a year mostly due to leakage.

The use of sensing systems to identify water leakage have been around for some time [2], but their uptake has not been prolific being unable to detect and localize leakages or burst accurately. ICT to support Water Distribution Network (WDN) typically consists of remote or online battery-powered telemetry units (data loggers) that record water data such as flow and pressure, over numbers of minutes, then aggregate this data and send to a server periodically - typically via the mobile phone networks. Contemporary approaches use Wireless Sensor

Network (WSN) [2, 3, 7–9] technologies to monitor the status of the water network and detect leakage or water bursts closer to real-time. For instance, in [10], a testbed with 25 sensor nodes has been deployed in Singapore's water network. These sensor nodes transmit high sample rate flow and pressure data in 30-second and 5-minute intervals respectively through 3G or WiFi to a server which then analyses the data. Similarly, in [11], a "smart water networks demonstrator" with 24 sensor nodes has been deployed in parts of an operational network to evaluate sensing, data acquisition, analytical and control technologies, to assess the operational benefits of dynamic District Metered Areas (DMAs¹) [12] and facilitate testing of communication algorithms.

The main drawbacks of these approaches are: (a) the analysis of the data takes place offline, in base-stations or servers meaning that optimal real time decision-making for control would be unrealistic and (b) the sensor nodes require a lot of energy, which places upper bounds on the amounts of data that can be sensed and relayed for analysis. The latter issue particularly impacts on what is important to water companies: leakage localization. Given the cost of digging up roads to fix leaks, timely and accurate determination of the location of a leak is now more important than identifying just the existence of leakages. However, to carry out localization high-fidelity sensing and analytics are required to triangulate leakage accurately; current systems are not quite there yet [3].

In parallel, civil engineers advocate that next generation water networks will not be passive water delivers, but active highly-distributed control systems that route the water intelligently to match demand and route around failures etc. [11]. Such a dynamical system will heavily rely on sensing and actuation and will effect a dual control system of a water distribution network coupled with a smart distributed sensor/actuator network. This will require that both networks interact in a complex cyber-physical way and make use of state-of-the-art low-powered high-precision sensing and processing technologies. State of the art high-precision sensors can help build such a system, but it is too costly, and not very useful, to send all this higher precision data back to off-line servers through a long range communication for processing. Two

¹This is a defined area of the water distribution system that can be isolated by valves and for which the quantities of water entering and leaving are measured/metered.

of the few sophisticated operational control systems include I²O [13] and Derceto [14], who have developed and applied "optimal" automatic remote control of pressure regulator valves (PRVs) and pumps, respectively. However, these systems are based on diurnal training that produces static control schedules for predefined periods of time. Such methods have a limited capacity to respond to disturbances to the system in real-time, which can make their operations sub-optimal.

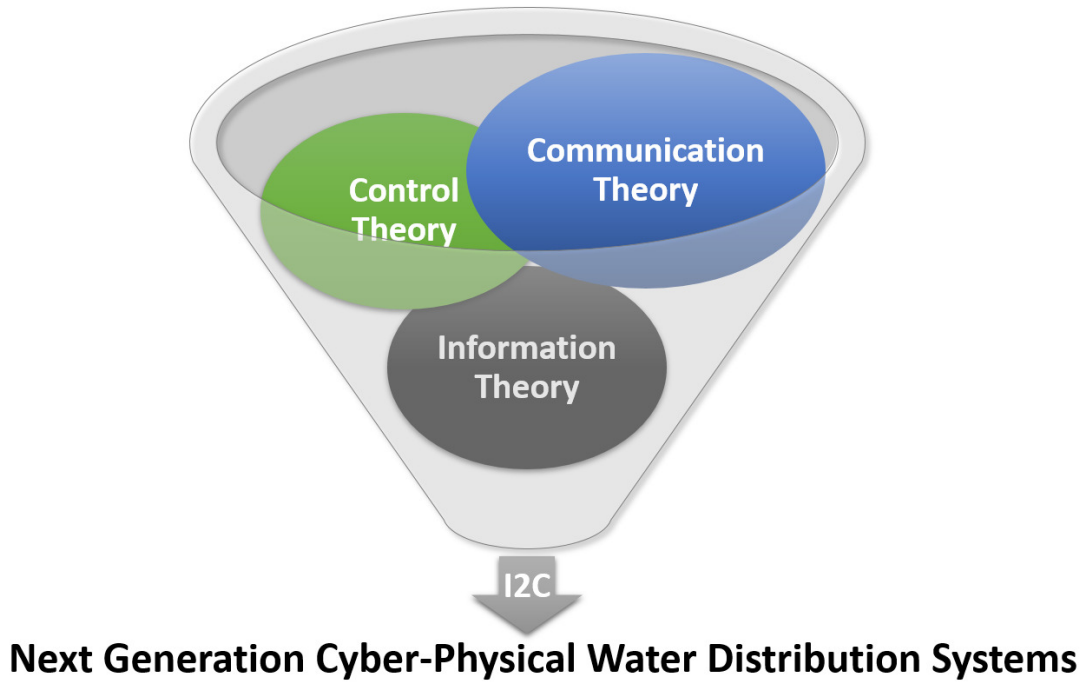


Figure 1.2: Next generation CPSs require the fusion of I2C.

Based on the above, none of the current state-of-the-art systems is able to cover the real needs of industry. In order to fill the gap and utilize the underground energy-constrained wireless sensor and actuator network infrastructure efficiently, this thesis aims to address smart water network efficiency from an innovative perspective by combining Information, Control, and Communication theory (I2C) and introducing concepts such as in-node data processing, compression and decision making, event-driven communication based on control theory, and energy sustainability by exploiting energy harvesting, and data-rate and communication scheduling adaptation.

1.2 Contributions

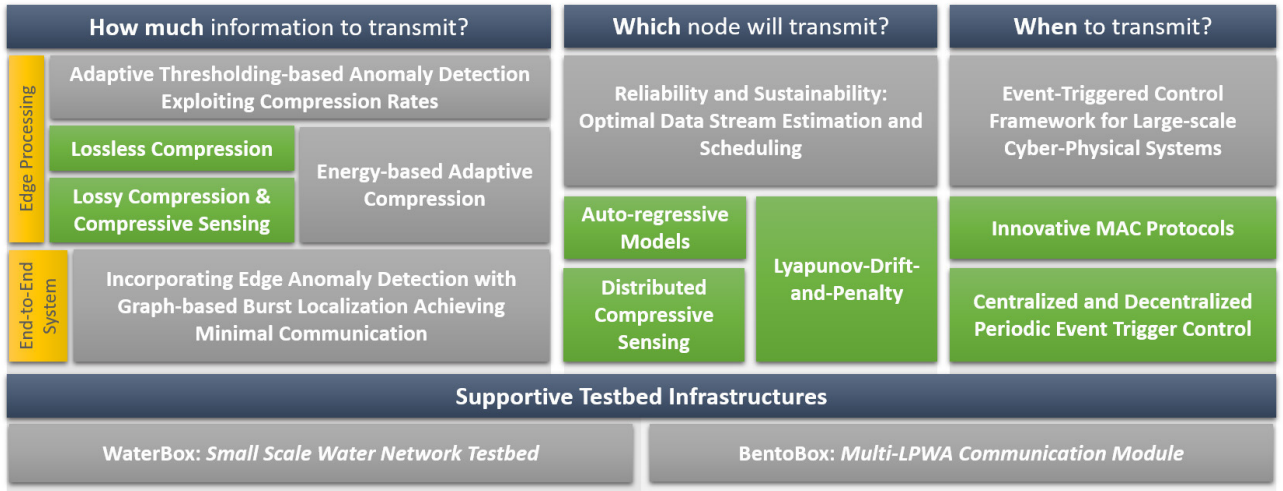


Figure 1.3: Thesis contributions.

The broad aim of this thesis is to optimize the communication over energy-constrained large-scale CPSs using as proof of concept smart water networks. Specifically, we tackle the high demanding communication layer of the harsh environment of smart water networks by answering three question:

- **How much information** must be transmitted by each node?
- **Which node** should transmit at every communication period?
- **When** has each sensor to transmit the local state of the physical layer?

This thesis makes the following contributions to the state-of-the-art:

- **Chapter 3 - *Evaluation Platforms*:** Control constitutes a critical process for water networks and experiments may be disastrous in a real deployment. To create a fail-safe environment for water network event-driven control experiments, a small-scale testbed, WaterBox, was developed. In addition to control, existence of long-range communication is required for water networks due to the environments that they are in. Therefore, we need to select communications technologies that interface the sensor/actuator nodes to data centres at city scale and beyond. To this end, we look beyond the data sheets and white papers of Low Power Wide Area (LPWA) communication technologies [15] and provide insights into the performance of three emerging LPWA solutions (Lora, XBee868/Zigbee,

and NWave/Weightless) based on real world experiments using our innovative sensor node, BentoBox, with different traffic loads, in different urban deployment contexts and under the same environmental conditions. The experimental results of the conducted experiments were used as evaluation setup for our algorithms.

- **Chapter 4 - *Data Reduction and Edge Analytics*:** An algorithm to jointly detect anomalies while compressing data losslessly [16] at the same time into the sensor node, so-called edge, is being presented. By incorporating this in-node anomaly detection feature with an efficient graph-based burst localization algorithm, we propose an end-to-end system which enables a minimal communication scheme. Alternatively, in order to reduce more the amount of transmitted data compare to the lossless compression, lossy compression techniques were introduced. Specifically, we exploit the power of compressive sensing (CS) [17], which enables significant compaction of the original information content in a few random incoherent mathematical projections by introducing negligible reconstruction error. To further utilize and balance the trade-off of error-free lossless compression and powerful compressive sensing performance, an edge adaptive compression scheme is been developed. As far as we know, this is the *first* time that the edge processing is exploited in smart water networks and the *first* edge anomaly detection approach based on the compression rate analysis.
- **Chapter 5 - *Scheduling and Sustainability*:** In order to achieve sustainable sensor node operation by utilizing the limited and dynamic harvested energy, data-rate and communication scheduling adaptation was examined. In this chapter, we exploit the physical features of water networks to accurately estimate water pressure conditions by using data from neighbor correlated sensor nodes instead of the local sensor node using multi-variable auto-regressive models [18]. To compute the optimal scheduling policy, we formalize a stochastic optimization problem, based on Lyapunov Drift-and-Penalty method [19], to maximize the estimation reliability, while ensuring the systems sustainable operation under dynamic conditions. To solve the optimization problem, Data Transmission Scheduling (DTS), a theoretically proven optimal scheme, and FAST-DTS, a lightweight online algorithm that can adapt to arbitrary energy and correlation dynamics were developed. In

addition, Distributed Compressive Sensing (DCS) [20] is being examined as alternative to auto-regressive model-based estimation. To our knowledge, this is not only the *first* time that data stream estimation is adopted to enable sustainable smart water systems, but also the *first* scheduling approach based on data stream estimation reliability in energy harvesting networks [21].

- **Chapter 6 - *Event-Triggered Control*:** Control methodologies must incorporate and integrate the many constraints that wireless communication and energy aware computer systems bring. To be able to build the next generation adaptive control system for water networks under a hardware constrained infrastructure, we introduce the notion of automatic and aperiodic control by exploiting Event-Triggered Control (ETC) [22]. To enable ETC in large-scale CPSs, we propose and implement three innovative MAC protocols to support two centralized and three decentralized periodic event trigger control approaches. To the best of our knowledge, this is the *first* real deployment of the asynchronous decentralized ETC techniques to a real plant and the *first* design and implementation of the corresponding MAC protocol.

1.3 Statement of Originality

I declare that the content of this thesis was composed by myself. 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.

1.4 Publications

The work presented in this thesis is supported by the following publications.

1. Sokratis Kartakis and Julie A. McCann. **Real-time Edge Analytics for Cyber Phys-**

- ical Systems using Compression Rates.** *In 11th International Conference on Autonomous Computing (ICAC), pp. 153-159, USENIX, IEEE, 2014*
(Contributes to Chapter 4)
2. Sokratis Kartakis, Edo Abraham, and Julie A. McCann. **WaterBox: A Testbed for Monitoring and Controlling Smart.** *In 1st International Workshop on Cyber-Physical Systems for Smart Water Networks (CysWater), p. 8, CPSWeek, ACM, 2015*
(Contributes to Chapter 3)
3. Sokratis Kartakis, Alexander D. Gluhak, and Julie A. McCann. **Real world performance characterization of Low Power Wide Area technologies for IoT applications,** *Intel Event, 2015.*
(Contributes to Chapter 3)
4. Sokratis Kartakis, George Tzagkarakis, and Julie A. McCann. **Adaptive Compressive Sensing in Smart Water Networks.** *In 2nd International Electronic Conference on Sensors and Applications (ECSA), Multidisciplinary Digital Publishing Institute, 2015*
(Contributes to Chapter 4)
5. Sokratis Kartakis, Marija Milojevic Jevric, George Tzagkarakis, and Julie A. McCann. **Energy-based Adaptive Compression in Water Network Control Systems.** *In 2nd International Workshop on Cyber-Physical Systems for Smart Water Networks (CysWater), pp. 43-58, CPSWeek, IEEE, 2016*
(Contributes to Chapter 4)
6. Sokratis Kartakis, Weiren Yu, Reza Akhavan, and Julie A. McCann. **Adaptive Edge Analytics for Distributed Networked Control of Water Systems.** *In 1st International Conference on Internet-of-Things Design and Implementation (IoTDI), pp. 72-82, IEEE, 2016*
(Contributes to Chapter 4)
7. 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 10th International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization (WiNTECH), pp. 2-8, MobiCom, ACM, 2016*

(Contributes to Chapter 3)

8. Sokratis Kartakis, Shusen Yang, and Julie A. McCann. **Reliability or Sustainability: Optimal Data Stream Estimation and Scheduling in Smart Water.** *ACM Transactions on Sensor Networks (TOSN), Revised.*

(Contributes to Chapter 5)

9. Sokratis Kartakis, Anqi Fu, Manuel Mazo Espinosa, and Julie A. McCann. **Evaluation of Decentralized Event-Triggered Control Strategies for Cyber-Physical Systems.** *IEEE Transactions on Control Systems Technology (TCST), arXiv preprint arXiv:1611.04366, Under Review*

(Contributes to Chapter 6)

10. Sokratis Kartakis and Julie A. McCann. **Communication Technologies for Smart Water Grid.** *Book Chapter in Water Smart Grids: A Cyber-Physical Approach, Accepted by CRC Press, 2016*

(Contributes to Chapter 2 - 4)

In addition to my research, during my PhD, I involved in the Ninth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2015), Cambridge, MA as Demonstrations, Contests and Poster chair, and I was awarded with

1. 1st Place in Intel IoT Roadshow Hackathon, London 2015, for "Smart Vending Machine and Dynamic Pricing" project
2. 2nd Runner-up Best Poster by a 1st year PhD student at Google 2014 Poster and Demo Competition

Chapter 2

Background Theory

This chapter briefly introduces the necessary theoretical background related to the main work presented in this thesis. Specifically, this chapter describes the physical structure of contemporary smart water networks; provides the theory behind Low Power Wide Area (LPWA) communication technologies which are exploited in Internet of Things (IoT) and city-scale CPS applications; aggregates data reduction techniques over wireless sensor networks focusing on lossless and lossy compression, and centralized and distributed compressive sensing; explains the Lyapunov-Drift-Plus-Penalty method and the benefits in network scheduling; and analyzes the centralized and decentralized event-triggered control strategies. In addition, detailed backgrounds relating to the covered topics will be provided in each chapter.

2.1 Water Network Architecture

Contemporary water network structures have been applied to facilitate the monitoring and control of water distribution within city scale deployments. While valves control flows and pressures at fixed points in the water network, pumps pressurise water to overcome gravity and frictional losses along supply zones, which are divided into smaller fixed network topologies (on average of 1500 customer connections in the UK) with permanent boundaries, District Meter Areas (DMAs). The water pressure distribution and flows into each DMA is continuously

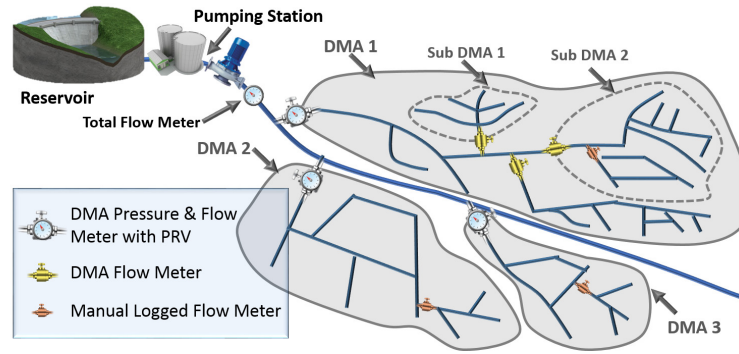


Figure 2.1: DMA-based water network architecture.

monitored with the aim to enable proactive leakage management, simplistic pressure management, and efficient network maintenance. For more accurate results, contemporary strategies implement DMAs in even smaller areas so-called sub-DMAs with in average 500-1000 customer connections. Based on the above, a DMA-based water supply network structure consists of three individual layers (Figure 2.1): (a) storage and pumping, (b) supply zones and DMAs, and (c) end users (customers).

Operational control problems include optimisation based pump scheduling and valve operation. For a network with given demand patterns, we determine a control sequence for pumps and valves, over a certain time horizon into the future (often 24 hours). This is done in order to minimize pumping costs and reduce leakage while satisfying customers' demands with sufficient hydraulic pressure. In most water distribution systems, the pumping of treated water from reservoirs to supply zones and storage tanks consumes most of the energy budget for a utility. For example, by a clever use of the storage tanks and reservoirs, pumping schedules can be shifted to cheap tariff periods resulting in significant savings [23,24]. Such pump scheduling is particularly challenging because the associated mathematical optimization problems combine integer variables (pump ON/OFF) with continuous nonlinear constraints of nodal pressures and pipe flows; these result in optimization problems of Mixed Integer Nonlinear Programming (MINLP) type. Since the difficulty in solving MINLP problems scales prohibitively large with problem size, such approaches cannot be used in a near real time operational setting for large scale water networks. In [24] and references therein, the problem can be approximated by one that finds optimal reservoir trajectories. A much smaller global optimization method can

then be solved for each pumping station to find efficient pumping schedules that give rise to the already computed optimal trajectories. Our work studies the performance of stable new control methods that track optimal reservoir level trajectories in the presence of disturbances and measurement noise; concepts that will be tested on WaterBox include: centralised and distributed controllers, communication energy minimizing protocols, and efficiency of different controllers under disturbances.

One of the main objectives of current research is the investigation of optimal hydraulic and topology control of water network systems to reduce pressure for leakage and pumping cost reduction. Many studies have shown that pressure management with internal and inlet PRVs is a cost effective and efficient way to reduce leakage levels [25]. Preliminary work on adaptively reconfigurable DMAs [12] has also shown the effectiveness of aggregating DMAs into larger controlled pressure zones especially during low water demanding periods i.e. 4am-2am; this is done by controlling the DMA boundary PRVs. For example, model predictive control (MPC) can be used with demand forecasts to minimise pressure, which reduces pressure driven leakage, while satisfying minimum pressure constraints at critical points in the network. These are implemented as flow-modulation curves (outlet pressure is controlled as a function of measured flow) [26]. The use of multi-inlet DMAs, in addition to giving better pressure performance, improves the security of supply. However, the interaction between modulating valves can cause instabilities (pressure oscillations across the network). Therefore, state-of-the-art systems examine stable optimal pressure control methods for multi-inlet DMAs [3].

To apply the optimal control techniques, data acquisition from real water network is required. Smart water networks in [2,10] constitute the first fully instrumented water distribution systems. Specifically, [10] is a smart water system proposed by MIT and adopts large overground solar panels and direct power from lampstands, requiring expensive instrumentation and deployment cost. Similarly, under the context of the Smart Water project [11], a smart water networks demonstrator was built by InfraSense Labs [27] under the supervision of our project partner Dr. Ivan Stoianov. In this project, 24 InfranSenseTS sensor/actuator nodes [28] were deployed to the inlet of an equal number of DMAs in Bristol Water network, which record high sample-rate pressure and flow data (up to 128 Samples/sec) and cover an area of approximately 7,544

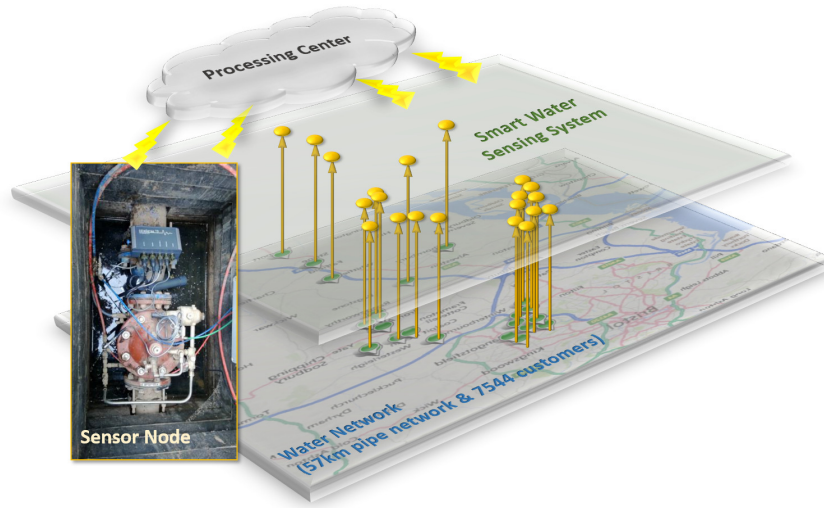


Figure 2.2: An illustration of a smart water system and sensor node hardware.

customer connections and 57km of pipe mains as shown in Figure 2.2. As collaborators to this project, high sample-rate pressure data from the 24 sensor nodes for a 170-day period was provided in order to conduct our research and to simulate our algorithms as will be presented in Chapter 4 and Chapter 5.

By having a DMA-based water network structure as an assumption for the topology of the physical layer of our city-scale CPS case study, the next step is to examine appropriate communication techniques to facilitate the transmission of the information from the mostly underground sensor/actuator nodes to the overground base stations or server side infrastructures. Contemporary Low Power Wide Area (LPWA) communication technologies, which exploit low frequencies, constitute potential solutions to enable the communication in city-scale systems and harsh environments, such as water distribution networks.

2.2 Low Power Wide Area Communication Technologies

Low power wide area (LPWA) communication technologies have recently emerged as a viable alternative to cellular and mesh networks for providing cost effective IoT connectivity in cities. Several standards and technologies are now competing, with data sheets and early deployment success stories serving as main weapons to attract customers. With several manufacturers and consortia now announcing national rollouts using different LPWA technologies, IoT solution

and connectivity providers are now faced with the dilemma of what technology may be applicable in different environments or scenarios. Unfortunately, there is little empirical evidence that provides a good understanding of the performance tradeoffs of these solutions applied to different application requirements and deployment contexts [15].

IoT deployments in urban environments for smart city applications provide one of the most appropriate scenarios for LPWA technologies (see Figure 2.3). However, they also impose diverse requirements on LPWA technologies in terms of application demands and deployment contexts. To illustrate the case let us briefly look at two simple smart city application examples.

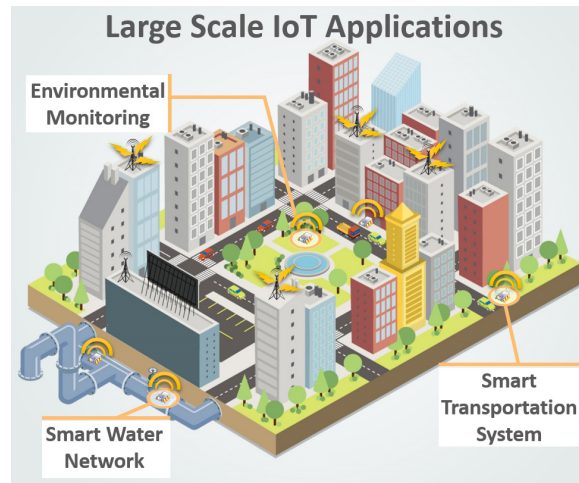


Figure 2.3: IoT Applications and LPWA Communication Scenarios.

Typical urban environmental monitoring, such as in a park, requires a small number of samples (e.g. water quality or soil conditions) to be taken at regular intervals sparsely distributed throughout the day. This is situated in an environment that is characterized in terms of communications either as line of sight or non-line of sight (LoS or NLoS respectively) - vegetation or people here tend to be the main obstructions. The delivery of data is delay-tolerant, that is the exploitation of the data is typically non-time critical. On the other hand, the built environment brings additional challenges, it remains a delay-tolerant class of application but the penetration of radio through built spaces can greatly impact the design choice for IoT. In contrast, monitoring of critical infrastructures (e.g. leak detection in a water network) has quite different requirements. Data transmission is event-based in nature, and only occurs if critical events or warning signs are detected. The penetration challenge here is exacerbated due to underground deployments of nodes, which places them in a very harsh radio environment. As

the effects of leaks or contamination can be severe, the delivery time of such events is critical and communication delays are not tolerated.

LPWA technologies are emerging with new low frequency physical layer communications capable of leveraging the trade-off between range and data-rates, and which are suitable for non-data-intensive IoT applications. Communication theory suggests that data rates, link distances and power consumption can be mutually traded in the physical layer design depending on the application scenario; yet existing wireless devices typically leverage the trade-off between range and power consumption as data rate is often non-negotiable for many application scenarios.

In order to increase the communication range, a wireless device essentially needs to increase either the transmission power of the transmitter or sensitivity of the receiver. This in turn increases the link budget of the communication system consisting of transmission (Tx) and receive (Rx), which technically implies enhancing the signal-to-noise ratio (SNR)¹ at the receiver. Spread-Spectrum and Ultra-Narrow-Band (UNB) are two candidate technologies offering low-power long-range communication, and based on these technologies there are some specially tailored products being offered by various vendors. For example, LoRa by Semtech [29] uses Chirp Spread-Spectrum modulation, and NWave [30], Sigfox [31] etc. are offering modules based on Ultra-Narrow-Band technology.

Spread-Spectrum Technology based devices spread the energy of the signal over a wide band which effectively reduces the spectral power density of the signal. This is accomplished by convoluting the band-limited information signal with a much higher bandwidth pseudo-random chirp sequence. This is popularly known as Direct Sequence Spread Spectrum (DSSS). Another variant of spread spectrum technology is based on frequency-hopping, which generates a wide band by transmitting at different frequencies, jumping from one frequency to another according to a pseudo-random code sequence. The ratio of the RF bandwidth (signal to be transmitted) to the bandwidth of the information signal is termed as processing gain which proportionately enhances the SNR, which implies that increasing the processing gain results

¹Signal-to-noise ratio is defined as the ratio of the power of a signal (meaningful information) and the power of background noise (unwanted signal)

in underutilization of the spectrum (and a decrease in data rate) but in turn improves the range [32].

Ultra-Narrow-Band (UNB) Technology attempts to enhance the SNR by reducing the noise-floor of the receiver by narrowing down the receive bandwidth (BW_{Rx}) which in turn reduces the modulation bandwidth resulting in a lower data rate. BW_{Rx} is related to the noise floor of the system as per equation: $PdBm = -174 + 10\log_{10}(BW_{Rx})$ i.e., the noise floor scales with the receive bandwidth (-174 dBm is the thermal noise floor at room temperature in a 1-Hz bandwidth) [33].

Conventional Narrow-Band Systems, such as XBee868, use modulation schemes (e.g. MSK, GMSK etc.) which change frequency and phase gradually during the whole bit-interval, which result in upper and lower sidebands around the carrier frequency. Ultra-Narrow-Band devices are attempting to remove the side-lobes [34] in order to narrow-down the channel-width, and they achieve this using modulation schemes which result in an abrupt change in frequency and phase while transmitting – examples of such schemes are DBPSK (differential binary phase shift keying), VMSK (very minimum sideband keying), minimum sideband (MSB) modulation etc. Although, this technology has been conceived in 1985, only now it has been practical as it requires RF filters with zero group delay [35], the implementation of which needs agile and sophisticated RF hardware components.

Technology Comparison: For a given output power (often defined by governmental regulations), the range of the RF link is determined by the data rate, i.e. lower rates provide longer ranges due to increased sensitivity for the receiver. There is, of course, also a trade-off, since very low rates mean a considerable long time of transmission, which in turn reduces battery lifetimes. Theoretically, given the same net data rate/throughput, systems based on both Ultra-Narrow-Band and Spread-Spectrum technologies can achieve similar ranges [33]. The drawback of ultra-narrowband technology are the higher requirements on the RF crystal, whereby a frequency error on the RF crystal leads to an offset on the programmed RF frequency. Therefore, large offsets may lead the signal to fall outside the channel and be filtered out by the receiver filters [33].

UNB technology promises better utilization of the spectrum in cases where implementation hurdles can be addressed. This utilization feature requires a relatively complex media access layer for handling the inter-channel interference and the coexistence with other RF devices in the same band. Additionally, the system needs to opportunistically manage a large number of channels in the same band due to the narrow channel-width in a volatile RF environment, particularly important in a city context.

On the other hand, spread-spectrum based devices are likely to under-utilize the spectrum relatively to what could be achieved with an ideal implementation of the Ultra-Narrow-Band technology. However, this simplifies other key issues such as interference and coexistence with other wireless devices as its signal appears as background noise to other systems. Other narrow-band devices, conversely, cause from low level to no interference with the spread spectrum systems because the correlation receiver effectively integrates over a wide bandwidth to recover a spread spectrum signal. This simplifies the implementation of the media-access layer as it involves only dealing with code-sequences from a user's point of view. In an urban space, this is expected to perform better as the spread spectrum signal is less susceptible to multi-path fading. One added inherent advantage that comes along with this technology, because of the available wide bandwidth, is the increased difficulty to jam, detect, intercept or demodulate the signals. Due to the pseudo-random nature of the code sequence, the signal in the air has been "randomized". Only the receiver having the exact same pseudo-random sequence, as has been predefined by the hardware vendor, can de-spread and retrieve the original signal. Consequently, a spread spectrum system provides signal reliability inherently.

Overall, from a long-term scalability perspective, Ultra-Narrow-Band seems to be more promising, however for early stage IoT applications, a more resilient technology will be crucial, which is likely to give advantage to spread spectrum based devices until Ultra-Narrow-Band devices mature to an acceptable level. In Chapter 3, we explore these hypothesis with regard to the technologies available now and with regard to the urban space and provide experimental evaluation results of these technologies and insight of their usage to Smart Water Networks.

LPWA technologies can enable the communication in city scale CPSs by providing lower energy

and less hardware infrastructure compared to other conventional communication solutions, such as WiFi or 4G which are inappropriate for deployments in harsh environments and low power hardware infrastructures. However, the main drawback of the LPWA technologies is the reduction of the communication data rates. The slow data rates may lead to higher energy consumption due to the long transmission time. Therefore, the incorporation of edge data reduction techniques with LPWA technologies is vital.

2.3 Data Reduction Techniques

A Wireless Sensor Network (WSN) consists of low power, inexpensive sensor nodes, which are responsible for retrieving data from one or more sensors, make computations, store data in local memory and send them to the network. The nodes of a WSN can be installed in a variety of environments (e.g., with high pressure, temperature, radiation and electromagnetic noise). However, node installation to harsh environments like underground or underwater [36], makes the access impossible. The majority of sensor nodes are battery-driven and consequently the per-node and the total reduction (especially in distributed architectures) of power consumption is vital to their lifetime.

The power consumption problem over a WSN can be encountered by: (a) optimizing the hardware power consumption (network card, memory, peripherals/sensors, CPU power consumption) or using energy harvesting [37], (b) optimizing network features (communication protocol, topology, radio selection), and (c) optimizing the software (minimizing sensor data computations, security algorithm complexity without making the system vulnerable, and wireless transmitted data size). This section provides a variety of data reduction techniques over WSN.

According to [38], the wireless transmission of a single bit can require over 1000 times more energy than a single 32-bit computation. Therefore, many researchers have approached the data reduction from many different points of view, by using: (a) probabilistic/machine learning techniques (data fusion, aggregation), and (b) data compression (lossless and lossy).

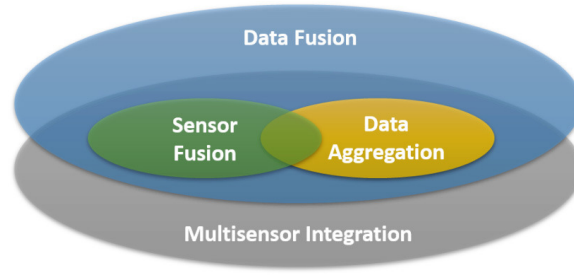


Figure 2.4: Data fusion terms.

Data fusion (Figure 2.4) is the process of gathering and transforming data into an accurate, consistent and useful representation by increasing the significance of such a mass of data [39]. In this process the system overcomes the sensor failures, technological limitations, and spatial and temporal coverage problems, while in WSN the main objectives are the improvement of accuracy and energy saving. Consistently with [39] and [40] the data fusion methods can be categorized by their objective in: inference, estimation, feature maps, abstraction sensors, aggregation and compression.

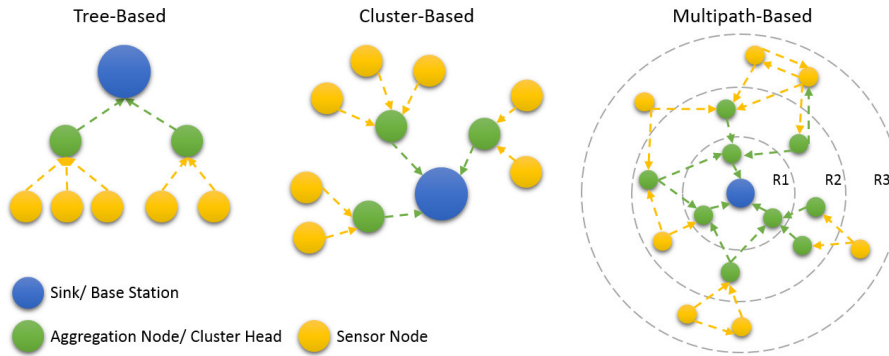


Figure 2.5: Aggregation routing approaches.

Data aggregation techniques are considered as the main data fusion method, which aims to solve the problem of data reduction in WSNs. These techniques consist of the routing protocols (tree-based, cluster-based, multipath-based and hybrid-based - Figure 2.5), the aggregation function (Temporal coherency-aware in-Network Aggregation (TiNA) [41], Data Aggregation and Dilution by Modulus Addressing (DADMA) [42], Adaptive Application-Independent data Aggregation (AIDA) [43], Synopsis Diffusion Framework [44], and the Quartile Digest [45]), and the data representation (this refers mostly to data compression). Routing protocols are responsible of selecting the next hop based on data content (that is they are data centric, e.g. position of the most suitable aggregation point, data type, priority of information), and

to provide timing strategies, while the aggregation functions fuse the data. The efficiency of these algorithms depends on the correlations among the data generated by different information sources. The correlation algorithms can be classified in spatial (the pattern of neighbor sensor readings are the same or similar), temporal (the pattern of future sensor readings is equal to or similar to that of previous sensor readings in a sensor), and semantic (correlating of the data reported by multiple sensors based on the semantics of the query).

2.3.1 Lossless & Lossy Compression

Until this point, we overviewed data fusion techniques and data aggregation methods (including routing approaches and data aggregation functions) and how to enhance them by using correlation algorithms. Furthermore, data reduction in WSN can be achieved by using appropriate data representations. More specifically, the meaning of data representation is how to organize the resulting data from the data fusion process in order to minimize the data size. Compression techniques are a solution to the representation problem. There are two different classes of compression techniques: (a) lossless, and (b) lossy.

Lossless compression occurs when compressed data can be perfectly reconstructed on decompression, for example the compression of a text or binary file. Figure 2.6 presents the four fundamental coding methods (Sliding Window - LZ77 [46], Dictionary - LZ78 [47], Prediction with Partial Match - PPM [48], and Burrows-Wheeler Transform (BWT) [49]), and the standard algorithms and applications which are based to these methods.

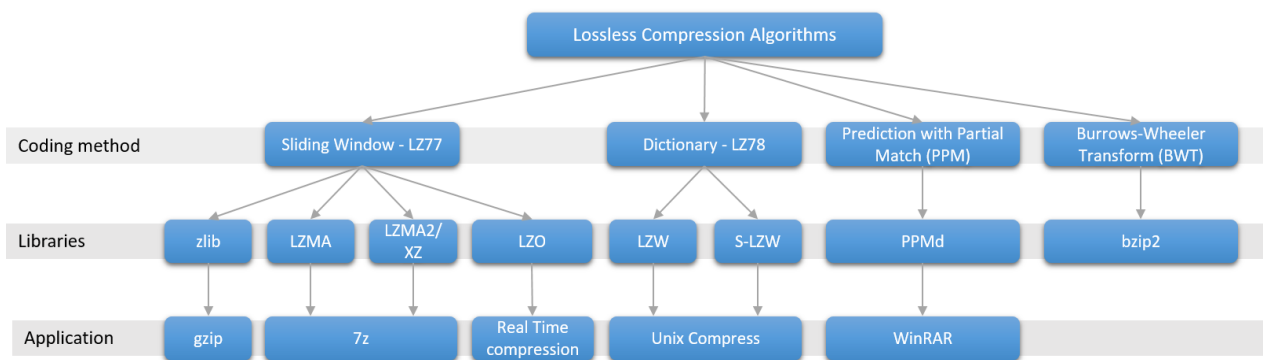


Figure 2.6: Lossless compression methods, libraries, and applications.

The selection of the appropriate lossless data reduction technique depends on the hardware limitations (i.e. memory size, cache availability, CPU computational power), power consumption per part of the node, the topology of the nodes (centric or distributed), the raw data types and content (i.e. bytes, text, web data), and the correlations between raw data. Thus, analysis, simulation, and evaluation are necessary to understand if the above techniques are applicable because the optimal compression techniques in the powerful computers is not necessarily feasible to the WSNs.

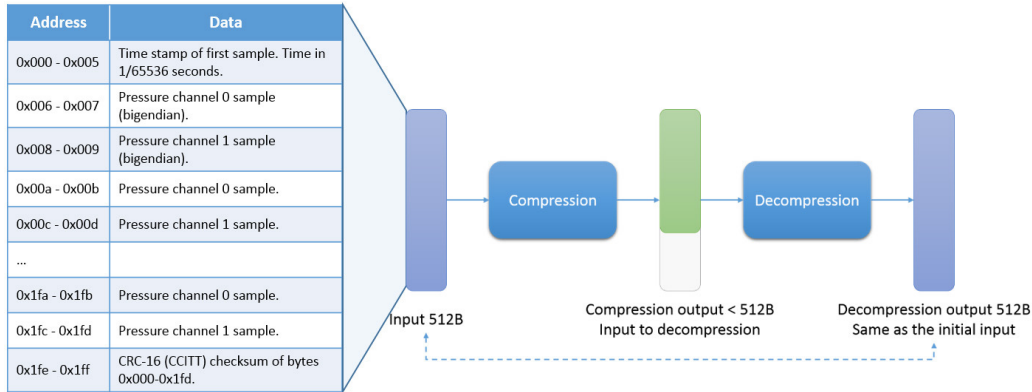


Figure 2.7: Lossless compression algorithms evaluation process.

By using Atmel Studio 6 simulator, we wrote a program to simulate the process described in Figure 2.7. Because contemporary Micro Controller Units (MCUs) [50] are able to handle the computation requirements of the aforementioned lossless compression algorithms, we decided to evaluate them only based on their memory requirements which constitute the only criteria for their applicability in embedded systems. The current system structures the data in 512 bytes and contains the timestamp of the current measurements, the pressure data of two input channels, and Cyclic Redundancy Check (CRC). This structure was used as input to compression algorithms which compressed the data to less than 512 bytes, then the data was decompressed and the final output was compared with the initial input.

Based on Figure 2.8, the quantitative results of memory usage per algorithm are the following: PPMd uses more memory than the other algorithms (10MB), bzip2 [51] requires around 9MB of memory and data segments of 100KB, zlib requires 120 KB, for LZMA [52] the lowest initial configuration values requires 6.5KB and the minimum dictionary size is 4096 B. In total, during the execution, the minimum memory usage can be 30KB. LZ4 requires at least 16 KB

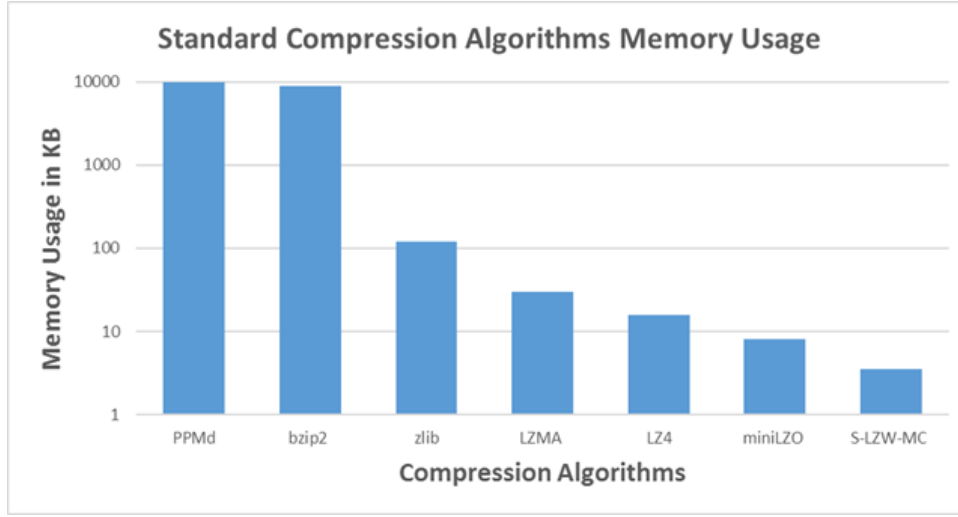


Figure 2.8: Standard compression algorithms memory usage.

of memory, miniLZO [53] requires memory only during the compression (dictionary creation) and the minimum memory requirement is 8.192KB, and the minimum required memory for S-LZW-MC [54] is 3.25KB. Furthermore, this algorithm is a lightweight embedded version of LZW, which requires less memory and computation power and, therefore, it is not necessary to benchmark this as well.

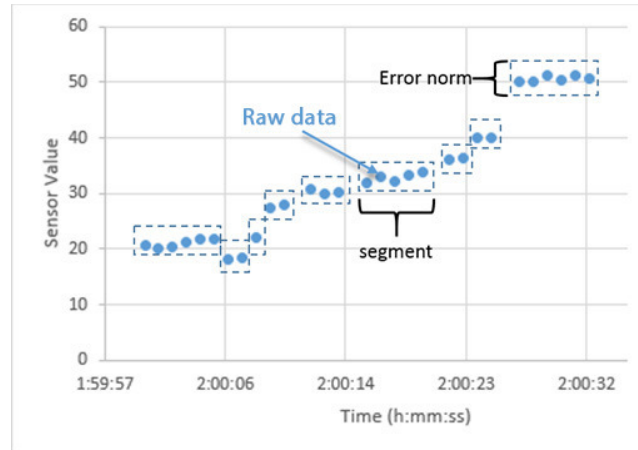


Figure 2.9: Lossy Compression - Raw data segmentation.

Additional to lossless compression, lossy compression constitutes another class of data reduction techniques. During a lossy compression process, some loss or degradation of the data can be tolerated, for instance in image, sound and video compression. Lossy compression often exploits spatial-temporal correlations to compress the data within a specific error norm (approximation error). Furthermore, dimensionality data reduction techniques like standard

orthogonal transformation methods (e.g., Fourier, wavelet transform) can be included with lossy compression. The main goal of lossy compression algorithms is to cluster raw data into the longest segments, based on an error norm (Figure 2.9). Instead of returning the raw data values, the algorithms (like Swing, Polynomial Regression, and MidRange) return the start and the end time of data segments, and a representative value (or other attributes like the slope of regression line) of this segment. The data segmentation, so-called quantization, introduces the error in lossy compression. Compressive sensing [55] constitutes another state-of-the-art lossy compression techniques which will be analyzed in later section.

Lossless compression techniques provide the most accurate sensor values compared to lossy compression techniques. However, according to [56], lossless compression techniques cause more energy and node resource consumption than lossy approaches for the following reasons: (a) lossless compression and decompression are usually more computationally intensive than lossy, (b) indexing cannot be applied in lossless compression, and (c) some lossless compression techniques (like gzip) achieves lower compression rates than lossy techniques (the specific evaluation result reveals that lossy performs two times better than lossless).

Overall, there is no perfect answer to which data reduction technique is the optimal or most appropriate solution in WSN. The reason is that data reduction techniques are strongly coupled with the application of WSN. For example, in centralized network approaches the most suitable routing and data aggregation algorithms are different from the distributed network approaches because in the centralized architecture the system has more information about the topology and correlation relationships between the sensor nodes. In addition, in applications with low precision tolerance, lossy compression could be inappropriate in spite of the data reduction efficiency. For all these reasons, before the selection of a specific technique, the WSN physical/hardware limitations, requirements, and architecture should be declared a-priori.

2.3.2 Compressive Sensing

For decades, the sampling process has been largely dominated by the classical Nyquist-Shannon theories. However, several studies have shown that many natural signals are amenable to

highly sparse representations in appropriate transform domains (e.g., wavelets and sinusoids) [57,58]. Compressive Sensing (CS) provides a powerful framework for simultaneous sensing and compression [55], enabling a significant reduction in the sampling and computation costs on a sensor node with limited memory and power resources.

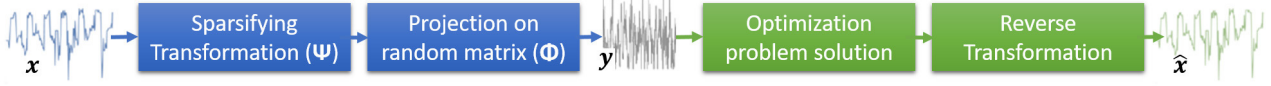


Figure 2.10: Compressive Sensing process.

According to the theory of CS (see Figure 2.10), a signal having a sparse representation in a suitable transform domain (e.g. FFT, Wavelet, Cosine) can be reconstructed from a small set of incoherent random projections. Specifically, let consider a finite length, discrete time signal $\mathbf{x} \in \mathbb{R}^N$ and a matrix $\Psi \in \mathbb{R}^{N \times L}$ whose columns correspond to a *transform basis*², in general overcomplete (i.e. $N \leq L$).

Definition 2.1. *Sparsity* of a signal is defined as the number of zero-valued (or smaller than a certain minor threshold) elements divided by the total number of elements in the signal under some representation.

The signal $\mathbf{x}$ has a sparse representation over a known Ψ , if a sparse coefficient vector $\mathbf{w} \in \mathbb{R}^L$ exists such that $\mathbf{x} = \Psi^T \mathbf{w}$, with $\|\mathbf{w}\|_{l_0} = K \ll L$, where $\|\cdot\|_{l_0}$ indicates the non-zero elements in $\mathbf{w}$ and Ψ^T denotes the inverse transform. Based on CS, an unknown vector $\mathbf{x}$ is recovered from an underdetermined system of linear equations

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi^T \mathbf{w}, \quad (2.1)$$

where $\mathbf{y} \in \mathbb{R}^M$ is the vector of compressive measurements and $\Phi \in \mathbb{R}^{M \times N}$ with $K < M < N$ is the sensing or measurement matrix. The sample rate SR can be derived from Φ by calculating the following trace:

$$SR = Tr(\Phi^T \Phi) \quad (2.2)$$

²Alternatively, in the literature of CS (i.e. [59]), researchers refer to Ψ as *dictionary* and define it as a set of elementary waveforms used to decompose the signal.

Due to the fact that $M < L$, the system (2.1) is not invertible and has infinitely number of solutions from which the sparsest are the most important. However, based on the proofs in [17], direct minimization of $\|\mathbf{w}\|_{l_0}$ is an NP-hard problem and therefore all the l_p -norms for $p \geq 1$ are special cases of interest for convexity. Particularly, l_1 -norm is the most popular having the tendency to sparsify the solution. Based on [60], a sparse signal $\mathbf{w}$ can be recovered from only $M = \mathcal{O}(K \log(\frac{N}{K}))$ non-adaptive linear projections onto a second measurement basis by solving the following relaxation

$$\min_{\mathbf{w} \in \mathbb{R}^L} \|\mathbf{w}\|_{l_1} \text{ subject to } \|\mathbf{y} - \Phi \Psi^T \mathbf{w}\|_{l_2} \leq \epsilon, \quad (2.3)$$

where $\|\mathbf{w}\|_{l_1} = \sum_{i=0}^L |\mathbf{w}_i|$ and $\epsilon > 0$. Solving the (2.3) optimization problem, the estimated sparse coefficient vector $\hat{\mathbf{w}}$ can be used to reconstruct the original signal by simply taking the inverse transformation

$$\hat{\mathbf{x}} = \Psi^T \hat{\mathbf{w}}. \quad (2.4)$$

In the literature, two main approaches have been used to solve the optimization problem in (2.3). The first approach, the so-called Basis Pursuit (BP) techniques [17,60,61] transform the problem to Linear Program (LP) [62]. In the second approach, the Matching Pursuit (MP) greedy methods [63,64] use an iterative way to solve the optimization problem.

The number of the required measurements to reconstruct a sparse signal is closely affected by the relationship of measurement matrix Φ and transform basis Ψ . In CS, the theories follow the assumption that $\mathbf{D} = \Phi \Psi^T$ is the concatenation of two orthogonal matrices and the principle that a signal cannot be sparsely represented both in Φ and Ψ [65]. This latter claim depends on the similarity between Φ and Ψ .

Definition 2.2. *Mutual coherence* represents the degree of similarity between the measurement matrix Φ and transform basis Ψ

The mutual coherence can be calculated as follow:

$$\mu(\Phi, \Psi) = \max_{\substack{i,j \\ i \neq j}} \frac{|\phi_i^H \psi_j|}{\|\phi_i\|_{l_2} \cdot \|\phi_j\|_{l_2}} \quad (2.5)$$

If Φ and Ψ contain correlated elements, the mutual coherence is large, otherwise small. Based on [66], the mutual coherence has to be within a range defined by linear algebra as $\mu(\Phi, \Psi) \in [1, \sqrt{N}]$.

In addition to mutual coherence, the usual criterion to qualify the Φ and Ψ matrices for recovering a sparse signal is the **Restricted Isometry Property (RIP)** [61]. Formally, matrix $\mathbf{D}$ satisfies RIP of order S with a restricted isometry constant $\delta_S \in (0, 1)$ being the smallest value such that

$$(1 - \delta_S) \|\mathbf{w}\|_{l_2}^2 \leq \|\mathbf{D}\mathbf{w}\|_{l_2}^2 \leq (1 + \delta_S) \|\mathbf{w}\|_{l_2}^2, \quad \forall \mathbf{x} \text{ with } \|\mathbf{x}\|_{l_0} \leq S \quad (2.6)$$

where $\|\cdot\|_{l_2}$ represents the l_2 -norm i.e. $\|\mathbf{w}\|_{l_2} = \sqrt{\sum_i w_i^2}$.

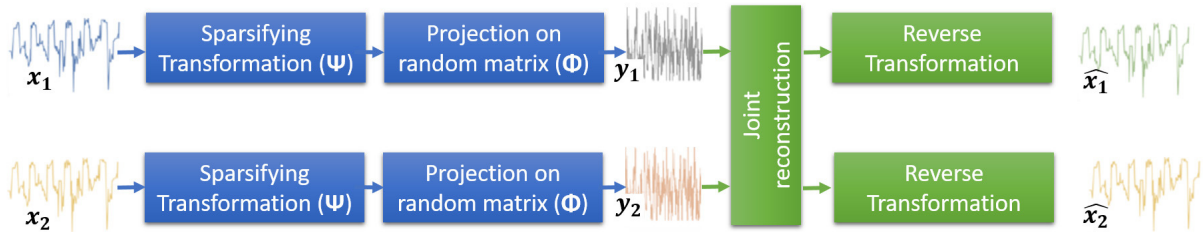


Figure 2.11: Distributed Compressive Sensing process.

CS has been used widely in multiple domains including image and video processing [67–69], communication and networking [70–73], and biological applications [74]. In spite the benefits of CS, data correlation among different data sources has not been considered. In order to exploit data correlation and to improve the reconstruction performance, Distributed Compressive Sensing (DCS) was introduced by Baron et al. [75]. DCS integrates the Distributed Source Coding (DSC) [76] and CS. Similarly to CS, in DCS (see Figure 2.11) each sensor node compresses the signal independently but the signal is being reconstructed jointly to the base station/ fusion center. Formally, based on DCS and Figure 2.11, each unknown vector $\mathbf{x}_i$, $i \in P$, where P is the number of the streams, is recovered from an underdetermined system of linear equations

$$\mathbf{y}_i = \Phi_i \mathbf{x}_i = \Phi_i \Psi_i^T \mathbf{w}_i, \quad (2.7)$$

where $\mathbf{y}_i \in \mathbb{R}^M$ is the vector of the i -th sensor node compressive measurements. Thus, all

unknown vectors $\mathbf{x}_i, \forall i \in P$, are recovered from

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} = \mathbf{\Phi}\mathbf{\Psi}^T\mathbf{W} = \mathbf{D}\mathbf{W} \quad (2.8)$$

where $\mathbf{Y} \in \mathbb{R}^{M \times P}$, $\mathbf{X} \in \mathbb{R}^{N \times P}$, and $\mathbf{W} \in \mathbb{R}^{L \times P}$ matrices with their columns correspond to the compressive measurements, actual measurements, and sparse coefficients respectively. The $\mathbf{\Phi} \in \mathbb{R}^{MP \times NP}$ and $\mathbf{\Psi}^T \in \mathbb{R}^{NP \times LP}$ are diagonal matrices which contain the measurement matrices $\mathbf{\Phi}_i, \forall i \in P$ and transform basis matrices $\mathbf{\Psi}_i^T, \forall i \in P$ correspondingly, as follows

$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{\Phi}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{\Phi}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Phi}_P \end{pmatrix}, \quad \mathbf{\Psi}^T = \begin{pmatrix} \mathbf{\Psi}_1^T & 0 & \cdots & 0 \\ 0 & \mathbf{\Psi}_2^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Psi}_P^T \end{pmatrix}$$

The matrix $\mathbf{D} \in \mathbb{R}^{MP \times LP}$ combines measurement and transform basis matrices

$$\mathbf{D} = \mathbf{\Phi}\mathbf{\Psi}^T = \begin{pmatrix} \mathbf{\Phi}_1\mathbf{\Psi}_1^T & 0 & \cdots & 0 \\ 0 & \mathbf{\Phi}_2\mathbf{\Psi}_2^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Phi}_P\mathbf{\Psi}_P^T \end{pmatrix}$$

Similar to CS, the sparse signal $\mathbf{W}$ can be recovered by solving the following relaxation

$$\min_{\mathbf{W} \in \mathbb{R}^{LP}} \|\mathbf{W}\|_{l_1} \text{ subject to } \|\mathbf{Y} - \mathbf{D}\mathbf{W}\|_{l_2} \leq \epsilon, \quad (2.9)$$

Solving the (2.9) optimization problem, the estimated sparse coefficient matrix $\widehat{\mathbf{W}}$ can be used to reconstruct the original signals by simply taking the inverse transformation

$$\widehat{\mathbf{X}} = \mathbf{\Psi}^T\widehat{\mathbf{W}}. \quad (2.10)$$

Based on [60], in CS, the selection of $\mathbf{\Phi}$ and consequently the sample rate can be determined

by the sparsity of the signal. In DCS, the selection of Φ_i is not a trivial problem. In the literature [77, 78], the notion of Mean Squared Error (MSE) has been used to define the bounds of Φ_i . For example, in [78], the authors define Φ_i based on the following upper bound of MSE:

$$\sum_{\mathbf{J} \in \bar{\mathbf{S}}} Pr(\mathbf{J}) Tr((\mathbf{D}^T \mathbf{D})^{-1}) \leq \alpha \quad (2.11)$$

where $\bar{\mathbf{S}}$ is the set of all support patterns, $\mathbf{J}$ is the support of $\mathbf{X}$ with probability $Pr((J))$, and $\alpha > 0$ consists of an application-specific parameter .

DCS provides two main benefits compared to classic CS: it decreases dramatically the number of required measurements, achieving low reconstruction error by exploiting data correlation from different node; and it transfers the complexity from edge to the base station, which is critical in resource constrained wireless sensor networks.

In this thesis, Chapter 4 presents analytically the benefits of both lossless compression and lossy compression through compressive sensing in Smart Water Networks. Further, in Chapter 5, we describe a way to combine the DCS with our smart scheduling system which guarantees sustainable CPSs by exploiting Lyapunov Drift-Plus-Penalty method.

2.4 Lyapunov Drift-Plus-Penalty

Lyapunov Drift-Plus-Penalty method [19] has been widely used to optimize queueing networks and other stochastic systems [79–82]. This technique has the ability to stabilize the length of the queues along a network while minimizes (or maximizes) the time average of a network penalty function. The main benefit of this approach is that the algorithm requires **only** the current state of the system which can be obtained easily at real time. Furthermore, there is **no need** of prediction or probabilistic approaches.

In order to make the above claims clearer, we present a simplified version of the example in [83] adjusted to a wireless sensor network scenario. Before the description of the problem, some basic knowledge related to queueing theory is necessary. Figure 2.12 illustrates the two main processes of sensing and transmission which are conducted in a sensor node i belonging to a network

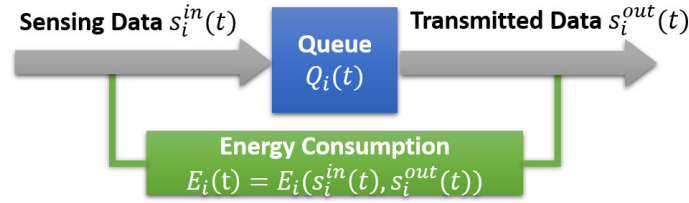


Figure 2.12: Sensor node operation example for Lyapunov Drift-Plus-Penalty method.

with size N number of nodes. At every discrete time t , the sensor node retrieves a number of measurements $s_i^{in}(t)$ from the sensors and store them to the queue of the communication module which transmits a number of measurements $s_i^{out}(t)$ over the network. As a result, at time t , the new queue length $Q_i(t+1)$ can be derived as follows:

$$Q_i(t+1) = \max\{Q_i(t) + s_i^{in}(t) - s_i^{out}(t), 0\}$$

or

$$Q_i(t+1) = \max\{Q_i(t) + y_i(t), 0\}, \text{ where } y_i(t) = s_i^{in}(t) - s_i^{out}(t) \quad (2.12)$$

Theorem 2.1. *The queue length is rate stable if $\lim_{t \rightarrow \infty} \frac{Q_i(t)}{t} = 0$ (with probability 1). This implies that the queue length is also rate stable if the time average*

$$\bar{y}_i(t) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} y(\tau) \leq 0$$

Proof. Theorem 2.1 can be proved by using telescoping sum, which can be found in Appendix A. □

By having declared the stability requirements of the queue, the next step is to describe a problem that can be solved by using Lyapunov Drift-Plus-Penalty. Following the example of Figure 2.12, let assume that exists a function $E_i(s_i^{in}(t), s_i^{out}(t))$ that describes the relationship among $s_i^{in}(t)$ and $s_i^{out}(t)$ and energy consumption of the i -th sensor node. The global energy consumption of the network at time t can be derived by $E(t) = \sum_{i \in N} E_i(s_i^{in}(t), s_i^{out}(t))$. Then, a possible problem could be:

*"Select optimal $s_i^{in}(t)$ and $s_i^{out}(t)$, $\forall i \in N$ that stabilize the queue length $Q_i(t)$, $\forall i \in N$ **and** minimize the time average of global energy consumption $\bar{E}(t) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i \in N} E_i(s_i^{in}(\tau), s_i^{out}(\tau))$ "*

along the network."

Formally, the optimal $s_i^{in}(t)$ and $s_i^{out}(t)$ at every interval t can be obtained by solving the following stochastic optimization problem:

$$\mathbf{min} \quad \bar{E}(t) \quad (2.13)$$

$$\mathbf{subject\ to} \quad \bar{y}_i(t) \leq 0, \forall i \in N \quad (2.14)$$

$$E_i(s_i^{in}(t), s_i^{out}(t)) \text{ in } S, \forall i \in N, \forall t \quad (2.15)$$

where S is a known set. Even if the problem is formalized properly according to queueing theory, based on [83], classic optimization problem solvers (i.e. linear programming) have significant disadvantages. For example, they cannot guarantee delays, support flow control and adaptability to changes. Lyapunov Drift-Plus-Penalty approach transforms the current problem, within five steps, to a new optimization problem that overcomes these drawback:

Step 1: Given Theorem 2.1, for each inequality constraint (i.e. constraint (2.14)), make a "virtual queue": $Q_i(t+1) = \max\{Q_i(t) + y_i(t), 0\}, \forall i \in N$

Step 2: Define the sum of squares of queues times $\frac{1}{2}$: $L(t) = \frac{1}{2} \sum_{i=1}^N Q_i(t)^2$

Step 3: Define the "drift" of $L(t)$: $\Delta L(t) = L(t+1) - L(t)$

Step 4: For every time t , observe $s_i^{in}(t)$, $s_i^{out}(t)$, and $Q(t)$. Then choose $E(t)$ in S to greedily minimize the Drift-Plus-Penalty equation: $\Delta L(t) + VE(t)$

Step 5: Define the Drift-Plus-Penalty bounds. Re-writing Equation (2.12) as inequality:

$$Q_i(t+1)^2 \leq [Q_i(t) + y_i(t)]^2 = Q_i(t)^2 + y_i(t)^2 + 2Q_i(t)y_i(t) \quad (2.16)$$

Assuming that squares of $y_i(t)^2$ are bounded, inequality (2.16) becomes:

$$\begin{aligned} \Delta L(t) &\leq B + \sum_{i=1}^N Q_i(t)y_i(t) \\ \Delta L(t) + VE(t) &\leq B + \sum_{i=1}^N Q_i(t)y_i(t) + VE(t) \end{aligned} \quad (2.17)$$

Then, the initial problem can be simplified and transformed to:

$$\min \sum_{i=1}^N Q_i(t)y_i(t) + VE(t) \quad (2.18)$$

$$\text{subject to } E_i(s_i^{in}(t), s_i^{out}(t)) \text{ in } S, \forall i \in N, \forall t \quad (2.19)$$

The new optimization problem (2.18)-(2.19), as derived based on Lyapunov-Plus-Drift method, stabilizes the queues and minimizes the energy consumption of the system requiring only the current state of the system. The Lyapunov Drift-Plus-Penalty method is used by the smart scheduling system in Chapter 5 to decide which node has to transmit data in order to stabilize the battery level (i.e. the virtual queue) of all the sensor nodes of a smart water network and maximize the data estimation reliability at the same time. The next vital decision in CPSs is related to when the sensor nodes have to communicate in order to ensure the stability of the physical layer. Event-triggered control is exploited to answer this question.

2.5 Event-triggered Control

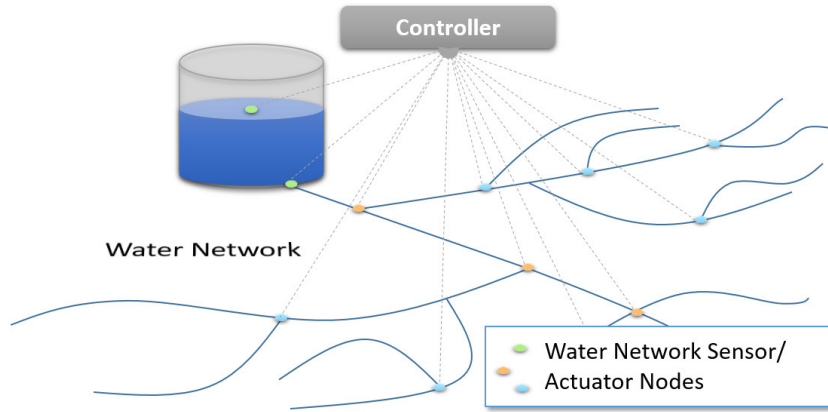


Figure 2.13: Ideal smart water network.

An ideal automatic water distribution network (i.e. Figure 2.13) consists of sensor and actuator nodes installed in different water network asset (e.g. valves, pressure regulators). The sensor nodes are responsible for transmitting the measurements to a central point or controller (e.g. a server or base station) and the controller has to calculate the state of the water network and make decisions about the control signal that has to send to the actuators. The purpose of the

control signals is to ensure the stability of the water network by leading the physical layer to converge to a certain point or state, so-called stable state. To this end, the most important task is to define automatic control scenarios and stability concepts under the context of water networks. The simplest scenario of an automatic control concept might be the following: *"Maintain the water supply in a specific level in order to cover water demand of each area, subject to the maximum allowed water pressure and flow to avoid leakages or bursts."*

After the definition of stability and control scenario, the next task in classic control systems is to specify an appropriate inter-sampling period of the water network state. This task is critical because if the controller delays to detect instabilities in the water network and actuate the appropriate valves under certain time constraints then it can cause leakages or burst. Additionally, if the controller overuses the communication network to receive measurements more often, it can drain the battery of sensor nodes. For that reason, engineers still rely mostly on rules of thumb [84–86] such as sampling with a frequency 20 times the system bandwidth, and then check if it actually works. While this solution is quite straightforward, a conservative choice of inter-sampling period may result in significant over-provisioning of the communication network. For example, sensor nodes have to transmit their measurements periodically even if an area of water network is stable with small variation in the values.

In order to avoid the waste of communication, event-triggered control notion was introduced [87,88]. In event-triggered control, instead of periodically updating the state and control input, the update instants are generated by the violation of a condition on the state of the plant. Many researchers have shown a renewed interest on these techniques [22,89–95] by establishing event-trigger control (i.e. [92]) and self-triggered control (i.e. [93]). In event-triggered control, the system continuously observes the state of the plant and generates control signals only in the case of a state violation. on the other hand, a self-trigger control system predicts when is the next violation time and generates the control signal at this pre-decided time. Within the context of this thesis, we will focus in event-trigger control which is more sensitive to system disturbances that are critical to water networks. The following section presents an overview of the process of designing event-triggered control systems.

2.5.1 Centralized Event-triggered Control

In event-triggered control systems, a sensor transmits when some internal measure of the novelty in the sensor information exceeds a specified threshold. In particular, this means that when the gap between the current $x(t)$ and the j -th more recently transmitted sensor state $x(r_j)$ exceeds a state-dependent threshold e_T , then the information is transmitted across the channel:

$$gap = |x(t) - x(r_j)| \leq e_T = threshold$$

The state-dependent thresholds are chosen in a way that preserves commonly used mathematical stability concepts such as input-to-state stability [95] or $\mathcal{L}_2$ [96] stability. This approach for threshold selection therefore provides a systematic way of triggering transmissions that provides some guarantees on the overall control system performance.

Stability concepts define mathematically when systems reach the equilibrium points by creating state boundaries (inequalities) and they are fundamental elements of event-triggered decision. The selection of a stability concept for a control problem depends on the existence of external disturbance in controlled systems. Based on that, the systems are divided into two categories, homogeneous without an external disturbance and inhomogeneous with an external disturbance. In homogeneous systems, the state trajectories $x : \mathbb{R} \rightarrow \mathbb{R}^n$ satisfy the initial value problem for almost all $t \geq 0, t \in \mathbb{R}$:

$$\dot{x}(t) = f(x(t)), \quad x(0) = x_0$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is Lipschitz continuous, and the corresponding stability concepts are the Lyapunov and Asymptotic stability. The equilibrium point is Lyapunov stable if there always exists an initial condition that permits us to confine the system state within an arbitrarily small neighborhood of the equilibrium point:

Equilibrium point $\bar{x} = 0$ is stable,

$$if \forall \epsilon > 0, \text{ exists } \delta > 0 \text{ such that } \forall t > 0, \|x(0)\| < \delta \implies \|x(t)\| < \epsilon$$

An equilibrium point is said to be asymptotically stable if the point is Lyapunov stable and if the state $x(t)$ asymptotically approaches the equilibrium point as it goes to infinity. The most important tool that provides sufficient conditions for Lyapunov and asymptotic stability is the Lyapunov function. A continuous differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is Lyapunov if:

- V is a positive definite function and
- $\dot{V} = \frac{\partial V}{\partial x} f(x)$ is negative semi-definite (if negative definite then asymptotic stable)

In inhomogeneous systems with an external disturbance, the state trajectories $x : \mathbb{R} \rightarrow \mathbb{R}^n$ satisfy the initial value problem for almost all $t \geq 0, t \in \mathbb{R}$:

$$\dot{x}(t) = f(x(t), w(t)), \quad x(0) = x_0$$

where external disturbance $w : \mathbb{R} \rightarrow \mathbb{R}^m$ and $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, maps the current system state, $x(t)$, and an external disturbance, $w(t)$, onto the state's time derivative and the corresponding stability concepts are the Input-to-State (ISS) and $\mathcal{L}_2$.

Definition 2.3. $\mathcal{L}_\infty$ is the linear space of all measurable functions with bounded supremum norm $\|w\|_{\mathcal{L}_\infty} = \text{ess sup}_t \|x(t)\|$

Definition 2.4. A function $\alpha : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is said to be class $\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$. A function $b : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is said to be of class $\mathcal{KL}$ if it is a continuous function that is class $\mathcal{K}$ with respect to the first argument and decreasing asymptotically to zero with respect to the second argument

An inhomogeneous system is ISS if there exists a class $\mathcal{KL}$ function and a class function such that for an initial condition, $f(0) = x_0$, the response under any input $w \in \mathcal{L}_\infty$ satisfies:

$$\|x(t)\| \leq \beta(\|x_0\|, t) + \gamma(\|w\|_{\mathcal{L}_\infty})$$

or alternatively

$$\|x(t)\| \leq \max \{ \beta(\|x_0\|, t), \gamma(\|w\|_{\mathcal{L}_\infty}) \}, \forall t \geq 0$$

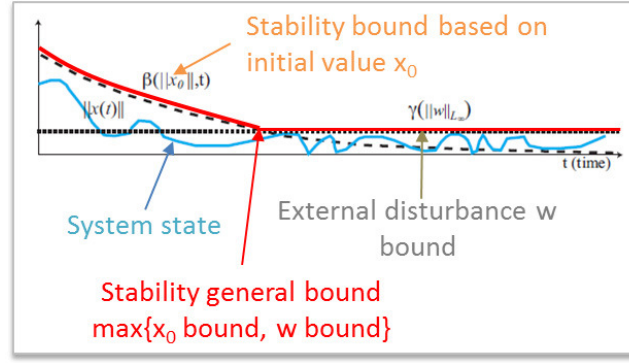


Figure 2.14: Input to State Stability bounds.

Figure 2.14 presents a more practical representation of the created bound by using the Input to State Stability (ISS) concept. Based on this image, the general bound is the maximum bound (red line) that is created by the bounds of initial values and the external disturbance.

The most important tool that provides sufficient conditions for ISS stability is the ISS-Lyapunov function. A continuous differentiable function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is an ISS-Lyapunov function if there exist class $\mathcal{K}$ functions $\bar{\alpha}$, $\underline{\alpha}$, γ , and β such that:

- $\bar{\alpha}(\|x\|) \leq V(x) \leq \underline{\alpha}(\|x\|)$
- $\dot{V}(x, w) \leq -\gamma(\|x\|) + \beta(\|w\|)$ where $x : \mathbb{R} \rightarrow \mathbb{R}^n$ and $w : \mathbb{R} \rightarrow \mathbb{R}^m$

$\mathcal{L}_2$ stability is another useful stability concept for inhomogeneous systems. In this case, one usually thinks of the system as a mapping, $G : \mathcal{L}_2 \rightarrow \mathcal{L}_2$ between two normed linear spaces, $\mathcal{L}_2$. This means that if one is given an input $\mathcal{L}_2$ then the system's output function Gw will also be a function in $\mathcal{L}_2$. The system G is finite gain $\mathcal{L}_2$ stable (or just $\mathcal{L}_2$ stable) if there exist finite positive real constants γ and β such that:

$$\|Gw\|_{\mathcal{L}_2} \leq \gamma\|w\|_{\mathcal{L}_2} + \beta$$

The right-hand side of the above inequality represents an affine function that over bounds the norm of the actual system's output. In particular, one can think of γ as a gain and β as an offset or bias. The so-called induced gain of G is then taken as the greatest lower bound on all of the possible γ 's for which the above inequality holds. This induced gain is often denoted as

$\|G\|$ and can be formally defined as:

$$\|G\| = \inf\{\gamma \in \mathbb{R} : \|Gw\|_{\mathcal{L}_2} \leq \gamma\|w\|_{\mathcal{L}_2} + \beta\}, \forall w \in \mathcal{L}_2$$

The induced gain provides an important way of defining a control system's performance. Many control synthesis problems can be formulated as so-called regulator problems in which the objective is to minimize the gain from the closed-loop system's uncontrolled external input to some output function. By making the induced gain of the closed-loop system sufficiently small, one provides some guarantees on the control system's performance level. The induced gain therefore becomes a direct way of the overall control system performance characterization. When the inhomogeneous system has a special affine form then there is a useful characterization of the $\mathcal{L}_2$ induced gain. This characterization can be used to design event-triggered systems that enforce the $\mathcal{L}_2$ stability concept.

The design of event-triggered control systems consists of two main steps. The first step is the design of a continuous-time controller that guarantees a stability concept such as Input-to-State (ISS) or $\mathcal{L}_2$ stability. The second step is the development of an event-triggered threshold such that the associated sporadically triggered control system preserves underlying stability concepts.

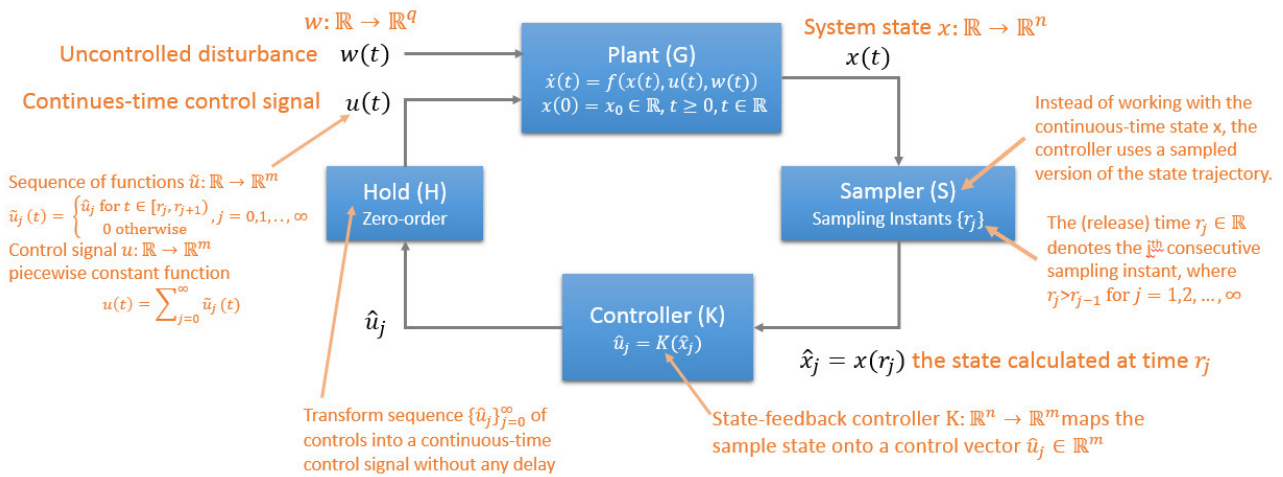


Figure 2.15 shows the design of a control system in an embedded system. In this system, the plant takes as input a control signal and external disturbance, and adjusts its physical state based on these inputs. Sampler is responsible to sample the state of the system. In the

case of a classic control system, the sampler samples the state in a periodic manner, while in event-trigger system the sampler re-computes the state after a state threshold violation. Then, a controller calculates the control signal based on sampled state of the system and Hold component transforms the discrete control signals into continuous. During the design phase, the state calculation function, control function, and stability concept bounds have to be defined.

The second phase of event-triggered control systems design is to create a threshold (inequality) which relies on the system's stability concept. The following threshold generation methodology has been described in [22]. Let's assume controller K enforces an ISS stability concept and system $\dot{x}(t) = f(x(t), K(x(t) + e(t)), w(t))$ where $e : \mathbb{R} \rightarrow \mathbb{R}^n$ and $w : \mathbb{R} \rightarrow \mathbb{R}^m$ are $\mathcal{L}_\infty$ input disturbance. Because the system is ISS, there exists an ISS-Lyapunov function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that:

$$\bar{\alpha}(\|x\|) \leq V(x) \leq \underline{\alpha}(\|x\|)$$

$$\dot{V}(x, K(x + e), w) = \frac{\partial V}{\partial x} f(x, K(x + e), w) \leq -\gamma(\|x\|) + \beta_1(\|e\|) + \beta_2(\|w\|)$$

Let's consider the sampled-data version where the sampler calculates the state every r_j , $\hat{x}_j = x(r_j)$. The gap function associated with the j th sampling time is a function $e_j : [r_j, r_{j+1}) \rightarrow \mathbb{R}^n$ in which:

$$e_j(t) = \hat{x}_j - x(t) \implies \hat{x}_j = x(t) + e_j(t),$$

for $t \in [r_j, r_{j+1})$ and $j = 0, 1, \dots, \infty$. So, the system's state must satisfy:

$$\dot{x}(t) = f(x(t), K(x(t) + e_j(t)), w(t)), \forall t \in [r_j, r_{j+1}) \text{ and } j = 0, 1, \dots, \infty$$

Under the ISS assumption, one knows that:

$$\dot{V} \leq -\gamma(\|x\|) + \beta_1(\|e\|) + \beta_2(\|w\|) \quad (2.20)$$

Let's assume the gap can be restricted so that for some $\sigma \in (0, 1)$:

$$\beta_1(\|e_j\|) \leq \sigma\gamma(\|x(t)\|), \forall t \geq 0 \text{ and } j = 0, 1, \dots, \infty \quad (2.21)$$

Inequality (2.21) can be used as a state-dependent threshold that forces the sampler to recompute the state of the system. At the beginning of each sampling period $[r_j, r_{j+1})$, the gap $e_j(r_j) = 0$. After that, the gap increases until the violation of the inequality. Then the sampler generates the new state and zeroes the gap again. Figure 2.16 presents the calculation of the gap and the event-triggered threshold violation.

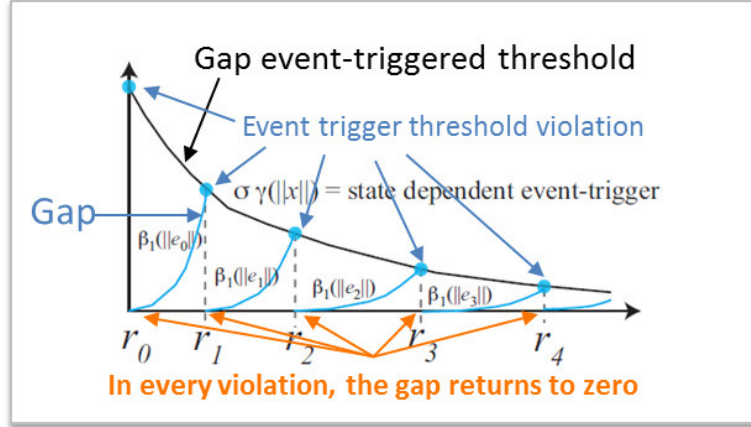


Figure 2.16: Event-triggered threshold violation.

From the Inequalities (2.20) and (2.21) implies the ISS-Lyapunov function declaration which ensures the stability of the system:

$$\dot{V} \leq -(1 - \sigma)\gamma(\|x\|) + \beta_2(\|w\|)$$

2.5.2 Decentralized Event-triggered Control

Based on the previous section, the Event-Triggered Control (ETC) threshold relies on the full state of the system. In the case of water network that means that all the sensors nodes transmit their measurements to a central controller and the central controller is responsible to calculate the new state of the system and to generate the control signals only in the case of a state threshold violation. As soon as, all the sensor nodes have to transmit their local states to the central controller, the system reduces only the actuations and not the communication. In order to reduce the communication by fully exploiting ETC technique, Mazo and Tabuada [94] and Lemmon [97] proposes a decentralized versions. In the decentralized ETC, each sensor node has each own event-triggered mechanism based on the local state. In the case of Lemmon, in

a local threshold violation, the sensor nodes broadcast their measurements to the correlated actuator nodes, while in Mazo and Tabuada, the sensor nodes transmit their measurements to a central controller and the central controller transmits the control signal to actuators.

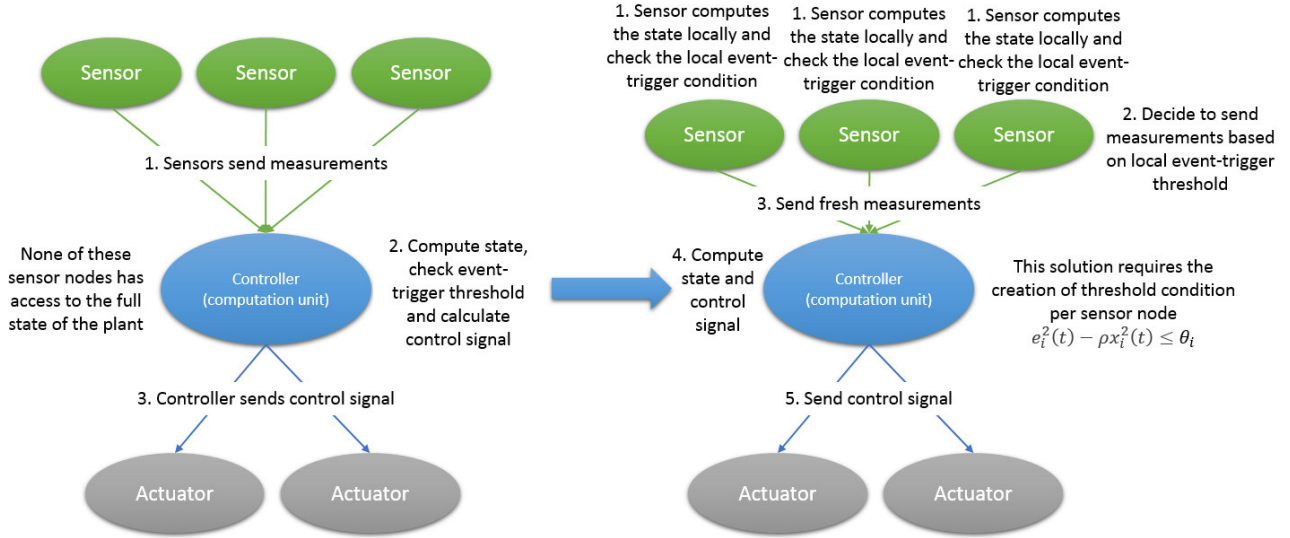


Figure 2.17: Centralized to decentralized event-triggered control systems.

Figure 2.17 illustrates the conversion of a centralized event-trigger control system to decentralized version based on [94]. The authors of this paper present an ISS system and assume that σ^{-1} and γ are Lipschitz continuous. This assumption allows simplifying the event-triggered threshold:

$$\beta(\|e(t)\|) \leq \sigma\gamma(\|x(t)\|) \implies \|e(t)\|^2 \leq \rho\|x(t)\|^2, \forall t \geq 0 \text{ and } \rho > 0$$

That implies:

$$\|e(t)\|^2 \leq \rho\|x(t)\|^2 \Leftrightarrow \|e(t)\|^2 - \rho\|x(t)\|^2 \leq 0 \Leftrightarrow \sum_{i=1}^n (e_i^2(t) - \rho x_i^2(t)) \leq 0$$

Assume parameters $\theta_1, \theta_2, \dots, \theta_n \in \mathbb{R}$ such that $\sum_{i=1}^n \theta_i = 0$ then:

$$\sum_{i=1}^n (e_i^2(t) - \rho x_i^2(t)) \leq \sum_{i=1}^n \theta_i$$

Where e_i and x_i the i th coordinates of e and x . Because it holds:

$$\bigwedge_{i=1}^n (e_i^2(t) - \rho x_i^2(t) \leq \theta_i) \implies \|e(t)\|^2 \leq \rho(\|x(t)\|^2)$$

the local event-triggering conditions is:

$$e_i^2(t) - \rho x_i^2(t) \leq \theta_i \tag{2.22}$$

After the definition of the local event-triggered threshold, the authors propose a heuristic approach by using Taylor expansion order to calculate the θ_k parameters, in order to make all the local inequalities to be violated at the same time. Because of the θ_k parameters, the local event-triggered thresholds are more conservative than the centralized threshold. In every violation of the local threshold the system supports on-line adaptation and re-calculates the θ_k parameters. The experiments in [94] reveal impressive savings by reducing the transmissions by 85% and the power consumption by 41% than the periodic state sampling approach.

This section describes the fundamental theory behind the basic scenarios of centralized and decentralized event-trigger control which will be analyzed further in Chapter 6 by introducing periodic and asynchronous event-triggered control and innovative MAC protocols to accommodate the event-based communication in CPSs.

2.6 Summary

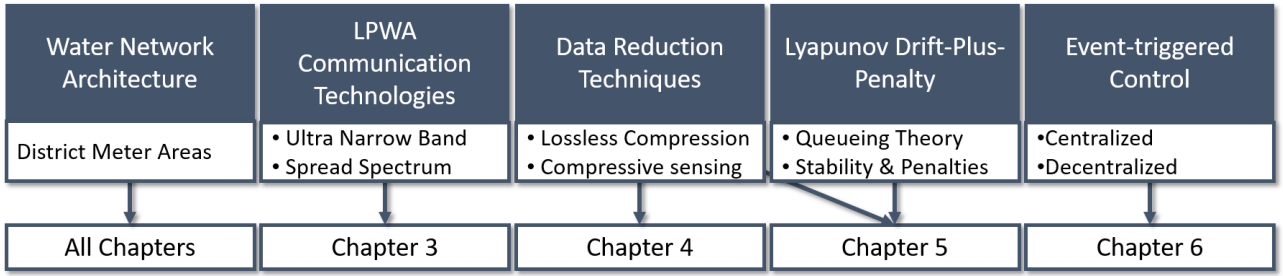


Figure 2.18: Background knowledge and chapter assignment.

This chapter briefly describes the background knowledge which is required for the corresponding chapters of this thesis (see Figure 2.18). In order to clarify the physical constraints of our system, firstly, we introduce the DMA-based architecture of contemporary smart water networks which enables resilient monitor and control of the water distribution. Having the physical constraints, we describe LPWA communication technologies, which are based on Ultra Narrow Band and Spread Spectrum, as the main candidates to accommodate long range and low energy communication over the harsh environments of water networks. Due to the LPWA low data rates, data reduction techniques are unavoidable. Therefore, in this chapter we present an overview of these techniques over wireless sensor networks focusing on reliable lossless and efficient lossy compression. Under the context of lossy compression, we introduce the powerful tool of centralized and distributed compressive sensing which enables reliable data reconstruction under sub-Nyquist sample rate. By having defined the communication technologies and ways to minimize the amount of information that has to be transmitted over a CPS, the next step is to provide an elegant method to schedule the communication within the network to ensure sustainable operation. In this chapter, we explain the Lyapunov-Drift-Plus-Penalty as such a method which is based on queueing theory and ensures system stability while minimizing (or maximizing) a certain penalty. Lastly, in order to define optimal transmission time within a CPS, we present centralized and decentralized event-triggered control theory which optimizes the communication and guarantees the stability of the physical plant by exploiting the stability concepts of control theory.

Chapter 3

Evaluation Platforms

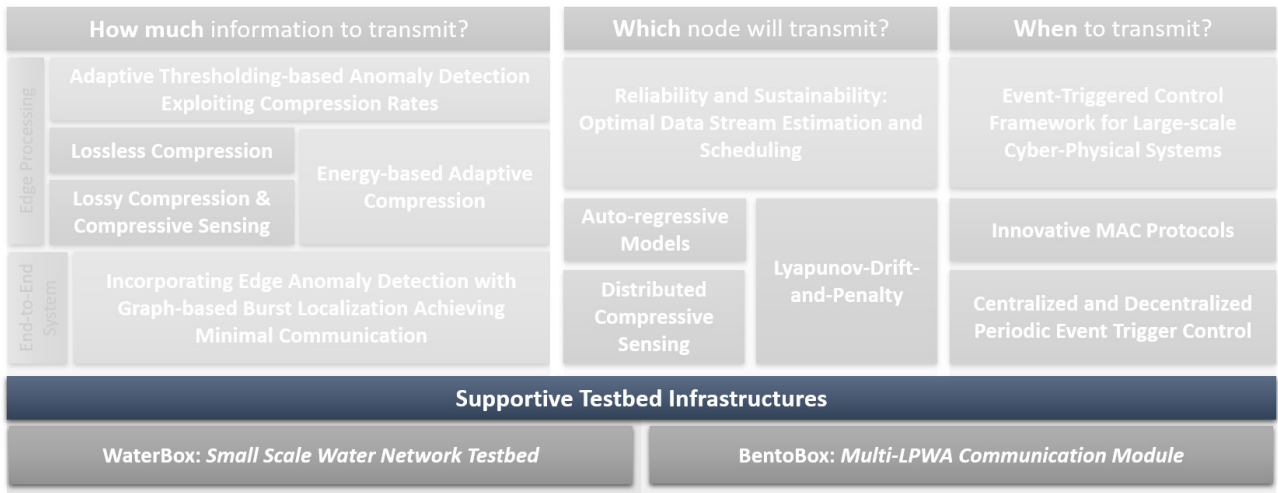


Figure 3.1: Chapter 3: Evaluation Platforms and Supportive Testbed Infrastructures.

Smart water distribution networks are a good example of a large scale CPS that requires monitoring for precise data analysis and network control. Due to the critical nature of water distribution, an extensive simulation of decision making and control algorithms is required before deployment. Although some aspects of water network behaviour can be simulated in software such as hydraulic responses in valve changes, software simulators are unable to include dynamic events such as leakages or bursts in physical models or to incorporate wireless communication delays. Furthermore, due to safety concerns, contemporary large-scale testbeds are limited to the monitoring processes or control methods with well established safety guarantees. Sophisticated algorithms for dynamic and optimal water network reconfiguration are not yet

widespread. This chapter presents a small-scale testbed, WaterBox, which allows the simulation of advanced monitoring and control algorithms in a fail-safe environment and enhances the collaboration between civil engineering researchers and computer scientists under the context the of NEC "Smart Water" project [11]. The flexible hydraulic, hardware, and software infrastructure enables a substantial number of experiments related to in-node data processing, compression and decision making, energy optimization, event-driven communication, and automatic control.

In addition to monitor and control emulation, long-range wireless communication constitutes equally important aspect of large-scale CPSs. Contemporary, Low Power Wide Area (LPWA) communication technologies have the potential to provide a step change in the enablement of cost-effective and energy efficient CPSs. With an increase in the number of offerings available the real performance of these emerging technologies remain unclear. That is, each technology comes with its own advantages and limitations; yet there is a lack of comparative studies that examine their trade-offs based on empirical evidence. This poses a major challenge to CPS architects and developers in selecting an appropriate technology for an envisioned CPS application, such as smart water networks. In this chapter, we look beyond data sheets and white papers of LPWA communication technologies and provide insights into the performance of three emerging LPWA solutions based on real world experiments with different traffic loads and in different urban deployment contexts. Under the context of this study, a specialized evaluation infrastructure, BentoBox, was created to incorporate the different technologies and provide experimental quantitative and qualitative information related to data rates, success rates, transmission mode energy and power consumption, and communication ranges. The results of experimentation highlight the practicalities of placing LPWA technologies in real spaces and provide guidelines to smart water networks solution developers in terms of LPWA technology selection. The experimental results were used to setup the evaluation of most of the algorithms in this thesis.

3.1 WaterBox: Water Network Monitoring & Control Testbed

Both the physical and computational elements of smart water networks require simulations. However, software simulators can cover either water network behaviour [98] or wireless sensor network features [99]. Even in [100], where outputs of hydraulic simulations [98] are combined with Matlab algorithms were insufficient to reproduce issues that arise in real systems; examples are pipe burst or leakage, sensor malfunctions, communication interruptions or dropouts. For these reasons, large scale testbeds have been created to allow researchers to test novel control algorithms in real conditions, i.e. projects [10, 11].

Automatic control methods are vitally important to solving some operational challenges like reducing pressure driven leakage, energy usage in pumping, leak localization etc. However, a naive deployment of new control technologies in critical infrastructures and the potential failures may have catastrophic consequences for large scale operational testbeds. Thus, the construction of a user friendly small scale testbed infrastructure in a fail-safe environment to efficiently develop, deploy and debug applications and algorithms is essential. Therefore, we developed WaterBox, which emulates monitoring and control processes for smart water networks. The main benefits of this testbed are as follows:

- It allows us to simultaneously study water network and sensor network limitations. The testbed makes it possible to mimic faults, failures and constraints such as leakages and bursts, communication issues, resource constraints (i.e. power, memory), and delays in time sensitive control processes.
- In addition to enabling developers and engineers to simulate real scenarios (e.g. multi-feed DMAs), it supports easy installation, assembly, adaptation, and reconfiguration of water network topology and computing and communication elements.
- It enables computer scientists to better understand the physical system and facilitates the cooperation with civil engineers. Furthermore, it enhances the teaching of Cyber-Physical Systems.

- Cutting edge technologies are utilized by allowing the development of next generation water network sensor nodes, algorithms and the coexistence of multiple communication modules.

3.1.1 Design Overview

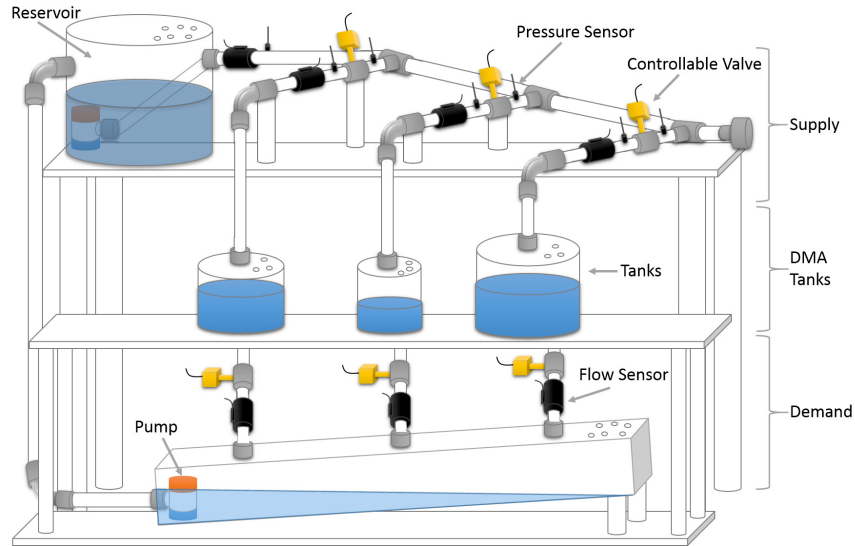


Figure 3.2: Testbed design overview.

The proposed small scale testbed, WaterBox, in Figure 3.2 aims to simulate a DMA-based smart water network, as has been presented in Section 2.1, and consists of three individual layers:

- **Supply (upper layer):** The water network backbone which simulates a reservoir, a pumping station consisting on two pumps (weak and powerful), and three neighbouring DMA inlets that are monitored by pressure and flow sensors, and controlled by analog valves.
- **DMA (middle layer):** DMA water volume state or DMA tank water levels with varying area sizes.
- **Demand (lower layer):** DMA water demand level which has one valve per DMA to mimic the variation of demand in time.

After the demand layer, the water is collected in a large tank and is recycled back to the upper-layer reservoir through a underwater pump. This closed loop feature supports long-term experiments without the need for water refilling.

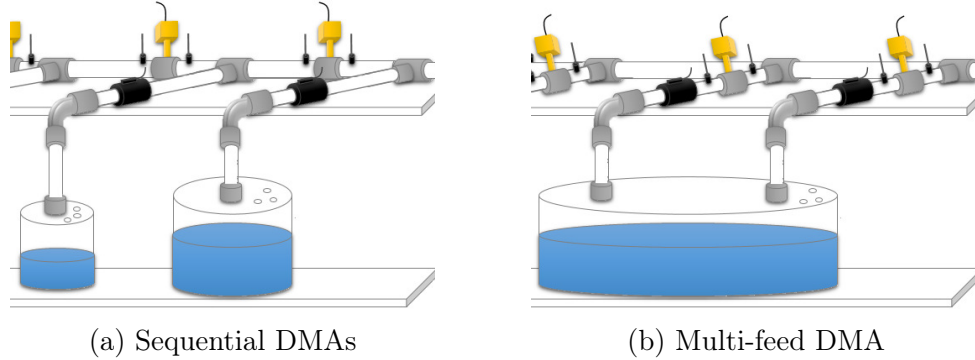


Figure 3.3: Testbed DMA reconfiguration scenarios.

Given the testbed’s hydraulic reconfiguration feature, the decision of implementing three DMA inlets derives from the need to cover a large number of real scenarios minimizing equipment requirements. For instance, the scenario of a supply zone in Figure 2.1 can be easily simulated with the configuration as shown in Figure 3.2. Furthermore, more complex and contemporary topologies can be simulated such as sub-DMA (series connection) and multi-feed DMA (parallel connection) structures by changing the order of the valves as shown in Figure 3.3a and Figure 3.3b respectively.

3.1.2 Hardware Infrastructure

This section describes the implementation of WaterBox (Figure 3.4a) focusing on the hardware and software infrastructure.

Frame and Hydraulics

In order to design a testbed for an in-lab environment that involves water experiments and electronic equipment, several things need to be taken into consideration. The first is the type of materials that will be used to build the frame and tanks. Clear acrylic was the main material used to build the frame due to its features of waterproofness, durability, resilience,

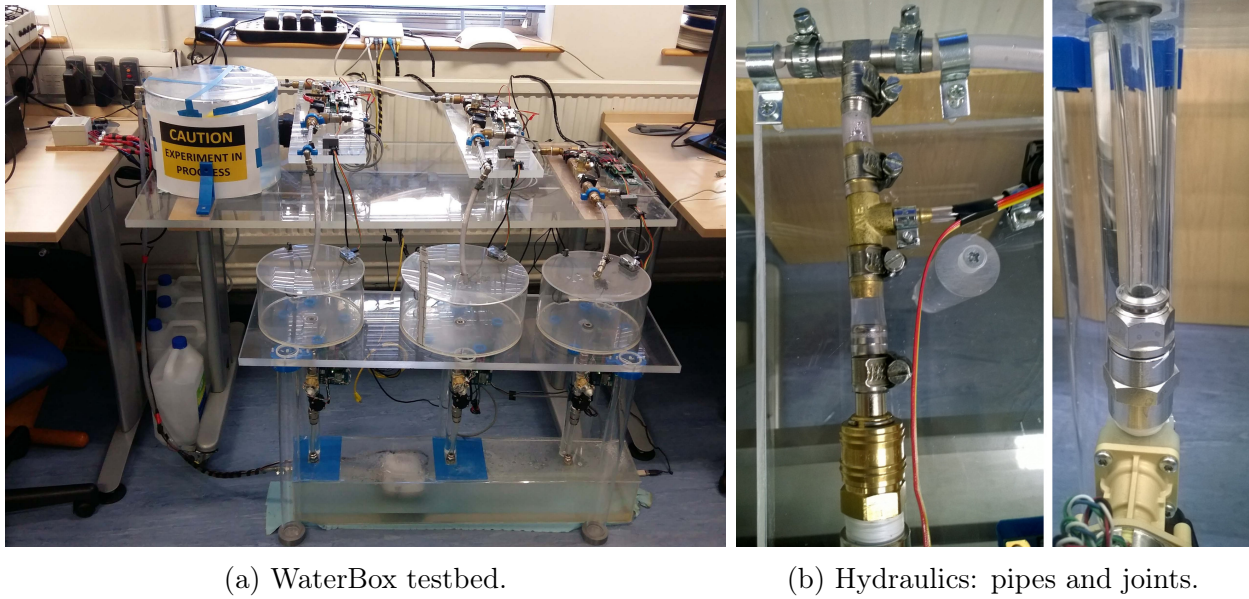


Figure 3.4: WaterBox hardware infrastructure.

and transparency, while stable metallic legs hold the upper level of the testbed. For the same reasons, the tanks and reservoir were made from clear acrylic and were shielded with heavy duty epoxy resin. To ensure the durability, the tanks and reservoir were filled with water and maintained full in a fail-safe environment for four days.

The second decision regards the size of the testbed assets. Due to in-lab space availability, the upper layer size is 135cm x 70cm x 60cm (width x height x depth) as a regular desk. By having this as reference point, the lower level selected to be 100cm x 50cm x 32cm. and cuboid tank 90cm x 15cm x 16cm. To ensure that the water volume of upper level will be less than the cuboid tank volume, the dimensions of the reservoir and three tanks are 33cm x 22cm (diameter x height) and 25cm x 14cm (for each of the small tanks) respectively. Additionally, two extra tanks (20cm x 14cm and 30cm x 14cm) were created to simulate the different DMA sizes.

After the construction of the frame and the tanks, the next step was the decision and installation of hoses and joints. The required filling time of tanks (200ml/sec), which is related to the supported water pressure and hoses diameter, was the main consideration during the decision process. The diameter of hoses is 12mm and all the joints are from stainless steel that both allow pressure up to 10 bars. A main requirement in testbed is to support visibility of water flow and thus clear flexible hoses were used. The testbed contains two types of joints (Figure 3.4b)

for the two layers respectively: (a) couplers (quick release and regular) with nozzles binding with hose clamp and (b) pneumatic. The difference lies on the need to simulate bursts and leakages on the upper layer and to ensure stability on the lower layer.

WaterBox Sensor Node

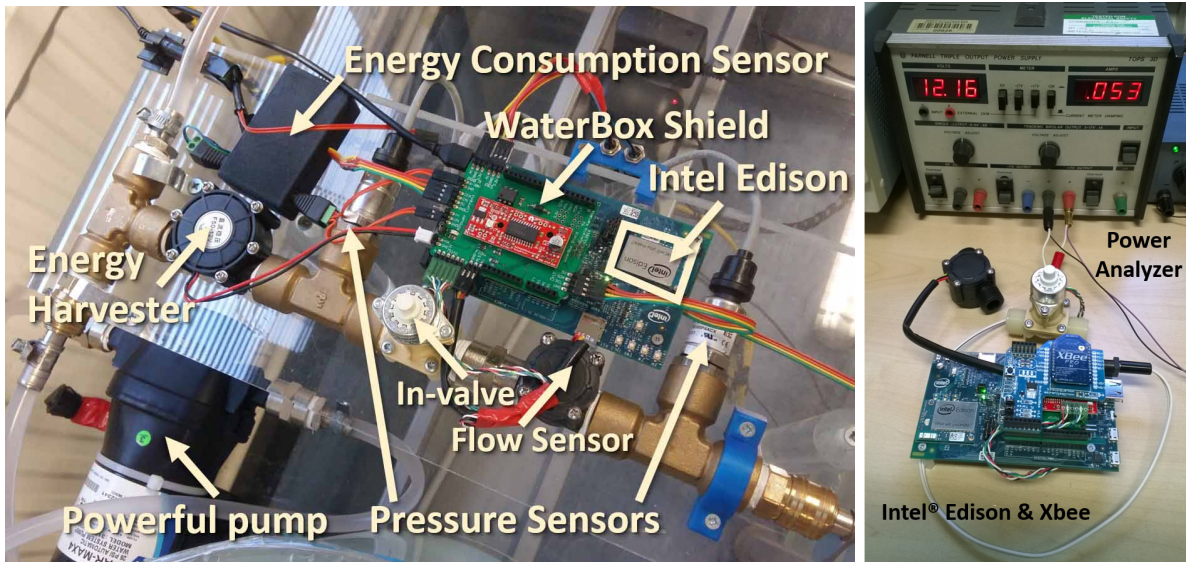


Figure 3.5: WaterBox sensor node and power analysis.

One of the goals of WaterBox development is to build a smart autonomous water network involving cutting-edge technologies that can be deployed to real networks. Furthermore, WaterBox should provide to researchers and developers an easy interface to implement and apply applications and algorithms referring to in-node decision making, control and communication optimization. This section describes a WaterBox sensor/actuator node (Figure 3.5) that allows user-friendly interaction, reconfiguration, and adaptation to simulation scenario.

Sensors and Actuator

Main requirement of in-lab small-scale testbeds is the existence of low power and cost sensors and actuators. WaterBox contains sensors and actuators that require up to 5V and 12V respectively and standard communication interfaces, while allows the integration of other sensors as well. The basic setup of a sensor node (Figure 3.5) consists of:

- **Flow Meter (Plastic 1/2" NPS Threaded):** This pulse-based sensor [101] requires 5VDC power, each pulse is approximately 2.25 milliliters. The pulse-based architecture of the sensor allows low power consumption and many contemporary systems use this type of sensors in smart grids and building power management.
- **Pressure Sensor - version 1 (MPX5700GP):** This gauge-based sensor operated between 0 to 7 bars pressure, requires 5VDC power supply, and produces analogue signal which can be transformed to kPa pressure measurements by using the following calibration function: $Pressure = (((input/VIN) - 0.04)/0.0012858) + ERROR$
- **Pressure Sensor - version 2 (PX2AN1XX250PSACX):** To increase the supported pressure range (0 to 20 bars) and the robustness of testbed, new industrial heavy duty sealed gauge pressure sensors [102] were installed which require 3.3V and 4mA. The corresponding transfer function is: $Pressure = (300.0 \cdot 5.0 \cdot input)/(VIN \cdot 1024.0)$
- **Ultra-sonic Distance (HC-SR04) and Temperature Sensor (DS18B20):** A distance sensor [103] was installed at the top cover and a waterproof temperature sensor [104] inside each tank to monitor the water level and temperature respectively. Due to lack of on wire bus and high sample rate, both of the sensors are connected to an external Arduino Pro Mini which communicates with main sensor node by using SPI bus.
- **Energy Harvester (Yosoo DC Water Turbine Generator):** To enable energy neutral operation a flow-based energy harvester was developed. The harvester is able to supply 12V with a variety of power levels based on the provided flow rate [105] per virtual "DMA" and the state of all the valves.
- **Custom (INA-based) Current Sensor:** To capture the energy consumption of a sensor/actuator node, a custom made high frequency sensor module with a separate MCU based on INA219 [106] was created. This module interfaces with a sensor node through an analogue output and a digital input pin. The high and low state of the digital pin indicate the beginning and end of energy capturing respectively. The analogue output represents the captured energy consumption. The implementation of high-speed current sensor module was carried out by Greg Jackson.

- **Servo-direct stepper motor controlled Valve (10.010.126):** This valve [107] has been specifically developed for applications that require analogue, continuous and instant adjustments of the flow rate. The valve requires 12VDC power supply, interfaces over standard six-pin stepper motor interface, and remains functional in up to 3 bars water pressure.

Additionally, a key aspect of this research is power analysis, thus a **power analyzer** was used in parallel. Figure 3.5 presents the energy requirements of our sensor node by using as development board an Intel Edison and as communication module an Xbee.

Pumps & Controller

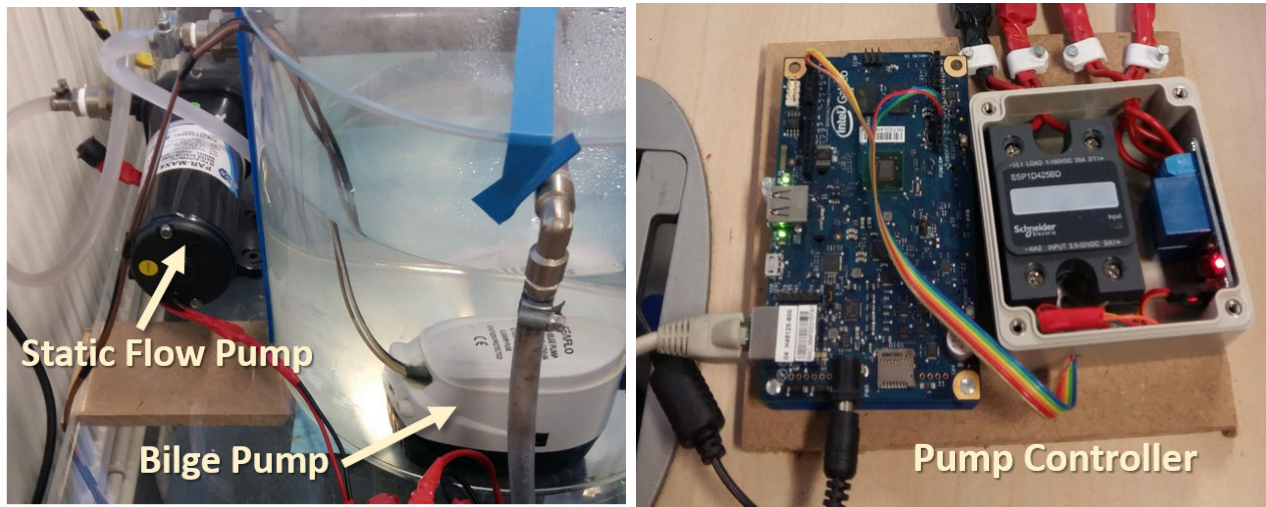


Figure 3.6: Pumps and controller infrastructure.

Figure 3.6 illustrates the pump and controller infrastructure. To emulate the day and night pumping operation of a water network in which the demand changes dramatically, we deployed two different pumps: a powerful static flow [108] and a weak bilge pump [109]. The pumps are controlled by a separate micro development board which maintains an HTTP server and enables the pump on/off actuation over the network.

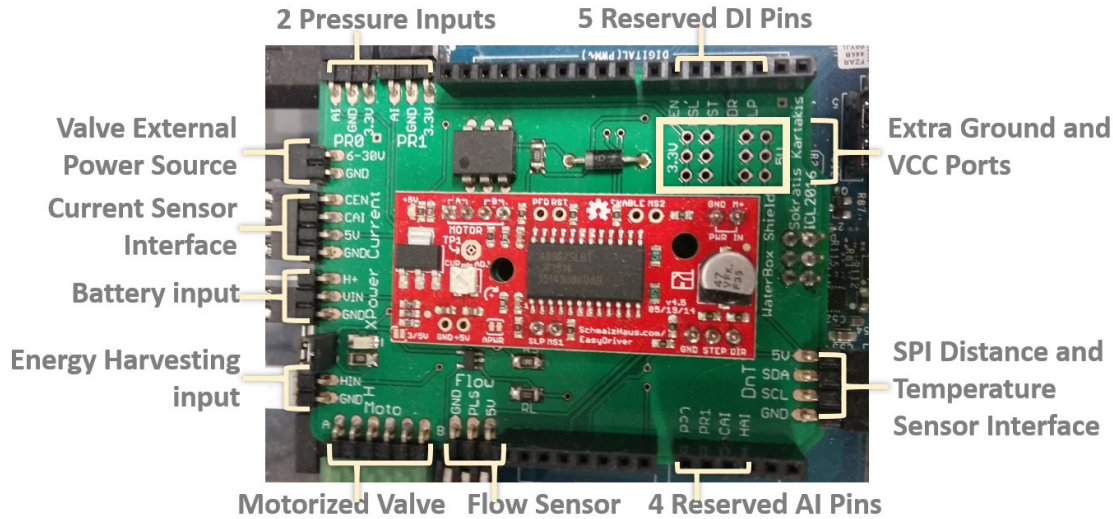


Figure 3.7: Smart water shield.

Smart Water Shield

Main challenge of designing sensor nodes for testbeds or prototypes is the component-based development of generic interfaces between sensors and development/boards that provide flexibility on changes. Custom-made solutions are unable to support simple modifications such as the installation of a new sensor. Arduino shields [110] are boards that can be plugged on top of the Arduino PCB extending its capabilities and provide extensive flexibility to prototypes. To exploit this flexibility, we designed our custom Arduino shield based on Arduino UNO R3, Smart Water Shield (Figure 3.7). Smart Water Shield reserves five digital and four analogue pins, provides one extra power input for more powerful valves (up to 30VDC), and support two pressure sensors (one per upstream and downstream), one motorized valve, one energy harvester, battery connection, SPI interface, and one flow meter.

Another benefit of the Smart Water Shield creation is the compatibility with a large number of development boards from Arduino (or other vendors that support shields [111]) with different capabilities. This fact enables the simulation of hybrid systems where the sensor nodes have different energy consumption needs and computation capabilities. An example of a critical simulation scenario, where this feature is necessary, is the control of the water network.

Development Board

Contemporary sensor nodes for smart water networks embed low power micro-controllers (MCU) [28] in order to extend battery life. In spite of energy savings, these MCUs have low memory capacity and computation capabilities (e.g. [28] contains only 2KB). This limitation prohibits the deployment of sophisticated algorithms that require enough memory to analyze a small window of historical data inside the sensor nodes. In recent years, to overcome the memory and computation limitations, many research institutes and companies have released high-performance development boards, such as Raspberry Pi [112] and Intel Galileo or Edison (which includes a dual core, dual threaded Intel Atom CPU at 500MHz and a 32-bit Intel Quark microcontroller at 100 MHz.) [113]. The main drawback of these boards is their high-power requirements being three orders of magnitude higher than MCUs.

The solution to the above issues is impacted by the following two factors: (a) the evolution of development boards and (b) pre-existence of energy harvesting systems in smart water networks. For instance, our experiments on power consumption have shown that the new Intel Edison board consumes (0.053 Amp) one order of magnitude less power than the Intel Galileo Gen 1 board (0.52 Amp). Furthermore, energy harvesting data from our large scale testbed, that embeds the Cla-Val system [114], shows that the produced power is up to 14 Watts depending on the pressure level.

Another benefit of these boards is the large variety of embedded communication modules such as WiFi, Ethernet, and Bluetooth Low Energy (BTLE). In addition, available low cost Arduino shields allow communication through Zigbee, 3G/4G, GSM etc. For example Figure 3.5 (right) presents the energy consumption experiments for Intel Edison with Xbee shield. By supporting these communication modules, WaterBox enables the evaluation of hybrid systems, routing methods, and MAC protocols.

Main goal of testbeds is the convenient development of algorithms from developers and researchers with different programming backgrounds. Intel Arduino boards support operating systems such as Linux Yocto, Ubilinux, and Windows IoT. These operating systems facilitate the creation of high-level applications that include e.g. multiple threads, file system, and

sockets. Furthermore, the multi-platform feature enables the use of high-level programming languages (on e.g. C/C++, Node.JS, Python) and frameworks that wrap difficult hardware details. Applications can be developed in powerful Integrated Development Environments (IDEs - e.g. Visual Studio) and debugged remotely through Ethernet or WiFi. Thus, WaterBox provides an easy, multi-platform, and flexible software and hardware environment that can handle most kinds of smart water scenarios.

3.1.3 Software Infrastructure

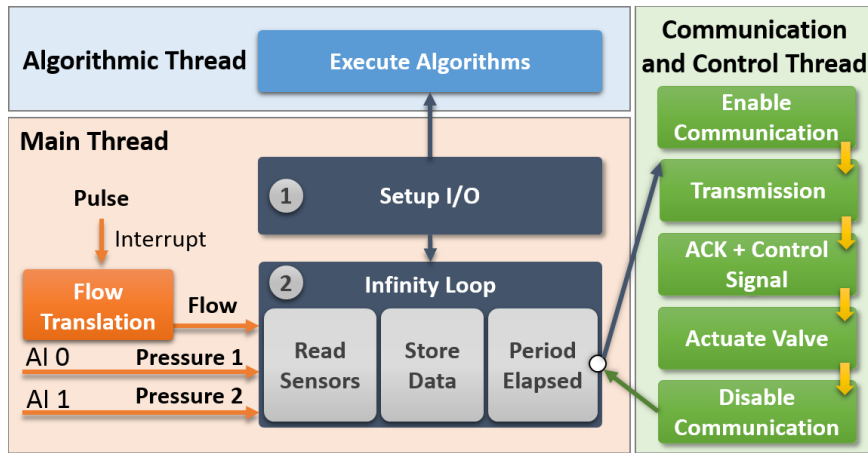


Figure 3.8: Sensor/ actuator node software infrastructure.

This section describes the WaterBox on-going experiments and presents the software architecture of the system to support them. In Figure 3.4a, WaterBox contains six sensor nodes, one per DMA supply and demand. The six sensor nodes consist of Intel Edison development boards with Ubilinux and are connected to the a local wireless network.

A basic application was created to allow new users, such as researchers and students, to apply their algorithms. Figure 3.8 illustrates the structure of this application. This application generates three main threads: (a) main, (b) algorithmic, and (c) communication and control. Initially, the main thread activates hardware I/O based on connected sensors and actuators. Additionally, in this phase a new thread is being activated to execute algorithms. After the setup, an infinity loop retrieves and stores measurements continuously, and fires an event periodically whenever a predefined time elapses. This event creates a new thread which enables the communication and control the valve (Figure 3.8 - right).

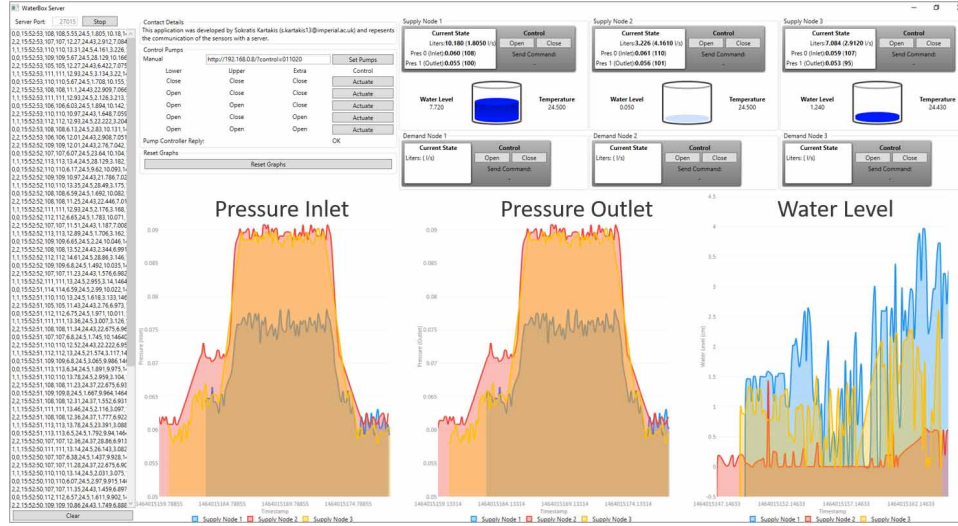


Figure 3.9: Experiment visualization software infrastructure.

A laptop is used as a local controller or base station and hosts a visualization application (Figure 3.9) which presents at real time the current state of the system, allows the manual control of actuators, logs the retrieved messages, and enables different MAC communication protocols per experiment. Additionally, due to lack of an indoor GPS time synchronization, a NTP server is running into the local controller and ensures less than a millisecond time synchronization accuracy among the nodes. Each node activates a NTP client demon as a background task which acts only during the idle or sleeping mode. The active demons and the background task of operating system cause high energy consumption and for this reason each sensor node logs the sleeping time and the corresponding energy consumption separately for further analysis.

Having this software infrastructure, our on-going experiments focus on the following categories:

In-node Decision Making: The main goal is to push the decision making such as leakage and hardware anomaly detection into the sensor nodes. This feature may allow the nodes to transmit alerts asynchronously and in real time without waiting for the next transmission period. [115] was the first deployed algorithm to detect anomalies by compressing data losslessly at the same time. During this deployment a number of compression algorithms were evaluated.

Control: The first set of experiments concerns the comparison of heuristic-based pump optimization with a Lyapunov stable reservoir level regulation control algorithm. The aim is to test the stability of the algorithms under communication constraints, model uncertainty, and

measurement errors. Other experiments test flow modulation schemes for single and multi-feed district areas.

Communication: Aim of our research is to optimize communication by transforming the traditional periodic transmission approach to event-based in smart water networks. The definition of an event constitutes our main challenge because data transmission delays and control signals' calculation may result in leakages and bursts. In order to ensure local stability and to optimize the communication we define the events based on the results of control experiments (i.e. control function, stability bounds). Furthermore, by exploiting the ability of sensor nodes to support multiple communication media, experiments based on the benefits of hybrid and opportunistic networks are being conducted.

Power Consumption: Energy consumption is vital for sensor nodes and depends on the algorithms' complexity, communication frequency, and sensor sampling. Therefore, at any aforementioned experiment type, power consumption is main metric that is being evaluated.

3.2 BentoBox: Low Power Wide Area Communication Evaluation Platform

The study of Low Power Wide Area (LPWA) communication technologies has been motivated by initial experiments performed when designing a real-world sensing solution using different LPWA modules. These preliminary studies began to observe that the theoretical capabilities of the modules differ drastically from real capabilities observed in our tests. Therefore, we could see the benefit of a more in depth evaluation of LPWA technologies. The methodology taken in this study is to mimic realistic conditions aiming to cover the majority of city CPS and Internet of Things (IoT) applications and to place the technologies in real environments for reasonable durations.

The increasing importance of the experimental evaluation of (IoT) solutions is also evident by the recent emergence of diverse IoT testbeds [116]. However, to our knowledge, no such

publically accessible testbed exists that permits comparative experiments with LPWA technologies to be performed in real deployment contexts. To this end, we designed BentoBox, a testbed node that allows in-the-wild multi-radio experiments with different LPWA communication modules (i.e. LoRa, XBee868, and nWave) to be performed in a collocated fashion, to enable conclusive comparisons across various LPWA technologies. For our experiments, a number of BentoBoxes were installed in and around the Imperial College area in different buildings and roof tops, in Hyde Park, and in manholes by reporting the following metrics: data rates, success rates, transmission energy and power, and communication ranges.

The multidimensional approach of our study, results in an empirical characterization of the performance limitations and trade-offs of the different LPWA technologies, and allows us to define the appropriate communication set-up for smart water network applications and deployment contexts.

3.2.1 Prior Limited LPWA Evaluation Studies

Many city-scale CPS applications require large scale wireless communication among devices, which can be located underground, within crowded streets, or in more sparse urban parks. Initially, evaluations of suitable technologies focused primarily on short-range communications such as WiFi [117–119], Bluetooth, and Zigbee [117–123] due to their low power features, therefore making them appropriate for battery powered CPS applications. To extend range such systems required mesh/multi-hop protocols, which were difficult to realize and increased deployment costs and communication delays. While a number of real deployments exist there are very few that report comprehensive evaluations of the technologies used [15]. In fact, many recent comparative studies make use of simulations, are based on datasheets [117–120, 124], or focus on practical evaluations over a single technological dimension e.g. power consumption [121–123].

Recently, low power wide area (LPWA) technologies have become available that allow kilometer-wide communication ranges. There is a lack of understanding of how such technologies behave under real conditions [15], their limitations are unclear, and for most studies, online product

datasheets constitute the only information source to evaluate their capabilities. Examples, [125] presents an evaluation of the LPWA Zigbee family but in a deserted area, while [126] evaluates the LoRa LPWA solution in a small-scale multiple building roof deployment. Likewise, [125] provides a comparison between CC1200-DK and LoRa iM880A but is based on line of sight (LoS) and semi-line of sight (in moderately/heavily wooded areas) experiments only. The narrow selection of the environments and conditions of these experiments is insufficient to define the real limitations of LPWA technologies under the context of CPS applications.

In this section, we evaluate three state-of-the-art LPWA technologies under the same spatiotemporal conditions: LoRa (no-standard), Xbee868 (Zigbee), and nWave (Weightless protocol). Our findings are based on real experiments, which were conducted in multiple distinct environments: a park with semi-LoS, a built up area, underground-to-over-ground, and underground-to-underground. The nature of the experiments were inspired by the many deployments of sensors in cities we have carried out in the past.

3.2.2 BentoBox Hardware Infrastructure & LPWA Experimentation

This section presents the hardware infrastructure and evaluation environmental scenarios and setup, while provides quantitative and qualitative performance comparison among three LPWA communication technologies. Additionally, it reveals critical insights for these contemporary technologies.

Hardware Infrastructure

To conduct the field experiments we developed BentoBox; a custom hardware platform based on the Intel Edison board. BentoBox incorporates multiple LPWA communication modules, software to facilitate easy programming of the modules for different experiments, and experimental data collection mechanisms (see Figure 3.10).

While communication modules for various LPWA technologies exist, in this evaluation campaign, we focused on the well-established XBee868 chip and two more recent technologies

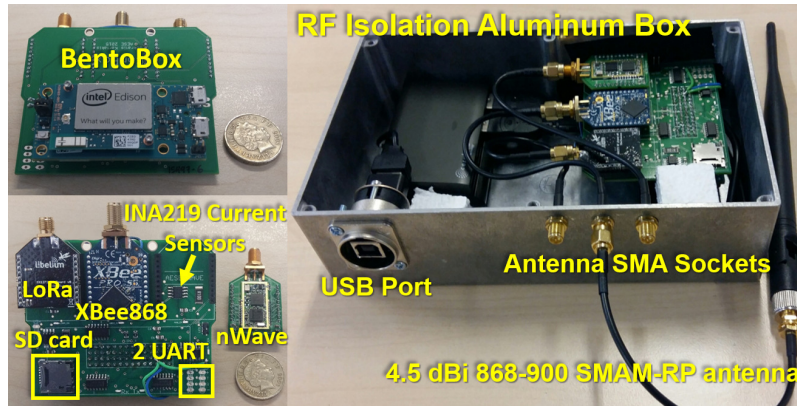


Figure 3.10: BentoBox: Hybrid LPWA Communication Box.

specifically developed for long range coverage: nWave and LoRa. The reasons for this selection were both technical (range and power consumption) and logistical (access to hardware to create a collocated node).

To provide optimal power management and enable energy performance characterizations, the board has the ability to power off and to measure the power consumption of each communication module separately. Additionally, a micro SD card extends the Edison storage capacity to log experimental data (Figure 3.10).

To ensure stable environmental and RF conditions, a critical aspect in our experimental design was the hosting hardware infrastructure of BentoBox. Each nodes were installed inside aluminum boxes, which shield the communication modules from the external RF environment by blocking other interfered signals and allowing optimal communication performance. Separate coaxial RF connectors (SMA) connect to the three communication modules. A shielded aluminum USB port connects the power supply and communication to an Intel Edison. In our experiments we placed a single 4.5 dBi 868-900 SMAM-RP antenna in a static position by connecting it to a SMA connector with coaxial cable, in order to maintain the same conditions for all modules.

Communication Modules

In our experiments, all the modules were set at maximum transmission power so that their coverage limitations were identified. LoRa determines its transmission mode based on three

Table 3.1: Technical parameters LPWA modules

Frequency: 868 MHz Power Supply: 3.3V		LoRa (Spread Spectrum)	Xbee Pro (Narrow Band)	nWave (Ultra-Narrow Band)
Data Rate (kbps) per datasheet		0.3-50 flexible	24	0.1
Power Consumption	Transmit	Range: 7 to 20 dBm 18 mA - 125 mA	Range: 0 to 25 dBm 85 mA - 500 mA	Range: -10 to 16 dBm 22 mA @10 dBm 54 mA @15 dBm
	Receive	11 mA	65 mA	N/A
	Sleep	1 uA	55 uA	1.5 uA
Rx Sensitivity		-137 dBm	-112 dBm	N/A
Max Link Budget		157 dBm	137 dBm	N/A
Payload in our Evaluation		10, 100, 180, 250 bytes	10, 50, 100 bytes	10, 20 bytes

configurable parameters: bandwidth (BW), coding rate (CR) and spreading factor (SF). We use 10 different modes as provided by the Libelium API [126] where mode-1 corresponds to the largest distance with lowest data rate (0.2 kbps), and mode-10 corresponds to a higher data rate (6.45 kbps) and shortest distance; the remaining 8 intermediary modes represent a tradeoff between link distance and data rate. Gradual variation of LoRa from mode-10 to mode-1 effectively results in enhancement in sensitivity from -114 dBm to -134 dBm (see Table 3.1), which increases the link budget for the same transmission power. On the other hand, the other modules, XBee868 and nWave, provide less configuration freedom by allowing only the selection of radio frequency and power level.

Experimental Methodology and Environments

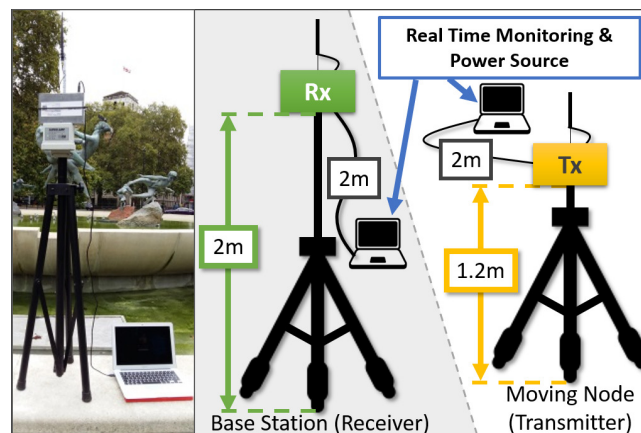


Figure 3.11: Base station (Receiver - Rx) and moving node (Transmitter - Tx) installation.

Four main experiments were performed at the university and its surroundings. During each of



(a) Ground to Ground Evaluations in Semi Line of Sight and Urban scenario.

(b) Building-Top to Underground and Over-ground Evaluations in Urban scenario.

Figure 3.12: Experimental Environments

these experiments we varied the transmission payload sizes (Table 1: "Payload in our Evaluation") and distances between the transmitter and receivers (Figure 3.12a and Figure 3.12b), transmitting 100 packets for each XBee868 and LoRa experiment and 20 packets for nWave. nWave was ran fewer times due to significant delays (i.e. more than 10 minutes per packet) in getting experimental results from the cloud server. LoRa has 10 modes as described earlier, each of which was tested per each experiment. All experiments for each technology ran over 868MHz, and were interleaved automatically using the BentoBox software in a round-robin fashion to minimize the difference in medium properties for each technology at a given moment.

Experiment set 1 (G2G) considered the case where both receiver and transmitter are located close to ground level and deployed in a semi-LoS configuration (see Figure 3.12a). The receiver was located at the highest elevation point of Hyde Park ¹ and 2m above ground (Figure 3.11 - left). The transmitter was placed on a pole at 1.20m height (Figure 3.11 - right). As the receiver was positioned at the edge of the park, it allowed us to explore two different environments: open green space and built environment. For each experiment set, we increased the distance of the transmitter from the receiver. In the case of the open green space area, we increased the distance in steps of 400m until none of the LPWA technologies were able to communicate successfully.

¹The altitude was measured accurately by using high resolution survey class GPS (<http://www.geosurvey.co.uk/geomax-gps-gnss-glonass/geomax-zenith-gps-gnss>).

In the case of the built environment, the increase was in 100m steps. The experiments were performed for both LoRa and XBee868. Unfortunately, the nWave base station required an Internet connection to a cloud based backend and constant 220V power source. Thus, we could not consider it for this deployment, as such facilities were unavailable at the experimentation site.

Experiment set 2 (G2RT) considered a ground to roof-top communication scenario, where the receiver was placed on top of an eleven floor building on the university campus. The transmitter node was placed on the ground at a height of 1.20m. As depicted in Figure 3.12b, we varied the distance between the transmitter and receiver in 100m steps until no reception was possible. As the roof top location provided us with an Internet connection, we could include nWave.

Experiment set 3 (U2RT) considered the communication case where a transmitter is placed underground and the receiver node remains at the roof top. Figure 3.12b shows underground (manhole) deployment. The transmitter was placed in a manhole and the Bentobox was placed just under 1m deep with the antenna pointing down into the manhole ² The manhole was covered with cast iron. All three LPWA technologies were evaluated.

Experiment set 4 (U2U) considered the underground to underground communication case where both sender and transmitter were placed in two manholes 80 m apart (see Figure 3.12b). The manholes were located in a mixed built, tree lined area with a mix of sand/brick clay. Similar to Experiment set 1, the utilization of nWave base station was impossible in such harsh underground environment. The placement of both nodes and antenna direction were as described in Experiment set 3.

Experimental Results

It should be noted that the results presented here are specific to our particular experimental set up; other environments and hardware support for these communications technologies may produce different figures. However, the aforementioned well-studied hardware and software

²The antenna direction was decided after optimal performance exhausted experimentation.

infrastructure aims to carry out the experimentation as scientifically as possible to produce guidelines to CPS solution developers. The results are in line with communication theory, intuition and initial controlled lab tests.

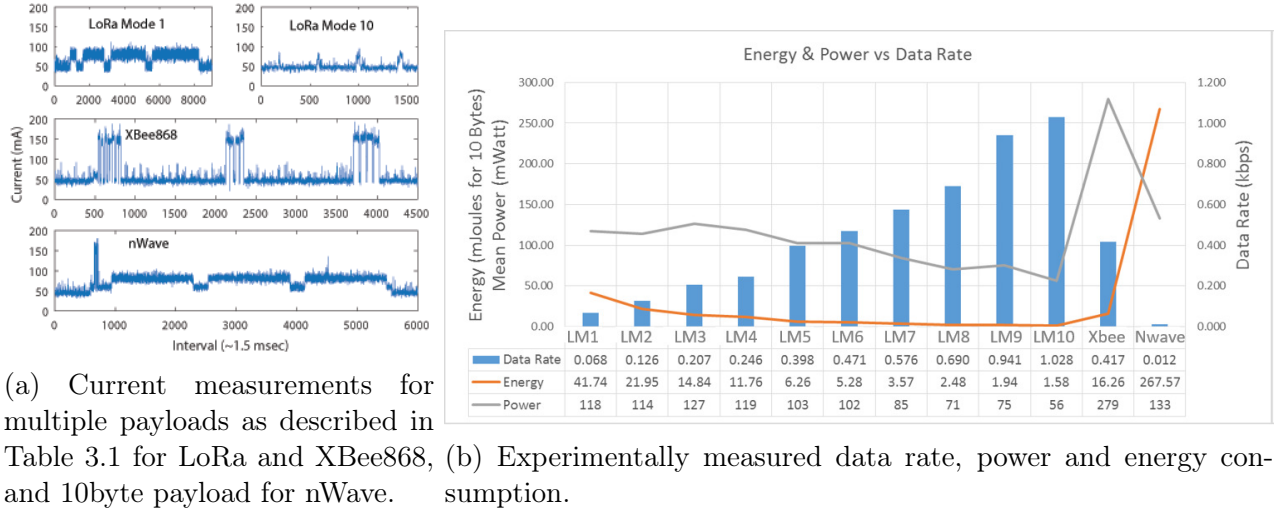


Figure 3.13: Data rate, power and energy consumption comparison.

Figure 3.12a presents the results from the G2G experiment carried out mostly in a semi-line of sight green area (Hyde Park, London) and partially in the built-up area; both nodes were placed at person level. We can observe the maximum ranges achievable for our experimental set up and associated success rates at each point. Taking the urban figures first (right two columns), we see reasonable reliability until about 500m from the base station for low data rate LoRa and XBee868. Then beyond that distance the reliability drops significantly for all mid to high data rate LoRa (LM4 and above) and XBee868. Much greater distances are achieved in the green area as expected; up to 2.4Km with lower data rate LoRa (LM1 - 0.07kbps). However, for higher data rate modes this range drops off after 500m (LM8 and LM9 - 0.69kbps and 0.94kbps respectively). XBee868 can maintain reliability for greater distances compared with higher data rate LoRa by achieving just over 1.6Km. Note, XBee868 drops its reliability at 1.2Km where the receiver was on the bridge (see Figure 3.12a) which was dense with people who, curious about the experiment, grouped around the equipment. Received signal strength indicator (RSSI) readings on the bridge dropped correspondingly. Interestingly, XBee868 is more affected by this phenomenon than the LoRa technology. This can be explained by our RSSI observations which sat at -118dBm then dropped to -127dBm, which is within the tolerable

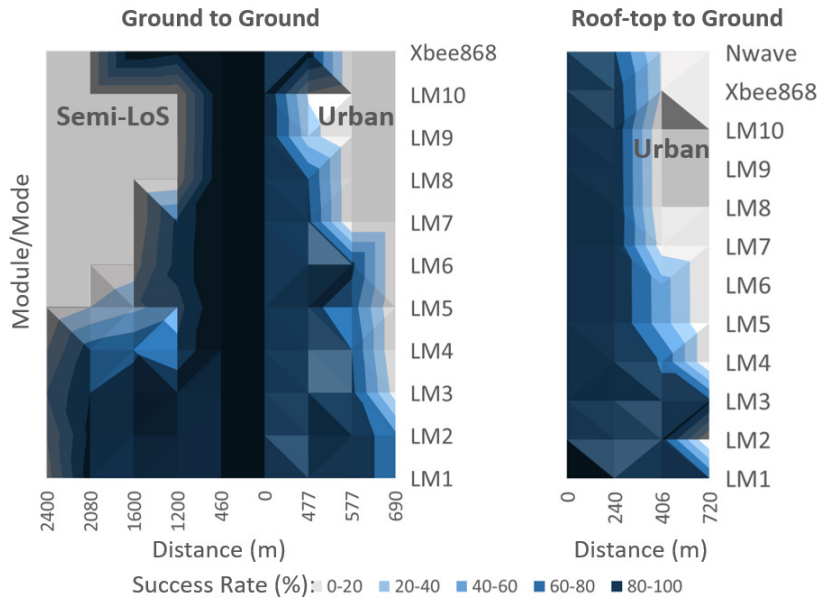


Figure 3.14: Success Rate chart for different modules/modes and range.

range for LoRa (LoRa is reliable down to -135dBm). Conversely, LoRa recovers to receive data reliability beyond 1.6Km (it achieves 92% at ~ 2 Km before becoming unreliable).

Figure 3.12b presents the comparative results among LoRa, XBee868, and nWave where the base station was on the roof-top and the nodes were either at ground level or in a manhole, in an environment dense with buildings all of six floors or taller. As one would expect, even with the elevation of the base station we achieve less distance with all technologies due to the built-up environment. Comparing the figures for overground to the urban results of Figure 3.12a we can see this difference clearly with a loss of about 20% reliability for about 450m for the LoRa results. XBee868 drops off much more significantly in the built-up area after 100m, even though it uses more power. nWave performs in a similar way to XBee868, though at significantly lower data rates (see Figure 3.12b). Based on Figure 3.12b, LoRa has an unpredictable behavior in the bounds of communication range (i.e. 760m) in which LM3 performs better (i.e. 85% success rate) than LM2 (i.e. 0% success rate). The reason of this drop is similar to the communication deficiency on the bridge in G2G experiment. The traffic combined with low RSSI (i.e. less than -130dBm) caused the unpredictable behavior of LoRa.

The results for underground to roof-top were surprisingly good for LoRa modes LM 7 and below; the success rates were similar to the above ground readings. However for LM8 and

above, very few packets were received from under the cast iron cover. Likewise, no data was received from nWave and XBee868.

The U2U experiment was limited to two manholes 80m apart in a semi-built space (small park surrounded by buildings - see Figure 3.12b). Here all technologies, except for nWave which was not included in this experiment set, achieved very reasonable reliability of greater than 95%; fact that opens a number of application opportunities for both LoRa and XBee868. To verify these results, we are planning to conduct experiments placing BentoBox into real water network chamber in the near future.

To provide an easier interpretation of the experimental results, Figure 3.14 and Figure 3.13b aggregate key information for the different communication modules under various environments in terms of range, success rates (reliability), power and energy consumption, and data rates. To ensure the comparison validity, the presented results refer to the experiment with 10 byte payload; a size which all module can support. Based on these figures three main lessons can be inferred:

Insight 1: In Figure 3.13, we observe that the data rate achieved by the XBee868 module is equivalent to that of LoRa LM5. Figure 3.14 indicates that for semi-LoS communication, XBee868 has a much longer range (around 2km) than LoRa LM5 (around 1.3km). This implies that narrowband modules such as XBee868 can provide higher data rates for the same link distances. However, at those points XBee868 uses 300mW while LoRa utilizes only 20-40mW power. Figure 3.13a illustrates precise current measurements during the communication process with different payloads as logged from BentoBoxes.

Insight 2: Figure 3.13 shows that XBee868 consumes significantly higher power (279mW) compared to the other modules. However, because of the fast transmission period the total energy consumption is similar to LoRa LM3. Because energy consumption is the main indicator of battery life, the above implies that battery-operated nodes equipped with XBee or LoRa LM3 will function for the same duration.

Insight 3: Based on Figure 3.13, the nWave module consumes similar average power in transmission mode as the LoRa module. However, the nWave transmission process lasts 34 times

longer than XBee868 and similarly, 5 times and 85 times than LoRa mode 1 and mode 10 respectively for the same payload (Figure 3.13b). As such, nWave module consumes 94% more energy in transmission modes compared to the power hungry XBee868 and between 84% and 99.5% against LoRa.

3.2.3 LPWA Communication in Smart Water Networks

The above presented findings provide useful insights for smart city solution developers that aim to deploy an enabling CPS infrastructure in a real world urban setting. Let's illustrate the usefulness of our findings using a real world problem.

Problem: Let's consider an example of a sustainable smart water network where engineers require 900 reliable pressure measurements every 15 minutes (e.g. 2 Bytes per measurement=1800 Bytes per interval without need of checksum or timestamp). Here, each sensor node is equipped with a 400mAh (~ 5330 Joules) battery and an energy harvester, i.e. water pressure difference recharges the battery by 9 joules per 15 minutes). Furthermore, the node consumes approximately 5 Joules for sensing and data processing etc.

Solution: Figure 3.14 and Figure 3.13 can guide the selection of the most appropriate LPWA communication technology. Let's assume the water network nodes are situated in a semi-LoS with a long distance apart. Figure 3.14 shows the longest range, reliable communication (over 85%) can be achieved using LoRa in LM1-LM3. When evaluating data rate and power consumption needs see Figure 3.13, here transmission of 1800 Bytes requires 3.51, 1.902, and 1.161 minutes in LoRa LM1-LM3 respectively; all represent acceptable values given the time constraint of 15 minutes. We can infer that the data transmission energy needs are 7513, 3951, 2672 mJoules every 15 minutes for LoRa LM1-LM3 correspondingly. As the smart water network must be sustainable and due to the energy harvesting system performance (9 joules per 15min) and the energy requirement for sensing, and data processing (~ 5 Joules), only LoRa LM2 and LM3 are acceptable. Additionally, the highest reliability (85% success rate) can be achieved using LoRa LM2 (Figure 3.14) over longer distances (2km). Thus, the best solution for this IoT application is the use of LoRa LM2 and creating clusters every 4km.

Lastly, during the design of CPS application, the hardware installation setup and network topology are important factors. For example, the native support for mesh routing of Xbee868 allows easy extension of network coverage. Additionally, the current nWave receiver requires a mini-PC base station connected to the Internet, which orchestrates the communication parameters. The reliance on an internet connection makes nWave inappropriate for localized deployment cases such as remote smart irrigation systems. Furthermore, the current implementation of nWave only supports uplink data collection to the base station.

3.3 Summary

In this chapter, we present the development of a small-scale testbed, WaterBox, which mimics smart water network monitoring and control scenarios, and BendoBox, a hybrid LPWA communication platform which is used to define the limitations and requirement of city-scale communication infrastructures. Both of the evaluation platforms were used to emulate our algorithms (see Chapters 5 and 6) and setup our simulations realistically (see Chapters 4 and 5).

WaterBox simulates smart water networks and enables the evaluation of in-node decision making, energy optimization, automatic control, and event-driven communication algorithms. In order to simulate a real DMA-based water network, we created an acrylic structure combined with hydraulic components that allows reconfiguration and anomaly scenarios representation (leakages and bursts). To create a flexible hardware infrastructure, we designed an Arduino-based shield, Smart Water Shield, that provides a clear interface among development boards, sensors (i.e. pressure and flow), and actuators (i.e. valves). The selected high-end development boards, Intel Edison, support a multi-platform, multi-threaded, and user-friendly development environment, which allow an easy use for researchers and students. To provide further development enhancements, we provide a fundamental software application which utilizes the hardware capabilities and procures an extendable programming template. After initial evaluation on WaterBox, the algorithms are being deployed on our large scale testbed and are being tested under real situations. Currently, we are working to extend our testbed by applying more complex

algorithms including new scenarios such as security, and new sensors such as water quality.

Smart water network applications require communication solutions that are low-cost and match the low-powered nature of sensing nodes allowing them to be placed or retro-fitted and maintained in a cost-effective way. We have seen a recent influx of new low-powered wide area communication offerings which empowers the developer with much choice but very little objective real-world information to aid design decisions. The experiments carried out in this chapter, based on BentoBox, highlight the practicalities of placing LPWA technologies in real spaces and provide *guidelines* for the urban CPS and IoT developers. However different environments provide different results and ultimately one's choice of technology is down to the application (e.g. many edge devices sending unidirectional data versus a bi-directional mesh) and is also influenced by vendor business models, technology maturity, costs (number of base stations), etc. In the study conducted for this chapter the LoRa approach would win, but the question of scalability and bandwidth allocation still exists: *"In a city of a billion devices would this approach still hold up?"* In this future scenario utilization of bandwidth may come to dominate which means the conclusions herein may no longer be relevant. To this end future work, will study the scalability and co-existence properties of these technologies.

Chapter 4

Data Reduction & Edge Processing

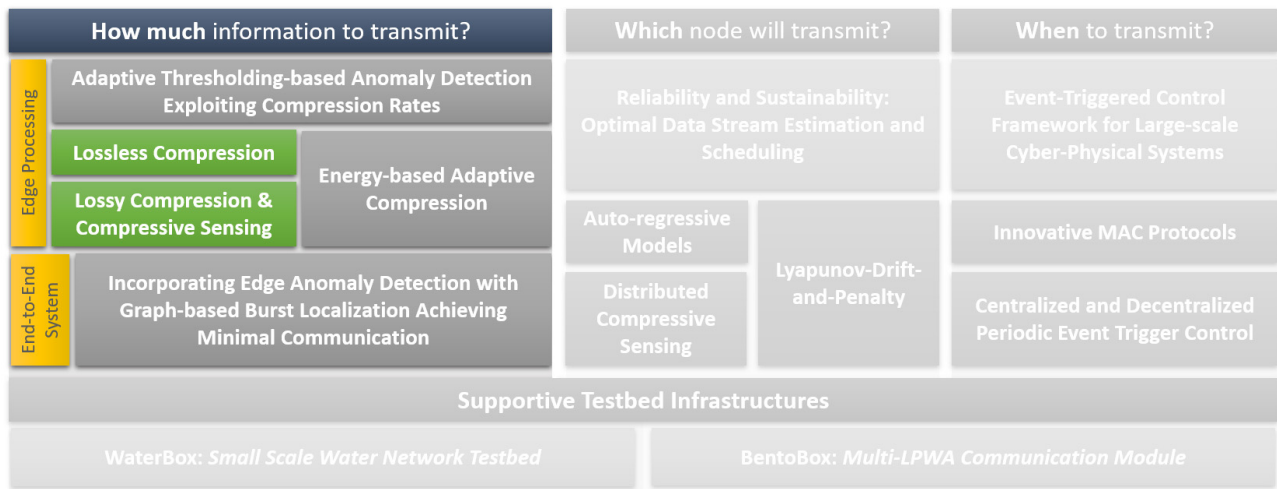


Figure 4.1: Chapter 3: Data Reduction & Edge Processing.

There is a movement in many practical applications of Cyber-Physical Systems to push processing to the edge. This is particularly important where the CPS is carrying out monitoring and control, where the latency between the decision making and control message reception should be minimal. However, CPS are limited by the capabilities of the typically battery powered low-resource devices. For example, in smart water network, low-resourced devices, such as smart meters/sensors, have been used to transfer information from the water network to data centers for further analysis. Due to the underground position of water assets, many water companies tend to deploy battery-powered nodes or data loggers which last beyond the 5-year mark. This prohibits the use of high-sample rate sensing therefore limiting the knowledge we can obtain

from the recorder data.

In order to enable the wireless acquisition of such high-sample rate data we propose lightweight edge processing techniques, the incorporation of edge processing and off-line algorithms, and in-node data reduction techniques. Specifically, in this chapter, we propose a edge processing scheme that both reduces the amount of data by applying lossless compression and at the same time carries out anomaly detection based on adaptive thresholding exploiting compression rates. In addition, we propose a self-tuning and automatic re-configuration mechanism which allows the application of our anomaly detection scheme to any kind of raw data automatically. The same mechanism provides the ability of anomaly detection self-optimization as the nature of the incoming data changes over time.

Given the ability to analyze data to the edge, we designed an end-to-end system that combines the lightweight edge anomaly detection algorithm based on compression rates and an efficient burst localization algorithm based on graph theory. The edge anomaly detection and localization elements of the system produce a timely and accurate localization result and reduce the communication significantly compared to the traditional periodic communication by exchanging only timestamps.

To reduce the amount of data even more, we present the application of Compressive Sensing (CS) [66], a lossy compression approach, in smart water networks due to the observed high sparsity level of data streams. CS enables significant compaction of the original information content in a few random incoherent projections by introducing a minimal reconstruction error.

By having the lossless compression and compressive sensing algorithms in our quiver, we propose a novel algorithm which combines the accuracy of standard lossless compression with the efficiency of a compressive sensing framework. Our method balances the tradeoffs of each technique and optimally selects the best compression mode by minimizing reconstruction errors, given the sensor node's battery state.

4.1 Adaptive Thresholding-based Real-time Anomaly Detection exploiting Compression Rates

There is a move to make Water Distribution Networks (WDN) more dynamic and intelligent using wireless sensors and actuation effecting a CPS to monitor and optimally control the water network in real time, by pushing analytics to the edge and increasing the decision-making capabilities of energy-constrained sensor nodes. Typically such CPS projects monitor the dynamical conditions of the water distribution network. Traditionally this data is sensed at the edge of the network then sent to off-line servers to identify potential failures. Here further analysis via fusion with other data sets, such as customer data may take place. To do this, high precision pressure and flow data, at rates that can exceed 100 samples a second per sensor which can equate to high-precision data averaging at over 512bytes per sec or 0.45Mbytes per 15 minutes. If the system has to transmit this amount of data in 15-minute intervals then the communication process alone will drain the battery of the sensor node rapidly. Therefore, our aim is to reduce the energy cost related to the communication without sacrificing the precision of the data. To this end, we evaluated a number of lossless compression algorithms.

During this evaluation, using real data, we observed a correlation between compression rate and data value fluctuation, and from this derived a scheme that enables the identification of transients or failures in the WDN. This means that instead of compressing raw sensor data and sending it to servers to be decompressed and then analyzed for anomalies, we can use the compression rate to detect anomalies and outliers directly on the sensor node. This is faster, more lightweight and provides early indications of an issue, which can be fed directly into the control function without having to communicate via servers saving time and energy. Furthermore, we have expanded the system using ideas inspired from active learning to support optimal selection of the algorithm's input parameters to enable self-tuning and automatic re-configuration.

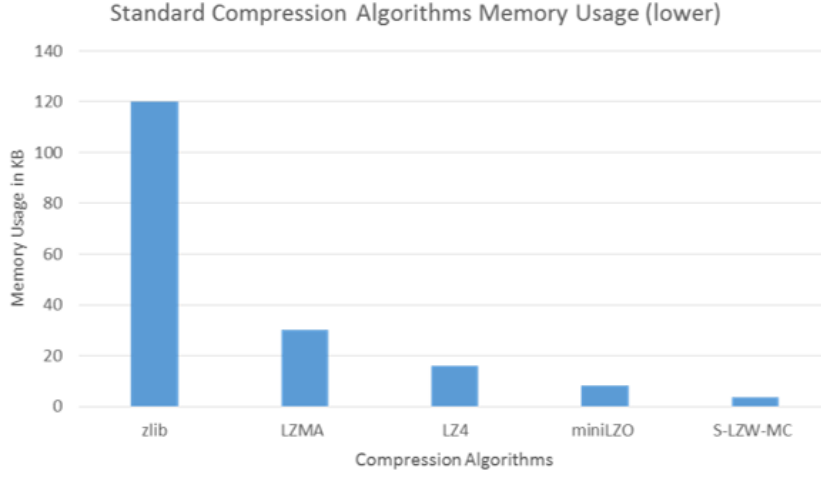


Figure 4.2: Compression algorithms evaluation.

4.1.1 Compression Rate and Raw Data Trend Correlation

In order to reduce the energy consumption that would be consumed by high data rate transmissions, while maintaining a high data precision, each sensor node uses lossless compression. Our choice takes memory and energy constraints of our devices into account, so computation and memory intensive algorithms are inappropriate, in spite of their potentially better compression rates. For example, current ultra low power MCUs have 64Kbytes memory [127], therefore, we limit compression to 10K.

Based on the above memory requirements, only MiniLZO [128] (coding method is sliding window - LZ77) and S-LZW-MC [129] (coding method is dictionary - LZ78), which require 8.192KB and 3.250KB of memory correspondingly (Figure 4.2), are applicable. We evaluated the compression rate of each algorithm using three different real datasets (Datasets A, B, and C - Figure 4.3 black line) of the smart water network [11], as presented in Section 2.1. Each data set consists of 5.5 million data pairs. In the evaluation the input stream is converted into 512-byte packets with the following structure: (a) timestamp (8-byte double data type), (b) 62 measurements (8-byte double * 62 = 469 bytes), and (c) CRC (8-byte double data type). Both MiniLZO and S-LZW-MC performed similarly (i.e. 54% and 55% for dataset A accordingly) by achieving almost the same compression rates for the three datasets. However, due to implementation limitations of S-LZW-MC (i.e. size of packet to be compressed), we decided to continue our experiments with MiniLZO algorithm.

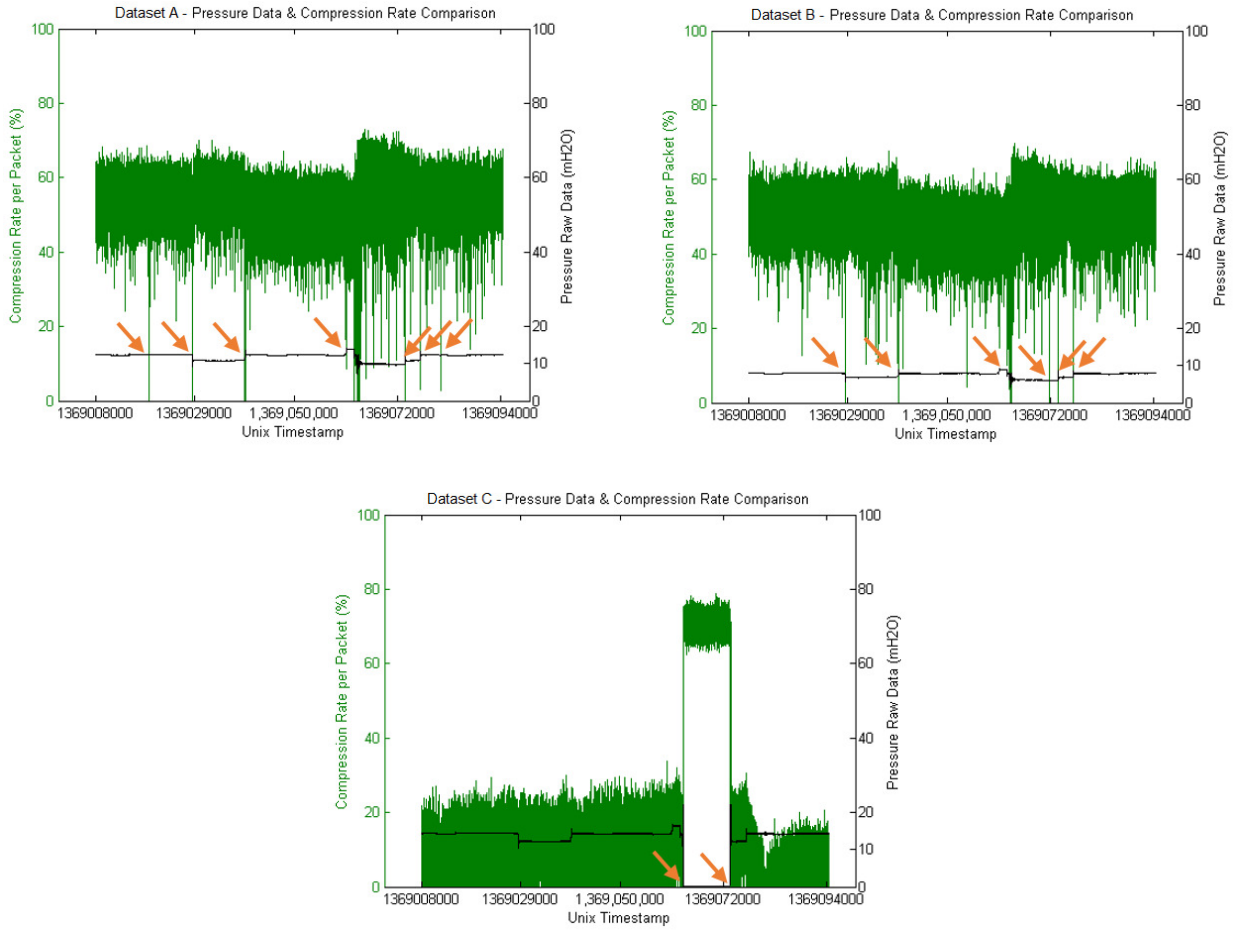


Figure 4.3: Compression rate & raw data comparison of Dataset A, B, and C.

Using the miniLZO compression algorithm, the results of the compression rate per packet can be seen in the three charts of Figure 4.3. Note that the original water pressure data is also overlaid on the same graphs in black. It is clear from Figure 4.3, that these traces highlight data anomalies as indicated by the orange arrows. From this, we formed the hypothesis that we could use the correlation of the compression rate and raw data.

We verified offline that the anomalies indicated on our graphs were true. From water technician logs we observed that they were valve position changes which were used to simulate water bursts, causing significant pressure data fluctuation. Due to the low byte correlation of the measurements at these points, the compression algorithm is unable to compress the data so the compression rate falls to 0%. In Figure 4.3, the drop in compression rate isolates the areas of raw data where the fluctuation pattern is changeable. The Dataset C is more problematic and the water pressure fluctuation is quite high (see Figure 4.3c) resulting in the compression rate

averaging at 16%. Further, in the same dataset a great drop in water pressure occurs (because the local valve was closed for a small period) which impacts compression rate, which increases to 70%.

Proving experimentally that our hypothesis, that compression rate fluctuation can be used to detect instabilities in high sample rate data, holds, we derived an analytics algorithm that is performed at the end of the compression stage. This has the added advantage that the analytics cost m times less in terms of scale and complexity, where m is the number of measurements per packet. Because each packet contains 62 measurements ($m = 62$), the produced compression rates are approximately 89,000 (5,518,000 total raw measurements / 62 measurements per packet). Thus, the analysis is applied to 98% less values.

4.1.2 Adaptive Thresholding-based Anomaly Detection Algorithm

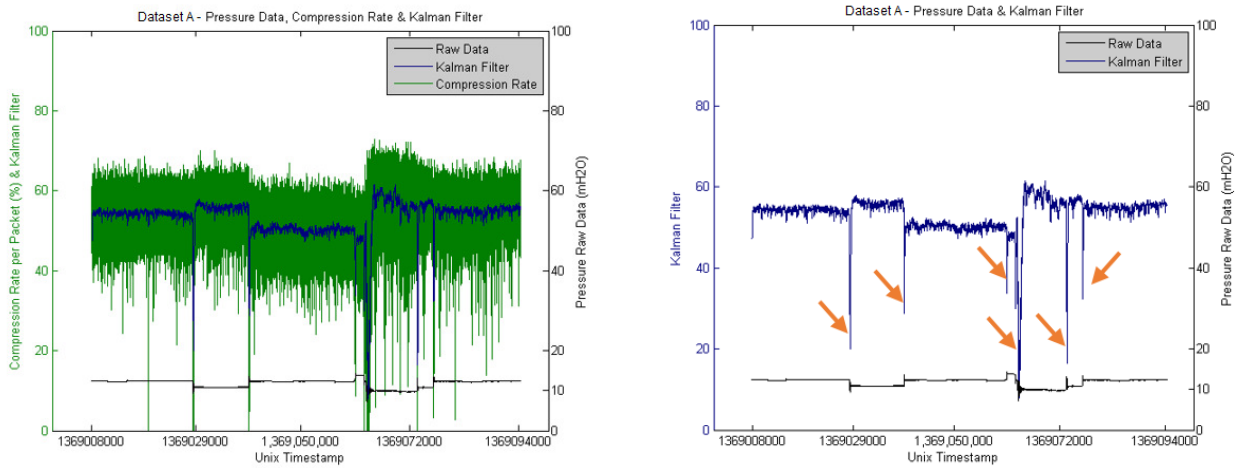


Figure 4.4: Apply Kalman Filter to Dataset A and drop detection.

We produce a scheme to automatically detect significant changes in compression rate and therefore identify the timestamps of anomalies. To maximize the anomaly detection while minimizing the number of false-positive results, noise is removed from the compression rate stream using a one-dimensional Kalman Filter [130, 131] indicated in Figure 4.4b with a blue line. The use of Kalman filters is motivated by: (a) its support of streaming analysis using only the current input measurement (and therefore is memory efficient), (b) no matrix calculations

are required (therefore it is computationally efficient), (c) ease of the algorithm tuning process, and (d) implementation simplicity.

For every new data value input, the Kalman Filter algorithm uses and updates the Kalman state. The Kalman state consists of the process noise covariance q , the measurement noise covariance r , the actual value x after noise removal, the estimation error covariance p , and the Kalman gain k . During the initialization process the parameters which need tuning are the noise q , the sensor noise r , the initial estimated error p and the initial value of x . The Kalman filter was manually initialized using the following parameters: $q = 0.005$, $r = 25$, $p = 0$, and x = the first compression rate measurement. In every new measurement, the algorithm updates the Kalman state using the following steps:

ALGORITHM 1: One-dimension Kalman filter

```

1 Omit a new measurement  $x_{new}$  ;
2  $p \leftarrow p + q$  // Kalman filter prediction ;
3  $k = p / (p + r)$  // Update  $k$  gain ;
4  $x \leftarrow x + k \cdot (x_{new} - x)$  // Get filtered value based on previous Kalman filter state  $x$  ;
5  $p \leftarrow (1 - k) \cdot p$  //Update estimated error  $p$  ;

```

After noise removal, the anomalies can be detected accurately because according to Figure 4.4b (which presents the x value of Kalman filter state and raw data) the anomalies are presented as great drops (orange arrows). The drops are being detected by using the average and the standard deviation of the compression rate moving average for a predefined window size w . We use this because it smoothes the states for easier analysis and reduces threshold computation to window sizes. Specifically, the algorithm computes the moving average of compression rate data with a window size $w = 128$ Kalman Filter x measurements, with the average avg and the standard deviation std of the moving average. The optimal selection of window size w is being discussed in a following subsection. In every Kalman state update, the algorithm checks if:

$$[x > (avg + std \cdot l)] \quad or \quad [x < (avg - std \cdot l)]$$

Where l represents the elasticity of the outlier detection (smaller values mean that the system

is more sensitive - in Figure 4.5a $l = 1.5$ and in b $l = 3$). As can be observed in Figure 4.5a (Dataset C), the algorithm suffers from a "cold start" effect (it identifies the first values to be outliers because the moving average is not calculated). To solve this problem, the algorithm initializes the *avg* and computes *std* by using the current compression rate value. Furthermore, another problem occurs when a significant variation of compression rate data is detected (Figure 4.5a). In that case, because the standard deviation has a high value, the algorithm needs more intervals for the moving average calculations to detect the outliers or anomalies. The solution is to reset the values, that is to initialize the *avg* and *std*, every time the distance between the boundaries created by the standard deviation becomes greater than a specific threshold t (in our system the threshold $t = 35$).

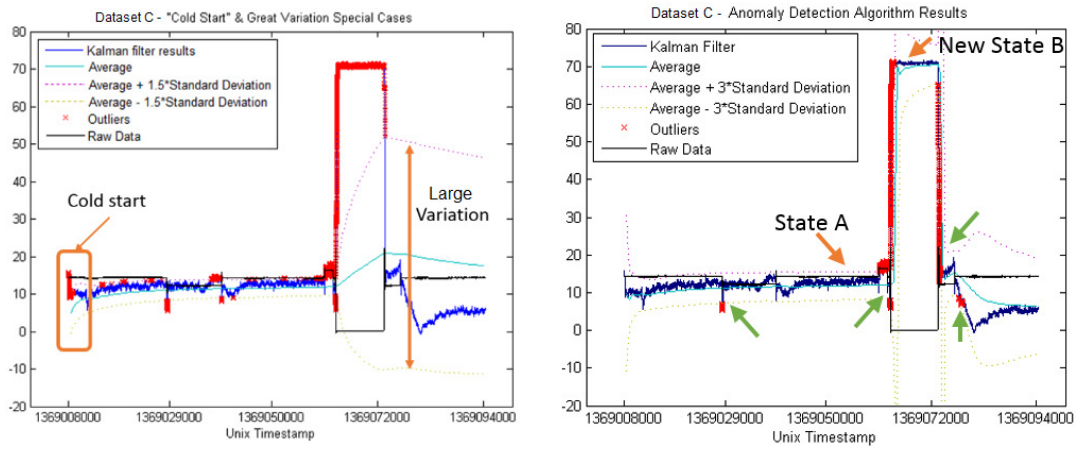


Figure 4.5: Dataset C - (a) "Cold Start", Large Variation and unelastic outliers detection ($l = 1.5$), and (b) fixed algorithm results ($l = 3$).

Figure 4.5b shows that this solves the "cold start" and "large variation" problems and illustrates the anomalies based on normalized data (green arrows). Furthermore, another benefit is that the algorithm can be adapted to changes in the behavior of the data stream. For example, Figure 4.5b, the algorithm detected the anomaly (red markers - x value = timestamp) when the compression rate changes from 10% to 70% as there is no immediate drop (State A to new State B), the algorithm recognizes that the system has a new steady state (B) until the next drop from 70% to 10%. Therefore, this shows that the algorithm adapt extremely fast to new conditions/states.

We applied this approach to Dataset A and B, and Figure 4.6 presents the results for Dataset

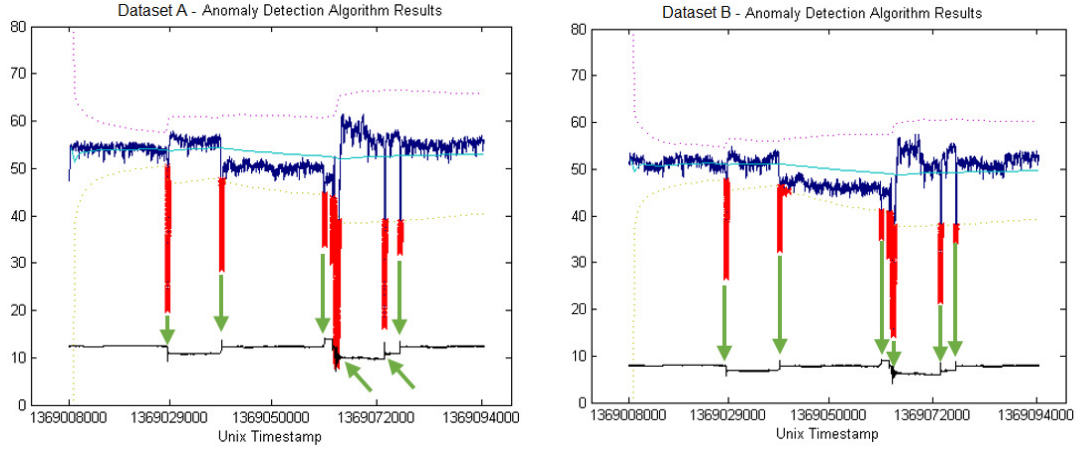


Figure 4.6: Algorithm results ($l=3$) for (a) Dataset A and (b) Dataset B.

A(raw data = black line). The red x markers are the anomalous values detected; the green arrows illustrate the process of matching the timestamps between compressed and raw data.

4.1.3 Optimal Self-Tuning via Active Learning

According to the above analysis, by tuning the input parameters, our algorithm can be applied to any case of high sample rate anomaly detection in hardware-constrained sensor nodes. However, to maximise the performance of this, an element of tuning is required as observed in the previous section. Table 4.1 aggregates all the tuning parameters required by our algorithm.

Table 4.1: Algorithm input parameters

Process	Parameters
Input stream split	Packet size m
Input stream data precision	Measurement bytes
Kalman Filter initialization	Noise q Sensor noise r Initial estimated error p
Moving average computation	Window size w
Boundaries creation	Elasticity l
Great variation threshold	Threshold t

The initialization of input parameters using a manual approach is inappropriate because it requires permutations of all the different combinations of parameters' values. In order to optimize the algorithm's tuning process, we borrow ideas from supervised learning techniques

[132]. This approach requires feedback from real users or an offline system to identify true anomalies, but this is not onerous¹. Here true anomalies for a single representative training dataset are labeled.

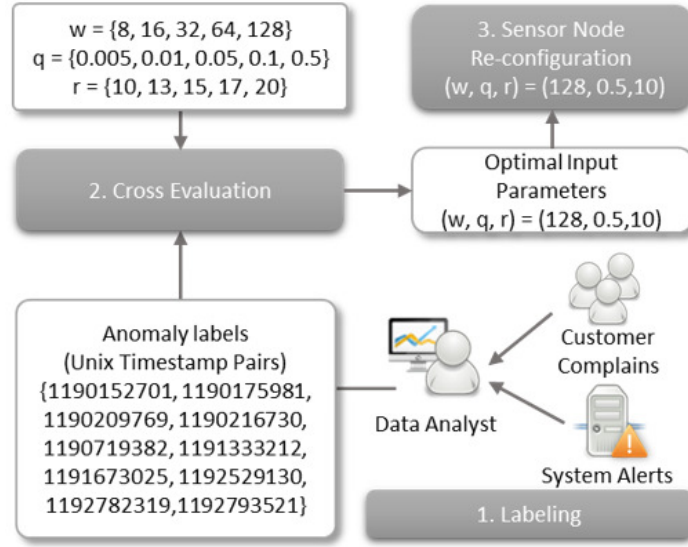


Figure 4.7: Sensor node re-configuration process.

For the results that we present here, we applied the active learning idea by asking water data technicians to manually label anomalies on a subset of our evaluation data, so-called training set. Then, we created an offline cross-evaluation system of Figure 4.7, which uses our algorithm and calculates the correct, false/positive (FP), and true/negative (TN) anomaly detections based on the initial labeling. This establishes the optimal input parameters as the combination that maximizes the following distance:

$$D = [Correct\ Detections - (FP + TN)]$$

Using this, one can imagine that a system would update parameters to re-configure the in-node anomaly detection algorithm over time. The data analysis component could recognize that the system requires re-configuration using customer complains (anomalies are being missed) and system alarms (which increase when the water network is unstable).

Before the creation of the cross-evaluation system, the algorithm's input parameters for datasets

¹Data anomaly detection can be confirmed off-line automatically by correlating candidate stream data anomalies with other data sets such as customer or water technician records.

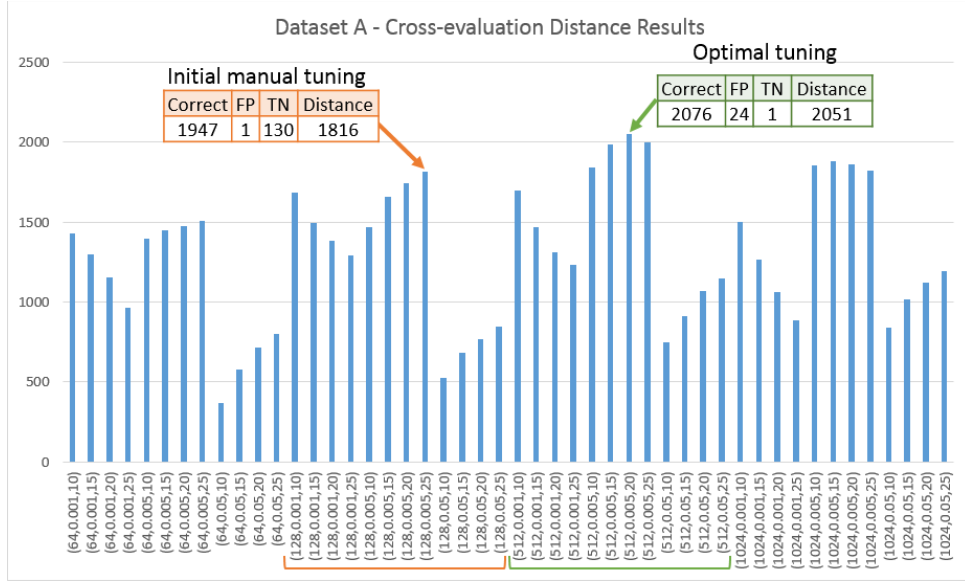


Figure 4.8: Cross evaluation distance calculation results - Dataset A.

were selected manually. In the previous section, we show that correct anomalies were identified, however, here we show that the cross-evaluation system further improves the algorithm's accuracy significantly. The reason is that manual observation of high sample rate data is difficult because of the high density data. For example, Figure 4.8 presents the results of the Distance D of each different combination of the following parameter sets for Dataset A:

$$w = \{64, 128, 512, 1024\}, \quad q = \{0.001, 0.005, 0.05\}, \quad r = \{10, 15, 20, 25\}$$

The orange arrow on Figure 4.8 are showing input parameters we derived manually, which were $w = 128, q = 0.005, r = 25$ (Figure 4.8 orange combinations). The initial selection of parameters was made on a conservative moving average window $w = 128$ because our intuition was that a smaller window used to calculate the thresholds, would provide greater accuracy. However, the cross-evaluation system shows that the optimal combination would have a window of $w = 512, q = 0.005, r = 20$ (Figure 4.8 green arrow) where the distance D indicates that accuracy will be increased by 15%. Because the cross-evaluation system uses an exhaustive approach, it always returns the optimal combination of input parameters reducing the effort and time to find the optimal combination manually.

4.1.4 Extended Evaluation

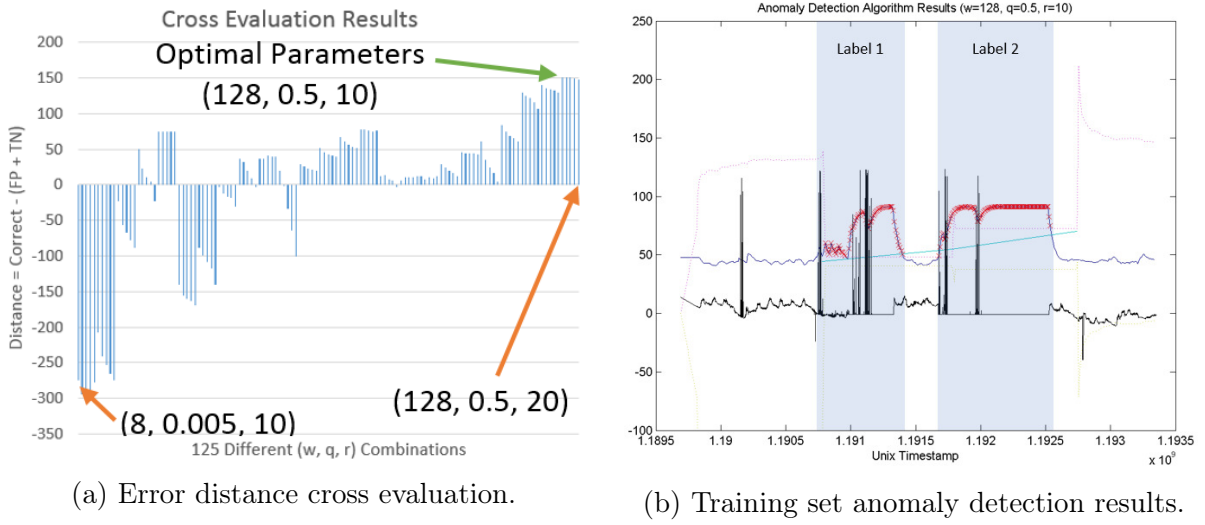


Figure 4.9: Applying edge anomaly detection algorithm in temperature data.

To understand the generality of the work beyond water applications, we applied our cross-evaluation system to datasets from St Bernard Mountain Pass sensor nodes in Switzerland [133] reporting temperature, soil moisture and watermark measurements.

Figure 4.9a and Figure 4.9b illustrates the results of our cross-evaluation system for the temperature training dataset. During the labeling process, we defined two periods as anomalies and we executed our cross evaluation process for all the following parameter combinations:

$$w = \{8, 16, 32, 64, 128\}, \quad q = \{0.005, 0.01, 0.05, 0.1, 0.5\}, \quad r = \{10, 13, 15, 17, 20\}$$

Figure 4.9a presents the results of distance D per combination of input parameters, where the optimal combination ($w = 128$, $q = 0.5$, $r = 10$) with 170 correct and 21 error anomaly detection ($D = 151$). Furthermore, from Figure 4.9b we can infer that the anomalies are detected precisely and in order to verify our results we manually checked 125 cross evaluation runs. We repeated the same process for soil moisture, and watermark sensor measurements from ten different sensor nodes and we achieved similar results.

4.2 Incorporating Edge Analytics and Off-line Graph-based Burst Localization Algorithm

The use of sensing systems to identify water leakages and bursts have been around for some time [2]. Given the cost of digging up roads to fix leaks, timely and accurate determination of the location of a leak is now more important than just leakage identification. To carry out localization high-fidelity sensing and analytics are required to triangulate leakage accurately. State of the art high-precision sensors can help build such a system by introducing communication overhead sending all this higher precision data back to off-line servers for processing.

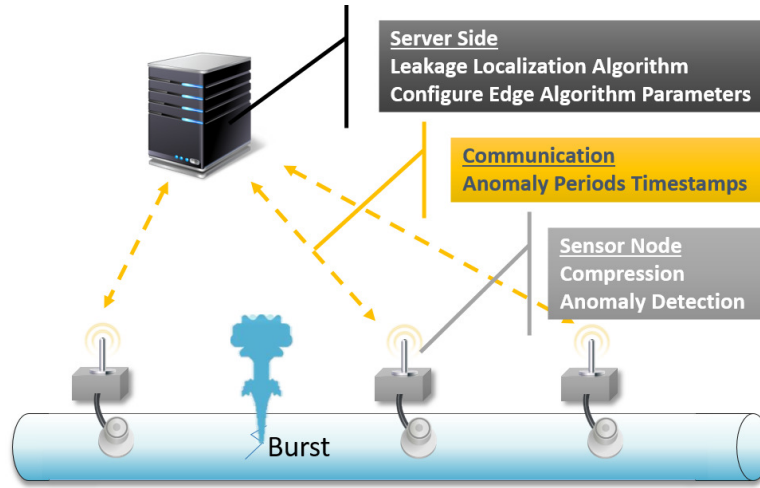


Figure 4.10: System architecture: sensor nodes, communication, and server side system.

To address the communication overhead we exploit our highly distributed lightweight scheme that combines compression and anomaly detection to identify water burst events as has been described in the previous section. On anomaly event detection we only need to send timestamp data to our novel graph topology-based scheme that recursively examines data from pairs of sensor nodes that detect an anomaly simultaneously to localize the leak. Though inspired by water distribution network leakage management, these algorithms are designed for generic sensing and analytics for other pipe based CPSs and beyond.

The work presented here is a result of cooperation with Dr. Weiren Yu, who was responsible to develop the theory behind the graph-based localization algorithm, while my contribution was related to design the cloud-edge incorporation system, adapt edge anomaly detection to

vibration data, conduct the experiments and evaluation, and implement burst localization algorithm. The outcomes of this collaboration show that we can not only significantly reduce the amount of communications by 99% between sensor devices and servers, but also that early transient or event (such as water bursts) detection can run on low-resourced sensor nodes meaning that local control functions can occur with minimal latency. The event detection software uniquely uses compression rates rather than raw data for analysis. The localization scheme demonstrates that it substantially improves positioning accuracy through capturing the water network topology information without sacrificing computational efficiency. The search algorithm combines "average distances" in the abstractions of the water network topology with differences in arrival times of the sensed anomaly as detected at the sensor locations. As far as we are aware we are the first to combine edge analytics with highly efficient graph processing to localize leaks. To evaluate our end-to-end system, we conducted real experiments to a lab-based water test rig of Welsh Water that have had controlled leakage and burst scenarios implemented.

4.2.1 Prior Related Work

In the realm of Civil Engineering, there have been numerous mechanisms used to detect leaks using novel devices that mostly focus on intrusive sensing techniques such as in-pipe sensors or even mobile sensor bots that roam within the pipe network. Some of this prior work we touch on below, but a comprehensive review is too voluminous for this publication. Our work is instead focused on less intrusive approaches that can specifically form a cost-effective distributed control system to implement the next generation dynamic water network topology. The work related to this briefly discussed in this section.

Edge processing. Current state of the art solutions, i.e. [10], use offline computationally intensive algorithms to detect anomalies and require large amounts of historical data, leading to the inability of the algorithm to be distributed and deployable on sensor nodes. For example, [134] describes a wavelet-based algorithm which continuously requires data from the sensor nodes. To overcome these drawbacks, we have introduced an efficient stream-based edge data processing and decision making algorithm [115] which enables lightweight event-based

communication.

Water Burst Localization. Over the last decade, there are some pioneering techniques proposed for water burst localization, such as wave propagation analysis [135], spectral clustering, and multiple hypotheses testing [136] (see [137] for a survey). Nonetheless, only a paucity of methods have been proposed in the context of a water network structure that explores graph topology. One excellent piece of work is Misiunas *et al.* [138] who leveraged a search-based technique to localize a burst point. Its main idea consists of two phases: in the first phase, the search is performed globally over all nodes in the network; in the second phase, if the burst is inferred to have occurred along the pipe, extra nodes are placed along each of the pipes and the global search is repeated. However, both steps of this method require that a global search over all sensor locations is performed. Therefore, its computation is expensive and will have scale issues in a water network with a high density of nodes.

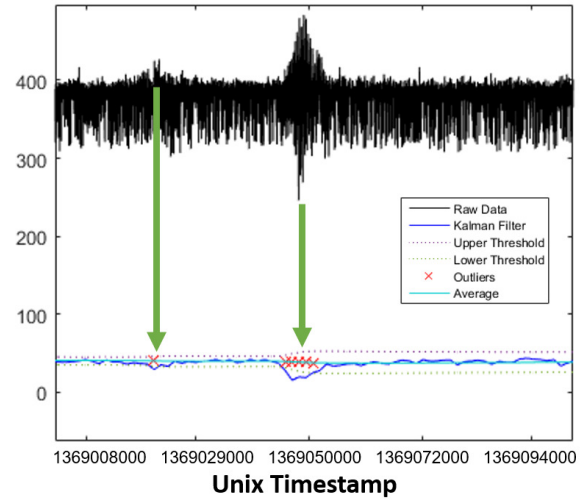
Recently, Srirangarajan *et al.* [134] utilize wave-based multiscale analysis of pressure signals to detect burst transients. To localize burst events, they also exploited the Dijkstra’s algorithm for calculating the shortest distance between every two sensor locations. Nevertheless, we observe that, when a burst occurs, its wave may travel in all the possible directions of the paths (rather than down only one path in one direction with the shortest distance) from the burst location to sensor points. Thus, to accurately position burst events, it is not appropriate to depend only on the shortest travel time between two sensor locations.

We have some prior work on localisation using graph theory [139–141] but with pressure data only. Again, not only have we reapplied this work to use vibration data, which has some subtle differences, but we have integrated the end-to-end localization systems for the first time and successfully. That is, the edge anomaly detection and localization elements of the systems which produce a timely localization result and reduce the communication by 99% compared to the tradition periodic communication. Further, this is the first use of the combination of manual and automatic localization feedback information to tune the end-to-end system parameters to maximize its performance in a low-cost way.

4.2.2 Edge Anomaly Detection based on Pipe Vibration Data



(a) Water network test rig.



(b) Anomaly detection over vibration data

Figure 4.11: Applying edge anomaly detection algorithm in vibration data.

To provide a non-intrusive solution to leakage localization we fit sensor nodes to water pipes of Welsh Water test rig (see Figure 4.11a). For this project we use the NEC Tokin Ultrahigh-Sensitivity Vibration Sensor that covers a frequency band of 10 to 15 kHz (and acceleration at 0.0001 G) with very low power requirements [142]. Such high fidelity sensors allow us to better explore water network transient phenomena, but the cost of fully transmitting that data is prohibitive using battery powered low-resourced devices. Therefore, each sensor node utilizes the system as presented in Section 4.1, to reduce the transmitted data and detect physical anomalies at the same time by exploiting vibration. In Section 4.1, we examined the application of the algorithm in pressure, temperature, and soil condition data. In this section, we configured the system parameters by using the self-tuning mechanism on vibration data. Figure 4.11b illustrates the results of edge anomaly detection algorithm as applied on vibration data.

By exploiting this edge anomaly detection algorithm, communication costs can be reduced significantly. The localization system requires only the anomaly detection arrival times from every sensor node. Therefore, each sensor transmits timestamps only instead of raw data, and only whenever an anomaly is observed at the edge. This is in contrast to the traditional periodic

sense and send routine of current water network sensing solutions as well as many WSN systems in general.

4.2.3 Graph-based Water Burst Event Localization

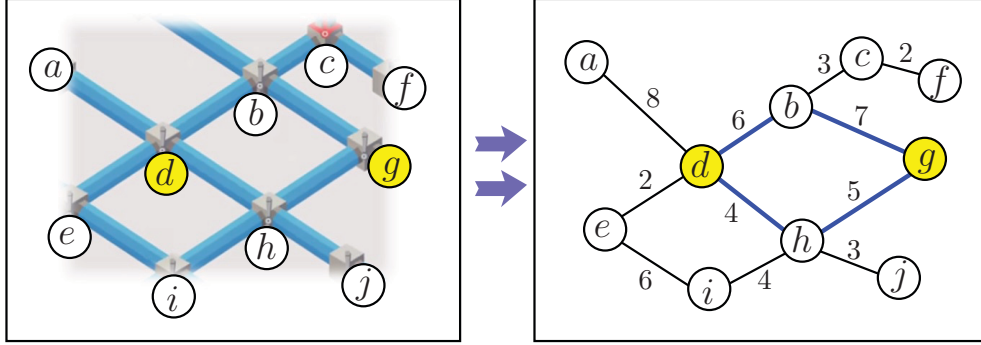


Figure 4.12: Modelling a water network (left) as a weighted graph (right)

In this section, efficient graph-based techniques are used to efficiently and accurately localize the water burst event. We first devise a novel graph topology-based measure that can quantify the “average length” between every two sensor locations, and then define our search method to localize burst events.

Graph Topology-Based Measures

A water network can be modelled as an attributed graph $G = (V_J \cup V_S, E, A)$, where V_J is a vertex set of pipe junctions, V_S is a vertex set of deployed sensor locations, E denotes an edge set of pipe sections connecting two vertices, and A carries the length of each pipe section.

To evaluate the “average length” between every two vertices in a graph G , let us first introduce the notions of the *distance matrix* $\mathbf{D}$ and the *adjacency matrix* $\mathbf{A}$.

Definition 4.1. Given a water network $G = (V_J \cup V_S, E, A)$ with $|V| = |V_J| + |V_S|$ vertices and $|E|$ edges, its distance matrix $\mathbf{D}$ is a $|V| \times |V|$ matrix, with its entry $\mathbf{D}_{u,v}$ being

$$\mathbf{D}_{u,v} = \begin{cases} \text{the length of pipe section } (u,v), & \text{if } (u,v) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

The adjacency matrix of G , denoted as $\mathbf{A}$, is defined by

$$\mathbf{A}_{u,v} = \begin{cases} 1, & \text{if } u \neq v \text{ and } \exists \text{ pipe section } (u,v) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

Example 1. Figure 4.12 depicts the water network G , whose edge weights are the length of pipe sections. By Definition 4.1, its distance matrix $\mathbf{D}$ and adjacency matrix $\mathbf{A}$ are

$$\mathbf{D} = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 6 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 8 & 6 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 5 & 0 & 4 & 3 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$\mathbf{A} = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}$$

□

Utilizing $\mathbf{D}$ and $\mathbf{A}$, we can determine the “average length” between every two sensor locations on graph G .

We first introduce a $|V| \times |V|$ matrix, $\mathbf{W}^{(d)}$, whose entry $[\mathbf{W}^{(d)}]_{(u,v)}$ is the “average length” of all paths with d hops between vertices u and v , that is,

$$[\mathbf{W}^{(d)}]_{(u,v)} = \frac{\text{the sum of the pipe section lengths over all paths with } d \text{ hops between vertices } u \text{ and } v}{\text{the number of the paths with } d \text{ hops between vertices } u \text{ and } v}. \quad (4.1)$$

To obtain the denominator of Equation (4.1), we can utilize the property of the power of an adjacency matrix. That is, the (u,v) -th element of the d -th power of $\mathbf{A}$, that is, $[\mathbf{A}^d]_{(u,v)}$, counts the number of the paths with d hops between vertices u and v .

However, evaluation of the numerator of Equation (4.1) is non-trivial as the power of a distance

matrix can only evaluate the *product* (instead of *sum*) of the pipe section lengths over all paths. As an example in Figure 4.12, to determine the *sum* of the pipe section lengths over all paths with 2 hops between vertices d and g , the result of $[\mathbf{D}^2]_{(d,g)}$ would produce the *product* of the pipe section lengths as follows:

$$\begin{aligned}
& [\mathbf{D}^2]_{(d,g)} \\
&= (\text{the } d\text{-th row of } \mathbf{D}) \times (\text{the } g\text{-th column of } \mathbf{D}) \\
&= \begin{matrix} & a & b & c & d & e & f & g & h & i & j \end{matrix} \begin{bmatrix} 8 & 6 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 \end{bmatrix} \cdot \begin{matrix} \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{bmatrix} 0 & 7 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 \end{bmatrix}^T \end{matrix} \\
&= 6 \times 7 + 5 \times 4 \neq \underbrace{(6+7)}_{d \rightarrow b \rightarrow g} + \underbrace{(5+4)}_{d \rightarrow h \rightarrow g} \tag{4.2}
\end{aligned}$$

We notice that, if the " $\times$ " operator in Equation (4.2) were changed into "+" operator, the result would turn into the more desirable *sum* of the pipe section lengths over all paths ($d \rightarrow b \rightarrow g$ and $d \rightarrow h \rightarrow g$) with 2 hops between vertices d and g . To obtain the correct "+"-based results, we can perhaps take advantage of the power of a distance matrix while changing its " $\times$ " sign (in Equation (4.2)) into "+" sign as well?

To address this, we introduce an element-wise operator $\exp(*)$. We construct the element-wise *exponential distance matrix*, denoted as $\exp(t\mathbf{D})$, as follows:

$$[\exp(t\mathbf{D})]_{u,v} = \begin{cases} \exp(t\mathbf{D}_{u,v}), & \text{if } \mathbf{D}_{u,v} \neq 0; \\ 0, & \text{if } \mathbf{D}_{u,v} = 0. \end{cases}$$

where $t \in \mathbb{R}$ denotes an arbitrary scalar variable.

Intuitively, the matrix $\exp(t\mathbf{D})$ is formed by replacing every nonzero element in $\mathbf{D}$, say x , with e^x , and keeping the zero elements of $\mathbf{D}$ unchanged.

Then, to assess the *sum* of the pipe section lengths over all paths with 2 hops between vertices

d and g , we compute the (d, g) -th element of $(\exp(t\mathbf{D}))^2$, that is,

$$\begin{aligned}
& [(\exp(t\mathbf{D}))^2]_{(d,g)} \\
&= (\text{the } d\text{-th row of } \exp(t\mathbf{D})) \times (\text{the } g\text{-th column of } \exp(t\mathbf{D})) \\
&= \begin{bmatrix} a & b & c & d & e & f & g & h & i & j \\ e^{8t} & e^{6t} & 0 & 0 & e^{2t} & 0 & 0 & e^{4t} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a & b & c & d & e & f & g & h & i & j \\ 0 & e^{7t} & 0 & 0 & 0 & 0 & 0 & e^{5t} & 0 & 0 \end{bmatrix}^T \\
&= e^{6t} \times e^{7t} + e^{5t} \times e^{4t} = e^{(6+7)t} + e^{(5+4)t}
\end{aligned} \tag{4.3}$$

In contrast to Equation (4.2), we can see that, by utilizing the operator $\exp(*)$, Equation (4.3) converts all “ $\times$ ” signs into “ $+$ ” signs. In light of Equation (4.3), our next step is to find out an “inverse” operator that can map $e^{(6+7)t} + e^{(5+4)t}$ back into $(6+7) + (5+4)$. Our key observation is that

$$\lim_{t \rightarrow 0} \frac{2}{t} \log \left(\frac{1}{2} (e^{xt} + e^{yt}) \right) = x + y \tag{4.4}$$

Thus, applying the “inverse” operator $\lim_{t \rightarrow 0} \frac{2}{t} \log \left(\frac{1}{2} (*) \right)$ of Equation (4.4) into Equation (4.3) produces

$$\begin{aligned}
& \lim_{t \rightarrow 0} \frac{2}{t} \log \left(\frac{1}{2} \left([(\exp(t\mathbf{D}))^2]_{(d,g)} \right) \right) \\
&= \lim_{t \rightarrow 0} \frac{2}{t} \log \left(\frac{1}{2} (e^{(6+7)t} + e^{(5+4)t}) \right) \\
&= (6+7) + (5+4)
\end{aligned} \tag{4.5}$$

whose result gives the *sum* of the pipe section lengths over all paths ($d \rightarrow b \rightarrow g$ and $d \rightarrow h \rightarrow g$) with 2 hops between vertices d and g .

Equations (4.3)–(4.5) provide an effective technique to obtain the nominator of $[\mathbf{W}^{(d)}]_{(u,v)}$ in Equation (4.1). To generalize our above result for any arbitrary element of $(\exp(t\mathbf{D}))^2$, we need to extend the “inverse” operator in Equation (4.4):

Theorem 4.1. *For any positive integer $N = 1, 2, \dots$, the following equation holds:*

$$\lim_{t \rightarrow 0} \frac{N}{t} \log \left(\frac{1}{N} (e^{x_1 t} + e^{x_2 t} + \dots + e^{x_N t}) \right) = x_1 + x_2 + \dots + x_N. \tag{4.6}$$

□

As a special case of $N = 2$, Theorem 4.1 reduces to the result in Equation (4.4). Theorem 4.1 is used for generalizing the result of Equation (4.3) for any *arbitrary* element of $(\exp(t\mathbf{D}))^k$. More specifically, in our aforementioned example, we choose Equation (4.4) (that is, $N = 2$ in Equation (4.6)) to “inverse” $[(\exp(t\mathbf{D}))^2]_{(d,g)}$ because there are *two* summands ($e^{(6+7)t}$ and $e^{(5+4)t}$) in Equation (4.3). In the general case, we observe that the number of summands for *arbitrary* element (u, v) of $(\exp(t\mathbf{D}))^k$ in Equation (4.3) should be consistent with (i) the choice of N in Equation (4.6) and (ii) the number of paths with d hops between vertices u and v (that is, $[\mathbf{A}^d]_{(u,v)}$).

Example 2. Consider the water network in Figure 4.12. To compute the sum of the pipe section lengths over all paths with $d = 3$ hops between vertices b and i , we first obtain its distance matrix $\mathbf{D}$ and adjacency matrix $\mathbf{A}$ (see Example 1). Next, we evaluate $[\mathbf{A}^3]_{(b,i)} = 3$ and

$$[(\exp(t\mathbf{D}))^3]_{(b,i)} = \underbrace{e^{(6+2+6)t}}_{b \rightarrow d \rightarrow e \rightarrow i} + \underbrace{e^{(6+4+4)t}}_{b \rightarrow d \rightarrow h \rightarrow i} + \underbrace{e^{(7+5+4)t}}_{b \rightarrow g \rightarrow h \rightarrow i}$$

Finally, choosing $N = 3$ in Theorem 4.1, we can “inverse” $[(\exp(t\mathbf{D}))^3]_{(b,i)}$ as follows:

$$\begin{aligned} & \lim_{t \rightarrow 0} \frac{3}{t} \log \left(\frac{1}{3} \left([(\exp(t\mathbf{D}))^3]_{(b,i)} \right) \right) \\ &= \lim_{t \rightarrow 0} \frac{3}{t} \log \left(\frac{e^{(6+2+6)t} + e^{(6+4+4)t} + e^{(7+5+4)t}}{3} \right) \\ &= (6 + 2 + 6) + (6 + 4 + 4) + (7 + 5 + 4) = 44. \end{aligned}$$

Hence, the sum of the pipe section lengths over all paths with 3 hops between vertices b and i is 44. □

After the nominator of Equation (4.1) is obtained, the “average length” $[\mathbf{W}^{(d)}]_{(u,v)}$ follows immediately:

Theorem 4.2. The “average length” of all paths, with d hops between every two vertices u and

v , and $d \geq 1$, can be computed as

$$[\mathbf{W}^{(d)}]_{(u,v)} = \begin{cases} \lim_{t \rightarrow 0} \frac{1}{t} \log \left(\frac{[(\exp(t\mathbf{D}))^d]_{(u,v)}}{[\mathbf{A}^d]_{(u,v)}} \right), & \text{if } [\mathbf{A}^d]_{(u,v)} \neq 0; \\ 0, & \text{if } [\mathbf{A}^d]_{(u,v)} = 0; \end{cases}$$

□

Notice that $\mathbf{W}^{(1)} = \mathbf{D}$. This is because, when $d = 1$ and $u \neq v$, $[\mathbf{A}^d]_{(u,v)} = 1$. Then,

$$[\mathbf{W}^{(1)}]_{(u,v)} = \lim_{t \rightarrow 0} \frac{\log([\exp(t\mathbf{D})]_{(u,v)})}{t} = \lim_{t \rightarrow 0} \frac{[(t\mathbf{D})]_{(u,v)}}{t} = \mathbf{D}_{(u,v)}.$$

Example 3. Recall Example 2. Since $[\mathbf{A}^3]_{(b,i)} = 3$ and the sum of the pipe section lengths over all paths with $d = 3$ hops between vertices b and i is 44, the “average distance” is

$$[\mathbf{W}^{(3)}]_{(b,i)} = 44/3.$$

□

Theorem 4.2 gives an effective way to evaluate the “average length” $[\mathbf{W}^{(d)}]_{(u,v)}$ with a fixed number d of hops in terms of $\mathbf{D}$ and $\mathbf{A}$. Based on $[\mathbf{W}^{(d)}]_{(u,v)}$, we can obtain the “average length” matrix $\mathbf{S}^{(L)}$ within L hops as follows.

Definition 4.2. Let $0 < \lambda < 1$ be a user-controlled decay factor. Given a water network G , its “average distance” matrix $\mathbf{S}^{(L)}$ within L hops ($L = 2, 3, \dots$) is defined by

$$[\mathbf{S}^{(L)}]_{(u,v)} = \begin{cases} \frac{1}{\beta} [\lambda \mathbf{D} + \lambda^2 \mathbf{W}^{(2)} + \dots + \lambda^L \mathbf{W}^{(L)}]_{(u,v)}, & (u \neq v); \\ 0, & (u = v). \end{cases} \quad (4.7)$$

where

$$\beta = \sum_{i=1}^L \lambda^i \cdot \mathbf{1}_{\{[\mathbf{W}^{(i)}]_{(u,v)} \neq 0\}}.$$

Here, $\mathbf{1}_{\{[\mathbf{W}^{(i)}]_{(u,v)} \neq 0\}}$ is an indicator function, which returns 1 if $[\mathbf{W}^{(i)}]_{(u,v)} \neq 0$, and 0 otherwise.

□

Intuitively, $[\mathbf{S}^{(L)}]_{u,v}$ captures the weighted average distance within L hops between vertices u and v . In Equation (4.7), the first term $\lambda \mathbf{D}$ signifies that the paths of 1 hop have a contribution

of λ to $\mathbf{S}^{(L)}$; the second term $\lambda^2 \mathbf{W}^{(2)}$ means that the paths of (longer) 2 hops have a (smaller) contribution of λ^2 to $\mathbf{S}^{(L)}$, and so forth. The parameter $\frac{1}{\beta}$ is a normalization factor, which guarantees that the sum of all the weighted factors $\{\lambda, \lambda^2, \dots, \lambda^L\}$ in Equation (4.7) is 1.

The constant λ is between 0 and 1, which can be thought of as a confidence level. Empirically, it is set to 0.6–0.8, representing the rate of decay as the leak signature wave spreads across pipe sections.

Example 4. Recall the water network in Figure 4.12 and its distance matrix $\mathbf{D}$ and adjacency matrix $\mathbf{A}$ in Example 1. We choose $\lambda = 0.85$ and $L = 5$ (realistic values for the water networks). By Definition 4.2, the “average length” matrix $\mathbf{S}^{(5)}$ can be obtained as follows:

$$\mathbf{S}^{(5)} = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \end{matrix} & \begin{bmatrix} 0 & 17.902 & 20.609 & 14.996 & 14.162 & 19.000 & 22.707 & 15.729 & 19.902 & 18.729 \\ 17.902 & 0 & 9.667 & 12.999 & 12.214 & 8.512 & 13.998 & 14.512 & 18.424 & 17.512 \\ 20.609 & 9.667 & 0 & 12.609 & 14.973 & 6.159 & 13.512 & 17.306 & 17.666 & 17.000 \\ 14.996 & 12.999 & 12.609 & 0 & 9.395 & 14.609 & 14.707 & 10.792 & 11.902 & 10.729 \\ 14.162 & 12.214 & 14.973 & 9.395 & 0 & 13.000 & 17.465 & 11.434 & 11.271 & 14.434 \\ 19.000 & 8.512 & 6.159 & 14.609 & 13.000 & 0 & 15.512 & 16.000 & 19.666 & 19.000 \\ 22.707 & 13.998 & 13.512 & 14.707 & 17.465 & 15.512 & 0 & 11.853 & 12.902 & 11.772 \\ 15.729 & 14.512 & 17.306 & 10.792 & 11.434 & 16.000 & 11.853 & 0 & 10.273 & 8.951 \\ 19.902 & 18.424 & 17.666 & 11.902 & 11.271 & 19.666 & 12.902 & 10.273 & 0 & 10.382 \\ 18.729 & 17.512 & 17.000 & 10.729 & 14.434 & 19.000 & 11.772 & 8.951 & 10.382 & 0 \end{bmatrix} \end{bmatrix}$$

□

As opposed to the previous work [134] that considers only one single path of the shortest length, $\mathbf{S}^{(L)}$ can capture multiple paths of different lengths between every two sensor locations by fully exploiting water network topology information. Thus, if the “average length” $\mathbf{S}^{(L)}$ is used to quantify the wave distance travelled from a burst location to a sensor location, water loss events can be positioned more accurately, as will be shown in the next section.

Effectively Localizing Water Loss Events

Having evaluated the “average length” matrix $\mathbf{S}^{(L)}$, we next present an efficient algorithm to position a water loss event with higher accuracy.

Our basic idea is to measure the difference in “average length” to two sensor locations that detect the burst transient at given times. Specifically, let $\bar{\nu}$ denote the average wave speed, and let t_u and t_v be the time points when the burst transient event is detected at sensor locations u and v , respectively. Note that the sensor points in the water network are time synchronized, and the time of the burst event t_x is unknown in advance, but such a burst event must occur before $\min\{t_u, t_v\}$ (earlier than either of the detected time at locations u and v). We observe that the time gap between $(t_u - t_x)$ and $(t_v - t_x)$ (which can be calculated as $|t_u - t_v|$) is mainly due to the difference in “average length” from the burst (source) location x to both sensor locations u and v . Hence, ideally we have the following equations:

$$t_u - t_v = (t_u - t_x) - (t_v - t_x) \Rightarrow \bar{\nu}(t_u - t_v) = \underbrace{\bar{\nu}(t_u - t_x)}_{\text{dist}(u,x)} - \underbrace{\bar{\nu}(t_v - t_x)}_{\text{dist}(v,x)}$$

which implies that

$$\bar{\nu}(t_u - t_v) = \text{dist}(u, x) - \text{dist}(v, x) \approx [\mathbf{S}^{(L)}]_{u,x} - [\mathbf{S}^{(L)}]_{v,x} \quad (4.8)$$

Then, we can enumerate each sensor location in V to find out the top- k (k is often set to 3–5 in practice) that best approximate solutions $\hat{X} \subseteq V$ of x to Equation (4.8), that is,

$$\hat{X} = \arg(\text{top-}k) \min_{x \in V} \{|\bar{\nu}(t_u - t_v) - ([\mathbf{S}^{(L)}]_{u,x} - [\mathbf{S}^{(L)}]_{v,x})|\} \quad (4.9)$$

Thus, the elements in $\hat{X}$ form a “hyperbolic curve” with two focal points u and v . To determine the precise location along this “hyperbolic curve”, we need to choose another pair of sensor locations, say u and w , as two focal points, with the aim to produce another “hyperbolic curve”, that is, to find out another set of the top- k best approximate solutions $\hat{Y} \subseteq V$ to the following equation:

$$\hat{Y} = \arg(\text{top-}k) \min_{y \in V} \{|\bar{\nu}(t_u - t_w) - ([\mathbf{S}^{(L)}]_{u,y} - [\mathbf{S}^{(L)}]_{w,y})|\} \quad (4.10)$$

The intersection of the two “hyperbolic curves” $\hat{X} \cap \hat{Y}$ will produce a small number of possible locations where a water loss event may occur. Finally, we can search locally for the most likely water loss position along pipe sections connected to the closest sensor locations in $\hat{X} \cap \hat{Y}$.

4.2.4 Evaluation

In this section, we provide further details about the setup of Welsh Water test rig and the evaluation results of our end-to-end system by exploiting pipe vibration data.

Experimental Setup

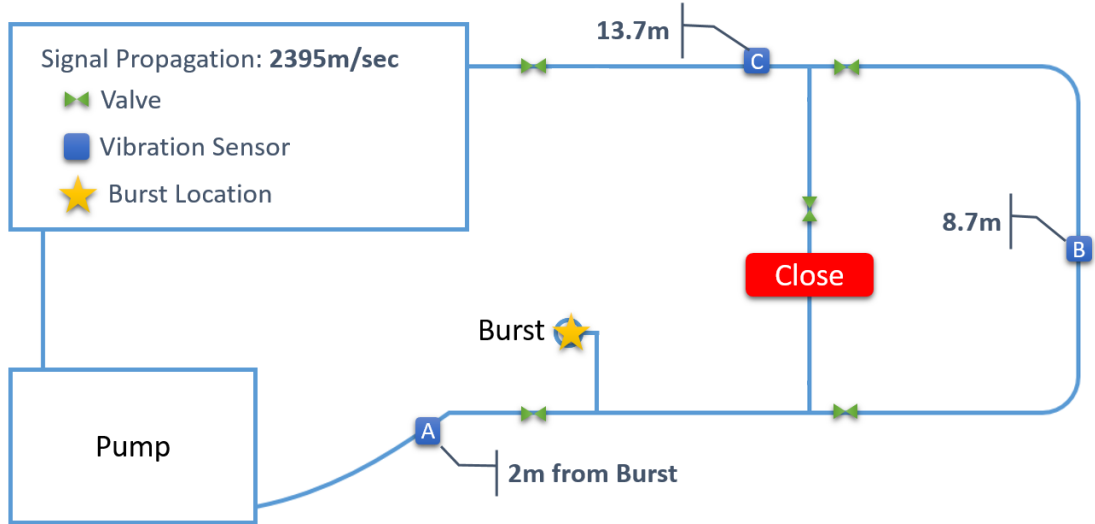


Figure 4.13: Lab-based pipe rig floor plan and infrastructure.

The ultimate aim of this Smart Water project is to study and develop real-time control algorithms that take feedback from sensors and aim to respond to anomalies, demand changes, and reconfiguration in an optimal fashion. In this project we used a lab-based test rig (Figure 4.13) which was instrumented with sensor nodes based on Intel Edison development boards and NEC Tokin Ultra high-sensitivity vibration sensors.

Figure 4.13 illustrates the infrastructure of the water network test rig which is housed in a 10x7m room. On top of this infrastructure, 7 nodes equipped with vibration sensors were deployed to the right area of the rig. Figure 4.13 presents only three nodes which produce

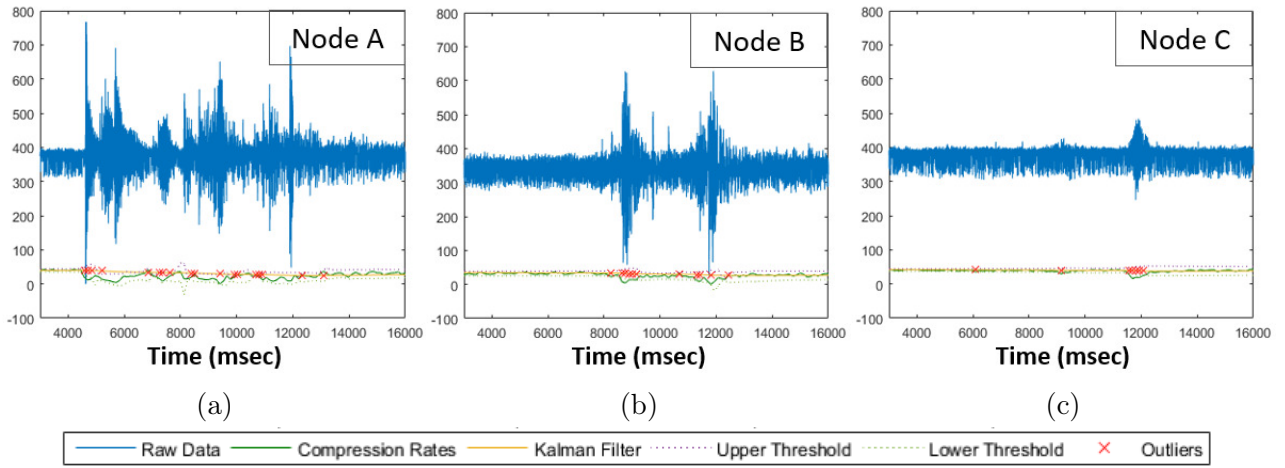


Figure 4.14: Anomaly detection from three sensor nodes (A, B, and C) which are located 2m, 8.7m, and 13.7m from the burst respectively.

Table 4.2: Edge Processing Evaluation Results

	Average Compression Rate	Observed Anomalies	Real Anomalies	Anomaly Detection Accuracy	Communication Savings (Transmit Compressed Data)	Communication Savings (Transmit only Timestamps)
Node A	28.01%	18	20	90%	79.26%	99.88%
Node B	30.06%	11	12	91%	87.69%	99.93%
Node C	39.25%	8	8	100%	92.22%	99.95%

non redundant data and are located in 2m, 8.7m, and 13.7m distances from the burst location respectively.

Evaluation Results

Using the aforementioned infrastructure, a variety of burst emulations were conducted with different pressure levels and observed using vibration data (1000 samples per second) from 7 sensor nodes. For sake of brevity, the non-redundant data from the three sensor nodes of Figure 4.14 were used. Figures 4.15 illustrate the results of the in-node anomaly detection algorithm that resides in nodes (A, B, and C) and using 16000 data points. For this we represent the water burst on a highly pressurised water network. Table 4.2 aggregates the performance of the in-node decision-making algorithm based on average compression, anomaly detection accuracy, and communication savings for transmitting compressed data or only timestamps.

Compression Rates: Data fluctuation influences the performance of the compression algo-

rithm and the position of the node in the topology means that data fluctuation rates differ. In this case the closest node to the burst (Node A) performs the lowest average compression rate. However even with this high level of fluctuation, the compression algorithm achieves more than 28% average compression rate.

Anomaly Detection The accuracy of the algorithm for all the nodes is greater than 90% for the dataset. The remaining 10% error is due to specificity as the current version of the algorithm remains slightly conservative (i.e. increased True-Negatives). For subsequent work we would adjust the processing to filter this data and identify outliers.

Communication Savings Table 4.2 presents two different communication types. In the first type, the node transmits the compressed data only when an anomaly is locally observed with a result that reduces communication by 79.26%; this is for the worst case of (Node A) compared to the base-line scenario of periodic transmission which is how such data is relayed today. On the other hand, the second communication type is where we only send timestamps to the localization algorithm which saves more than 99% communication.

Burst Event Localization:

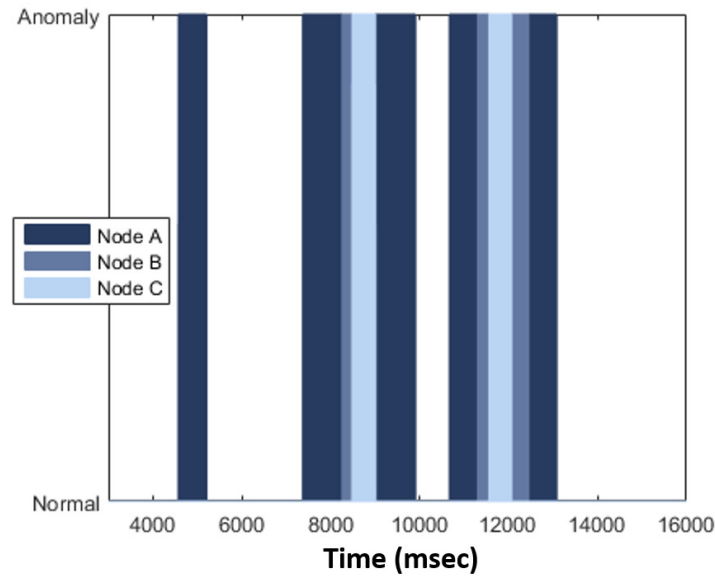


Figure 4.15: Anomaly detection periods.

To estimate the burst location, we use the proposed methods as per Equations (4.9) and (4.10) to localize the water burst event. The sound velocity in PVC is 2395 m/sec. By using the time

stamp information from all sensor locations in Figure 4.14, we can compute the difference in arrival times for every pair of sensor locations. Figure 4.15 illustrates the anomaly periods² from the three individual nodes, as defined in server side, and allows easier interpretation of anomaly arrival time differences. First, we use the time difference between node A and B , which will produce a "hyperbolic curve" with A and B being two focal points. Next, we use the time difference between node A and C , which will yield another "hyperbolic curve" with A and C being two focal points. The two "hyperbolic curves" will intersect at node $\star$ within an error radius no more than 0.5m. Note that the error radius is 0.5m because any burst event that occurs along the pipe section $\overline{ab}$ (whose length is 0.5m - see Figure 4.16) will produce the same time difference information. More precisely, as depicted in Figure 4.16, let x and y be the locations of any two burst events occurring along the pipe section $\overline{ab}$. Then, the time differences from any burst location to both A and B are exactly the same. That is,

$$\begin{aligned}
 |t_{(\star \rightarrow A)} - t_{(\star \rightarrow B)}| &= |t_{(b \rightarrow A)} - t_{(b \rightarrow B)}| \\
 &= |(t_{(x \rightarrow b)} + t_{(b \rightarrow A)}) - (t_{(x \rightarrow b)} + t_{(b \rightarrow B)})| \\
 &= |(t_{(y \rightarrow b)} + t_{(b \rightarrow A)}) - (t_{(y \rightarrow b)} + t_{(b \rightarrow B)})|
 \end{aligned}$$

This tells us that, by using the time difference information, the burst event can be determined inside a circle whose center point at node $\star$ with guaranteed $\pm 0.5\text{m}$ accuracy.

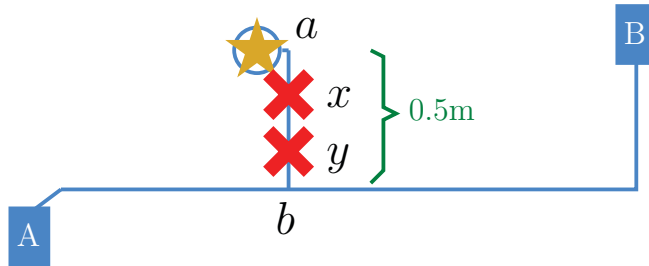


Figure 4.16: $\pm 0.5\text{m}$ accuracy in localizing the burst event

²Each anomaly period is defined based on consecutive anomaly data stream detection.

4.3 Applying Compressive Sensing to Water Networks

In the previous sections, we have examined, evaluated, and deployed lossless compression techniques in our sensing devices, which reduce transmitted data and consequently extend battery life proportionally. In this section, we introduce contemporary lossy compression approach of Compressive Sensing (CS) and examine its efficiency. Specifically, the advantages of CS are exploited for onboard compression and recovery at a base station of high-resolution pressure data recorded by sensors deployed in a water distribution network. Our experimental evaluation reveals the high performance of our proposed approach, when compared with lossless compression schemes such as [115], in terms of achieving much higher compression ratios, while maintaining highly accurate reconstructions of the original sensor data. Additionally, to our knowledge, this is the first time that CS is being adapted as a compression technique to Smart Water Network high resolution data.

4.3.1 Methodology

In this section, the design characteristics of the two main components are described, namely, the CS-based module for resolution pressure data (up to 128 samples per sec) compression, which executes on the sensor nodes (see Figure 4.17 left part), and the decompression module that reconstructs the data, which is implemented on a base station where all the sensor data are gathered (see Figure 4.17 right part).



Figure 4.17: Compressive sensing information transformation flow diagram

Edge CS-based Data Compression

Let $\Psi \in \mathbb{R}^{(N \times L)}$ be a matrix whose columns correspond to a transform basis, in general over-complete (i.e. $N \leq L$). In terms of signal approximation, it has been shown that if a signal

$\mathbf{x} \in \mathbb{R}^N$ is K -sparse in basis Ψ , meaning that the signal is exactly or approximately represented by K elements (columns) of this basis, then it can be reconstructed from $M = \mathcal{O}(K \log(\frac{N}{K}))$ non-adaptive linear projections onto a second measurement basis, which is incoherent with the sparsity basis³ [55, 143]. The general measurement model in the sparsifying transform domain is expressed as

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi^T \mathbf{w} \in \mathbb{R}^M, \quad (4.11)$$

where $\mathbf{y} \in \mathbb{R}^M$ is the vector of compressive measurements, $\Phi \in \mathbb{R}^{(M \times N)}$ is the measurement matrix whose columns are random vectors with independent and identically distributed (i.i.d.) components (i.e. each random component has the same probability distribution as the others, and the occurrence of one does not affect the probability of the other), Ψ^T denotes the inverse transform, and $\mathbf{w} \in \mathbb{R}^L$ is the sparse coefficient vector. Typical examples of measurement matrices, which are incoherent with any fixed transform basis with high probability (universality property [143]), include random matrices with i.i.d. Gaussian or Bernoulli entries.

Off-line CS-based Data Decompression

By employing the M compressive measurements and given the K -sparsity property in the transform basis, the original signal $\mathbf{x}$ can be recovered by solving an appropriate constrained optimization problem. Commonly used reconstruction techniques are based on convex relaxation [55] and greedy strategies (e.g. Orthogonal Matching Pursuit [144]). In this study, the NESTA algorithm is used, which was shown to achieve a very good trade-off between reconstruction accuracy and computation time [145]. We emphasize that the scope of this section is to illustrate the efficiency of the CS framework in achieving highly compact, yet very accurate, representations of real high-resolution sensor data recorded in water distribution networks. As such, an exhaustive comparison with the various reconstruction algorithms for finding the optimal solution is left as a separate study.

³the vectors of the measurement basis are not statistically correlated with the vectors of the sparsity basis

Focusing again on the optimization problem to be solved for reconstructing the original data, NESTA solves a synthesis-based problem of the following form:

$$\min_{\mathbf{w} \in \mathbb{R}^L} \|\mathbf{w}\|_1 \text{ subject to } \|\mathbf{y} - \Phi \Psi^T \mathbf{w}\|_2 \leq \epsilon, \quad (4.12)$$

where $\|\cdot\|_1, \|\cdot\|_2$ denote the $\mathfrak{L}_1$ and $\mathfrak{L}_2$ norm, respectively. Having estimated the sparse coefficient vector $\hat{\mathbf{w}}$, a reconstruction of the original signal is simply obtained by taking the inverse transform, that is,

$$\hat{\mathbf{x}} = \Psi^T \hat{\mathbf{w}}. \quad (4.13)$$

In the subsequent experimental evaluation, the discrete wavelet transform (DWT) [57] will be applied on the raw sensor data due to its computational efficiency and sparse representation accuracy. However, we note that the CS-based approach expressed by Equations (4.11) to (4.13) is generic and can be applied with alternative sparsifying transformations, other than the DWT.

4.3.2 Evaluation

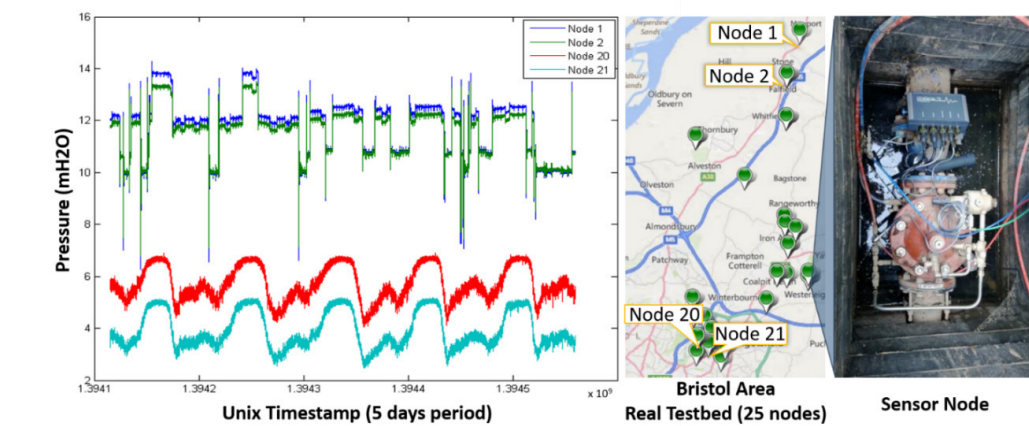


Figure 4.18: Bristol area testbed and pressure data from 4 sensor nodes

In this section, the performance of the CS-based approach is evaluated and compared with

well-established lossless compression techniques, as has been presented in Section 4.1, for compressing data streams recorded by a set of sensors deployed in the Bristol Water water distribution network of Figure 2.2. More specifically, the available dataset consists of high sample-rate pressure data (64 samples per second) from 25 sensor nodes for a 170-day period. For sake of brevity, this section presents the evaluation results for four pressure data streams from an equal number of sensor nodes as shown in Figure 4.18.

In the first test case, as Section 4.1 describes, data compressed using various lossless compression techniques. The applicable compression method for the current hardware infrastructure was MiniLZO [128], which uses sliding window as coding method (LZ77). We repeated the experiments of Section 4.1 by dividing each pressure data stream into non-overlapping windows of length $N=1024$ (or, equivalently, $1024/64 = 16$ sec) and applied MiniLZO algorithm for each data chunk by achieving 55% average compression rate.

Despite the fact that lossless compression allows the original data to be perfectly reconstructed from the compressed data, this is typically achieved at lower compression rates. To overcome these limitations, the acquired data are compressed via highly reduced sets of random measurements (ref. Equation (4.11)) and reconstructed by solving the constrained optimization problem expressed by Equations (4.12) and (4.13). To this end, each pressure data stream is divided again into non-overlapping windows of length $N=1024$. Then, a multi-scale decomposition in 10 levels is applied on each window using the DWT with the "db4" wavelet function. Subsequently, the resulting wavelet coefficients are projected onto the rows of a measurement matrix Φ with i.i.d. standard Gaussian entries (mean 0, and standard deviation 1). The efficiency of the proposed CS-based scheme is tested for a varying number of measurements, that is rows of Φ , $M = \rho \cdot N$, where $\rho \in \{0.25, 0.35, 0.45, 0.55\}$. This is equivalent to compressing the data in each window at compression rates $\{75\%, 65\%, 55\%, 45\%\}$, respectively.

The reconstruction accuracy is measured in terms of the root mean squared relative error (RMSRE). Specifically, if $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the original and reconstructed data, respectively, their RMSRE is defined as

$$RMSRE(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\frac{1}{N} \sum_{j=1}^N \left(\frac{x_j - \hat{x}_j}{x_j} \right)^2}. \quad (4.14)$$

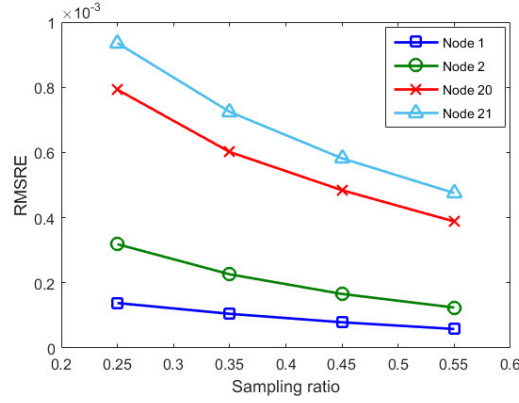


Figure 4.19: Average RMSRE per CS sampling rate for the four nodes.

Figure 4.19 shows the RMSRE, averaged over all individual windows, as a function of the CS sampling rate for each one of the four nodes. As it can be seen, the reconstruction accuracy is already high even for high compression rates (75%), whilst it improves (that is, the RMSRE decreases) as the compression rate decreases (or, equivalently, the CS sampling rate increases).

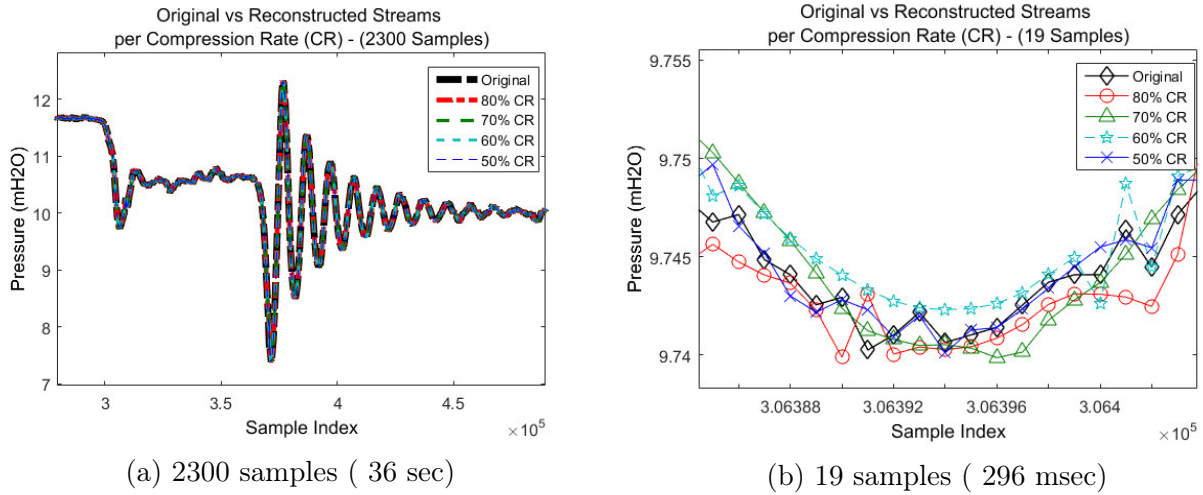


Figure 4.20: Original and reconstructed segments for sensor node 2 using NESTA algorithm

Finally, to illustrate the approximation accuracy of the original pressure data, Figure 4.20 shows segments of the original and reconstructed data streams for sensor node 2. As expected, the approximation accuracy improves as the CS sampling rate increases (see Figure 4.20b); however, a highly accurate reconstruction is achieved even for low sampling rates (see Figure 4.20a -

approximately ± 0.0025 mH₂O error) with a result of significantly higher compression rates than lossless approached which was 55% and consequently proportional battery life extension.

4.4 Energy-based Adaptive Compression in Water Network Control Systems

Up to this point, we have proposed two solutions to reduce data volume: (a) lossless compression [115] and (b) lossy compression by using the powerful framework of compressive sensing [146], which both of them minimize the communication by covering high information level needs to the server-side.

As has been described in Section 4.1 and 4.3, each compression technique has tradeoffs. Specifically, in lossless compression, the initial stream can be reconstructed completely without losing information, while compressive sensing introduces a reconstruction error. On the other hand, compressive sensing can significantly reduce the amount of data, and consequently the communication. Lossless compression places an upper bound on the compression performance.

This section propose a system to balance these tradeoffs and presents an optimal algorithm which selects the best compression mode in a dynamic and distributed fashion by minimizing the reconstruction error, given the current sensor node battery state. Based on evaluation results with real data, this algorithm can reduce the communication by around 66% and extend the battery life by 46% compared to the traditional periodic communication approaches.

4.4.1 System Overview and Preliminaries

Water utility companies deploy sensor nodes in contemporary smart water networks, such as [134], to monitor network states. These sensor nodes transmit high sample rate flow and pressure data periodically. This section uses data from the testbed in [11], as presented in Section 2.1, to simulate periodic communication and adaptive compression. Specifically, we

assume that each node communicates and selects the compression method at time t_j , where $j \in \mathbb{N}^+$, and in static time intervals $T = t_{j+1} - t_j$.

To apply lossless compression in every T , each pressure data stream is divided into non-overlapping windows of length $N=1024$ (or, equivalently, $1024(\text{samples}) / 64(\text{samples per sec}) = 16 \text{ sec}$). Then, the lossless algorithm is applied to each data chunk and compressed the data at a CR^{ls} compression rate. On the other hand, based on CS theory, a signal having a sparse representation in a suitable transform domain can be reconstructed from a small set of incoherent random projections. Thus, the user is responsible for defining the appropriate number of samples by setting the value of the Sampling Rate (SR), which is defined simply as the ratio of the number of random projections over the original signal length. To be consistent with lossless compression, this section uses compression rate (CR) instead of sample rate as the input parameter to the compressive sensing, where $CR = 1 - SR$. Similar to lossless compression, a recorded data stream is first divided into non-overlapping windows of length $N=1024$, and then CS is applied to each individual window for a varying compression rate. Then, for both lossless and CS, the decompression and recovery of high-resolution water pressure data is conducted in the base station or server side.

By having defined the main parameters of the system related to time, lossless compression and CS, the next important step is to formulate the relationship of CR with reconstruction error and battery energy consumption.

Monotonically Non-Decreasing Reconstruction Error

Based on experimental results, by applying CS to real data from our smart water sensing system, we observed that the reconstruction error, which is measured in terms of the Root Mean Squared Relative Error (RMSRE), $RMSRE_i(CR(t))$ exhibited the following property:

Observation 1. For any given sensor $i \in \mathcal{S}$, consider two distinct compression rates for CS encoding at communication interval t_j , $CR(t_j)$ and $CR'(t_j)$, where $CR'(t_j) \geq CR(t_j)$. Then, we have

$$RMSRE_i(CR'(t_j)) \geq RMSRE_i(CR(t_j)), \forall i \in \mathcal{S} \quad (4.15)$$

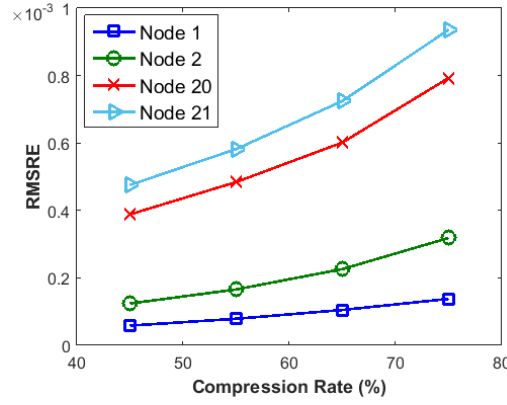


Figure 4.21: Reconstruction error ($RMSRE$) and compression rate

Based on this observation, the higher the compression rate is, the higher or equal the reconstruction error (i.e. **a monotonically non-decreasing behavior**). Figure 4.21 illustrates this observation of four different pressure signals. The inequality (4.15) is a very important property that will be exploited in the compression rate selection.

Power Management

As has been described, battery driven sensor nodes are used in many smart water networks. In a sensor node, the energy of the battery can either be consumed by in-node operations (i.e. sensing and computation) and wireless transmission, which is modeled as follows:

$$0 \leq B_i(t) \leq B_{\max} \quad (4.16)$$

where $B_{\max}$ is the battery capacity.

The dynamics of the battery at each sensor node i in every communication period t_j is modeled as a linear approximation as follows

$$B_i(t_j) = |B_i(t_{j-1}) - (1 - CR_i(t_j))E_i^{tr}(t_j) - E_i^{in}(t_j)|_+ \quad (4.17)$$

where for any real number x , $|x|_+ = 0$ if $x \leq 0$, $|x|_+ = x$ otherwise. $E_i^{tr}(t_j)$ represent the energy costs to transmit the uncompressed data of the last period $(t_{j-1}, t_j]$, which is related to the compression rate $CR_i(t_j)$ of the selected compression mode, and $E_i^{in}(t_j)$ includes all the

in-node operations (i.e. sensing, compression).

Based on Equation (4.17) and assuming that $E_i^{tr}(t_j)$ and $E_i^{in}(t_j)$ are the static variables of the last period $(t_{j-1}, t_j]$ and $E_i^{tr}(t_j) \gg E_i^{in}(t_j)$, it is clear that the selection of the compression rate $CR_i(t_j)$ (especially in compressive sensing approach) is the most important factor of battery energy consumption. The next section formulates an optimization problem which answers the fundamental question: *"Which is the most appropriate compression technique between lossless and compressive sensing (lossy), and in the case of the latter technique which CR balances the tradeoff between RMSRE reconstruction error and battery energy consumption?"*

4.4.2 Optimal Compression Technique Selection Formulation

The aforementioned problem can be seen as the maximization of the battery level $B_i(t_j)$, while minimizing the reconstruction error $RMSRE_i(CR_i(t_j))$. The solution of the optimization problem is being executed in every communication period t_j internally to the sensor node. In the case of the *lossless* compression, the actual stream can be fully reconstructed with no error ($RMSRE_i(CR_i(t_j)) = 0$). During the sensing process, each sensor node i compresses the retrieved data losslessly to reduce the size of data that has to be stored locally. Thus, the $CR_i(t_j)$ is predefined by the lossless compression algorithm and is equal to $CR_i^{ls}(t_j)$. For these reasons, in every communication period t_j , the sensor node selects to transmit losslessly compressed data if and only if the required energy to transmit this data is lower than the current battery level $B_i(t_j)$ weighted by a gain $K_p \in (0, 1]$ ⁴, or in other words if the following inequality holds:

$$K_p \cdot B_i(t_{j-1}) > (1 - CR_i^{ls}(t_j))E_i^{tr}(t_j) + E_i^{in}(t_j) \quad (4.18)$$

If Inequality (4.18) is violated then the sensor node is unable to transmit losslessly compressed data and has to select the optimal compression rate $CR_i(t_j)$ for the compressive sensing approach by solving a multi-objective optimization problem with two confronted goals. The **objective functions** of the system are related to the reconstruction error and the new battery

⁴ K_p indicates a battery level threshold in which the system select compressive sensing as compression technique. Otherwise, the systems would selects compressive sensing only when battery is close to depletion level.

level and can be described as follows:

$$\min_{CR_i^{ls}(t_j) \leq C_i(t_j) \leq CR_{max}} f_1 = RMSRE(CR_i(t_j)) \quad (4.19)$$

$$\max f_2 = B_i(t_j) \quad (4.20)$$

$$\text{subject to } \forall i \in \mathcal{S}, \forall j \in \mathbb{N}^+ t, RMSRE_i(CR_i(t_j)) \leq RMSRE_i(CR'_i(t_j)) \quad (4.21)$$

$$\forall i \in \mathcal{S}, \forall j \in \mathbb{N}^+ t, 0 \leq B_i(t_j) \leq B_{max} \quad (4.22)$$

$$B_i(t_j) = |B_i(t_{j-1}) - CR_i(t_j)E_i^{tr}(t_j) - E_i^{in}(t_j)|_+ \quad (4.23)$$

The constraint (4.21) maintains the monotonically non-decreasing ratio between the CR and the error (see Observation (4.15)) where $CR_i(t_j) \leq CR'_i(t_j)$. In order to perform the compression and transmission, the battery must not run out of the capacity at any point of time. The battery level at the specific time instant, must not exceed the maximal allowed capacity. As the consequence of those system needs, the inequality constraint (4.22) is formulated. Equality constraint (4.23) represents the battery dynamics for each interval as described in (4.17).

4.4.3 Compression Rate Selection Algorithm

Multi-objective optimization is an integral part of optimization activities and has a tremendous practical importance, since almost all real-world optimization problems are ideally suited to be modeled using multiple conflicting objectives. The classical means of solving such problems were primarily focused on scalarising multiple objectives into a single objective and evolutionary methods have been used to solve a multi-objective optimization problem as in [147]. Since this problem has only two conflicting criteria (objectives (4.19) and (4.20)), the nature of the problem requires a fast solution, and the simplicity of implementation, the selected method for solving the problem is a multi-objective genetic algorithm. Algorithm 2 incorporates both the selection between lossless and compressive sensing compression technique based on energy constraints and the optimal selection of compression rate for compressive sensing by exploiting a multi-objective genetic algorithm. The output of our proposed algorithm is the optimal $CR_i(t_j)$

ALGORITHM 2: Pseudocode for optimal compression rate selection

```

input :  $CR_i^{ls}(t_j)$ ,  $B_i(t_{j-1})$ ,  $K_p$ ,  $CR_{max}$ ,  $f_1$ ,  $f_2$ ,  $E_i^{tr}(t_j)$ ,  $E_i^{in}(t_j)$ 
output:  $CR_i(t_j)$ 
1 if  $K_p \cdot B_i(t_{j-1}) > (1 - CR_i(t_j))E_i^{tr}(t_j) + E_i^{in}(t_j)$  then
    |  $\backslash\backslash$ Select lossless compression
2 |  $CR_i(t_j) = CR_i^{ls}(t_j)$ 
3 else
    |  $\backslash\backslash$ Optimize  $CR_i(t_j)$  for compressive sensing
    |  $\backslash\backslash$  Initialize constraints
4 |  $Cnstr = \{(4.23), (4.22), (4.21)\}$ 
    |  $\backslash\backslash$ Initialize  $CR_i(t_j)$  bound
5 |  $Bnd_{CR} = (CR_i^{ls}(t_j), CR_{max})$ 
    |  $\backslash\backslash$ Execute Multi-objective Genetic Algorithm
    |  $\backslash\backslash$ Simpler dynamic programming can be applied here
6 |  $CR_i(t_j) = GAmultiobj(\{f_1, f_2\}, Cnstr, Bnd_{CR})$ ;
7 evaluate  $B_i(t_j) = |B_i(t_{j-1}) - CR_i^{cs}(t_j)E_i^{tr}(t_j) - E_i^{in}(t_j)|_+$ 
8 return  $CR_i(t_j)$ 

```

in the current interval t_j .

4.4.4 Evaluation

In this section, the performance of our adaptive compression algorithm is evaluated and compared with the two vanilla scenarios of periodic communication (a) without compression and (b) with only lossless compression in terms of battery life and reconstruction error. To simulate our approach, datasets consisting of high sample-rate pressure data recorded by 25 sensor nodes of our real world testbed [11] in a 170-day period were used. For the sake of brevity and due to the distributed nature of our system, this section presents the evaluation results for one pressure data stream.

Energy Parameters

The sensor node hardware platform we used is the Intel Edison development board [148], the same as used in WaterBox (see Chapter 3). XBee868 is used as communication module for its reliability and support for high sample rate over the distance range considered for our experiments, where it outperforms both LoRa and NWave (see Chapter 3). Specifically, we assumed

that each sensor node communicates with a base station in intervals of 1024 measurements. By having this infrastructure, Table 4.3 defines precisely the energy parameters of our system. Notice that the energy consumption because of the sensing $E_i^{in}(t_j)$ is relative to the communication one, $E_i^{tr}(t_j)$. This is a realistic setup of an actual sensor node in water industry while ,as the evaluation results reveal, emphasizes the importance of our approach. In this evaluation, K_p was selected heuristically to cover the needs of this specific application.

Table 4.3: Energy parameters

Parameter	Value	Description
B_{max}	57600000 mJoules	LiPo Battery 3200mAh
$E_i^{tr}(t_j)$	2439.20 mJoules	Transmission Energy Consumption per Measurement
$E_i^{in}(t_j)$	3056 mJoules	Sensing Energy Consumption
K_p	0.005	Battery gain

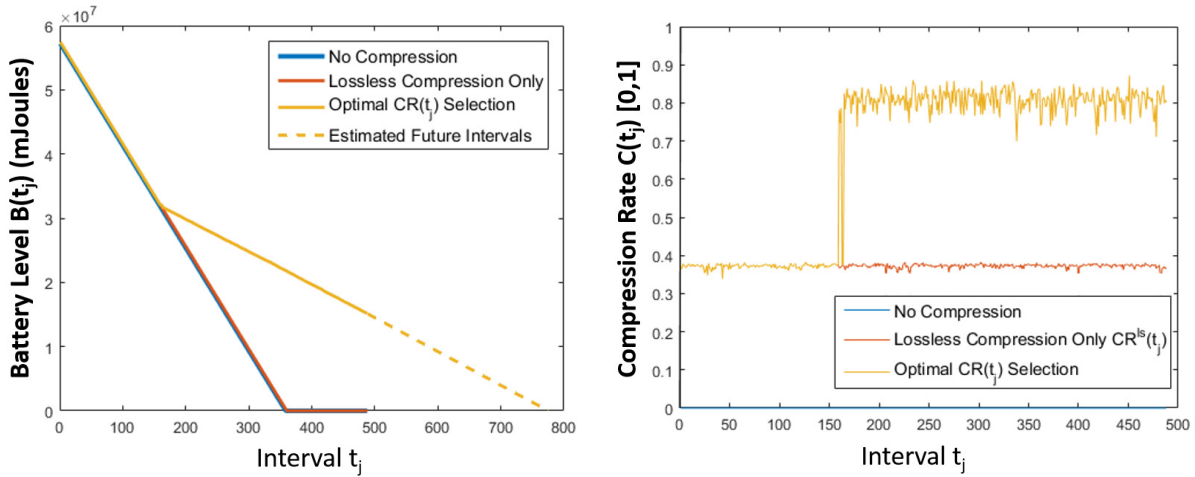
Multi-objective Genetic Algorithm Configuration

Table 4.4: Genetic algorithm parameters

Parameter	Value
Population type	Double vector with size 50 individuals
Selection function	Tournament
Reproduction crossover fraction	0.8
Migration forwarded fraction	0.2
Iterations	20
Distance measure function	Distance crowding
Pareto front population fraction	0.35
Stopping criterion	Maximum number of generations or the maximum stall time

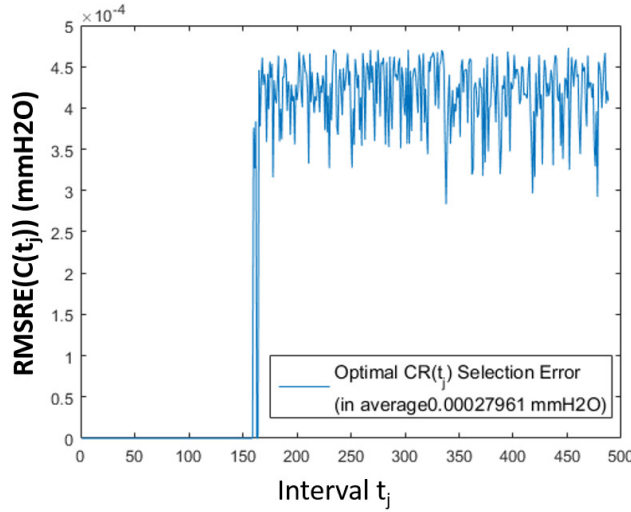
The Multi-objective Genetic Algorithm (MGA) is an iterative, global search, meta-heuristic method, which is being executed in every communication interval for which our algorithm selects compressive sensing as the optimal compression method. The parameters were configured with the same values for all executions. The initial population is randomly generated from the set of feasible solutions, based on the corresponding constraints and boundaries. Table 4.4 presents the evaluation setup of MGA.

Evaluation Results



(a) Battery level comparison per interval

(b) Compression rate comparison per interval



(c) RMSRE results per interval

Figure 4.22: Optimal compression selection evaluation results

This section presents the evaluation results of our optimal compression rate selection algorithm compared to the two vanilla scenarios (no compression and lossless compression only) in terms of battery and communication savings, and the reconstruction error because of compressive sensing. Figure 4.22a illustrates that battery life was extended by 46% and 37% for vanilla scenario (a) and (b) respectively. Additionally, based on Figure 4.22b, the communication was reduced around 66% and 29%. At this point, it is important to mention the impact of E^{in} to the energy savings. Despite the 66% communication reduction, the battery life has been extended only by 46%. The reason is the high energy requirements of the development board in sensing mode (i.e. E^{in}). With a different hardware, where the E^{in} is extremely low, the

Table 4.5: Savings compared to periodic communication (a) without compression and (b) with only lossless compression

Compared to	Optimal CR Selection Savings		
	Battery life	Communication	Error (RMSRE)
Without Compression	46%	66.37%	$2.80 \cdot 10^{-4}$
Only Lossless Compression	37%	29.15%	

battery life extension percentage would converge to the communication reduction percentage. The the same reason the two vanilla scenarios perform similarly in term of battery level. Due to the use of compressive sensing (lossy compression), our system introduces a reconstruction error. However, based on Figure 4.22c, the reconstruction error, RMSRE, is extremely low; on average $2.80 \cdot 10^{-4}$ mmH₂O⁵. Finally, Table 4.5 summarizes the aforementioned evaluation results.

4.5 Summary

In this chapter, we presented a variety of techniques which enable the edge processing, data reduction and incorporation between edge and offline systems aiming sensor node energy savings through communication and computation reduction. We firstly proposed an innovative approach that analyzes high sample rate data and detect anomalies by using compression rates rather than raw data. The main benefits of our system are: (a) the size of the program at run time, which can be applied in embedded systems, (b) data reduction (and proportional communication and energy costs) by 55%, (c) computation reduction by 98%, (d) the algorithm can be applied independently of the content of the raw data with an appropriate initial tuning, (e) the adaptation of our method to match trend/state changes in the raw data. We extend the system to be able to derive initialization parameters for self-configuration and to adjust said parameters as the nature of the underpinning data changes over time thus showing significant performance improvements over manual tuning by a further 15%. Future work of our approach is to examine the effect of changes in data precision (e.g. by using float instead of double values), to test our algorithm with other lightweight compression algorithms.

⁵Unit of water pressure. The operational water pressure fluctuates in tens scale.

Then, we combined the lightweight compression and anomaly detection with graph topology analytics for water distribution networks to create a burst detection and localization scheme. We show that our approach not only significantly reduces the amount of communications between sensor devices and the back end servers, but also can effectively localize water burst events by using the difference in the arrival times of the vibration variations detected at sensor locations. Our results can save up to 99% communications compared with traditional periodical reporting situations. Further our localisation can find the position of the anomaly for our particular scenario within 0.5m error. This data driven approach is significantly better than many hydraulic modelling approaches that at best identify a leak to the length of a given pipe which can be tens of metres [149].

To improve the edge data reduction, the power of compressive sensing was exploited for achieving high compression rates, yet high reconstruction accuracy, of real high sample rate data. The experimental evaluation revealed 35% better performance compared with well-established lossless compression algorithms. The current approach employs a fixed sparsifying transformation for the sensor data. However, the degree of sparsity can be increased by employing an adaptive sparse representation. To this end, the framework of dictionary learning [150] can be exploited, in conjunction with a joint sparsity assumption to account for correlations either between the windows of each individual data stream or between distinct data streams. Finally, a CS-based approach could be exploited for designing algorithms to detect abnormal sensor behavior by employing the random compressed measurements directly.

By having fully explored the benefits of the aforementioned data reduction techniques, we proposed an adaptive compression scheme that balances the tradeoffs of lossless compression and compressive sensing by minimizing the reconstruction (decompression) error, given the sensor node battery state. The evaluation of our proposed approach revealed a significant reduction of the communications between sensor devices and back-end server, at the order of 66%, in conjunction with an extension of the battery life by 46% compared the traditional periodic communication approaches.

In spite of the benefits of data reduction, edge processing and incorporation with off-line sys-

tems, the presented techniques make decisions locally to the edge without considering the state of other sensor nodes of the same network. Furthermore, the main goal of these techniques is to extend the battery life without considering energy harvesting capabilities. To address these problems, the next chapters will consider a scheduling system that both optimizes the decision making of communication along the network and proves sustainable operation of smart water networks.

Chapter 5

Sustainable & Reliable Scheduling for Smart Water Networks

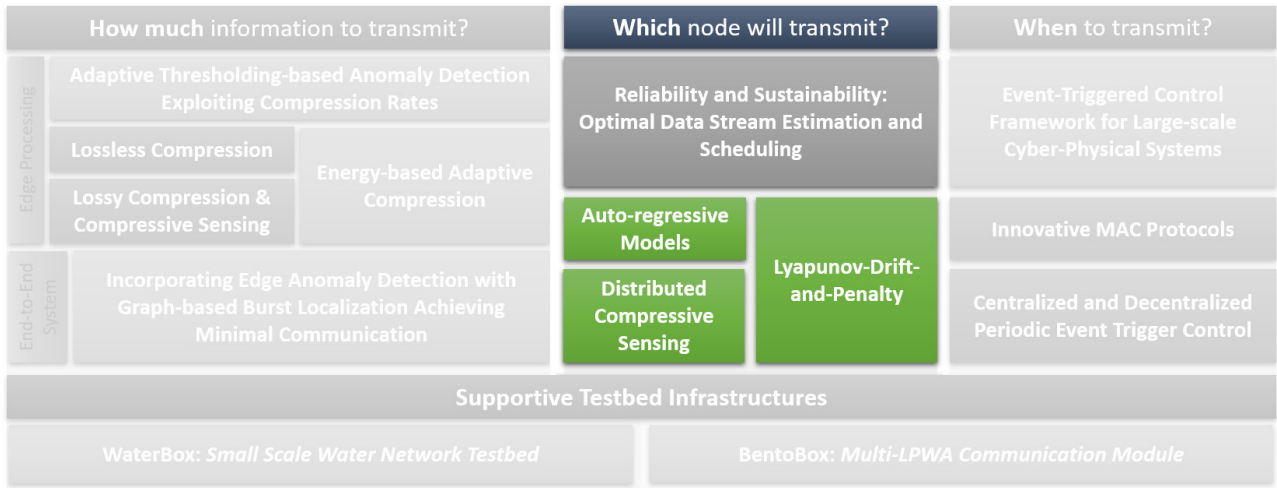


Figure 5.1: Chapter 5: Sustainable & Reliable Scheduling for Smart Water Networks.

In smart water networks, physical states such as water pressure and pipe vibration need to be sampled at high frequencies (e.g. more than 128 samples per second per sensor). This high frequency is required to capture instabilities or transient events of fast signals, like pressure signals propagated with sound velocity inside the pipes. Additionally, high sample-rated data can be exploited by server-side algorithms to generate high level of information precisely, such as the burst localization algorithm in [151], which localizes bursts with 0.5m accuracy.

However, to transmit the required high-speed raw sensor data streams through long-range

(several kilometers) wireless communications, high transmission power is required that lead to fast battery depletion. As a result, utility companies, who have already spent billions of dollars in water network maintenance, have to provide the supplementary expensive maintenance for frequently replacing batteries, which is considered unaffordable. Therefore, a sustainable smart water sensing system design for autonomous water network monitoring, which balances the communication, is highly desired.

Current smart water systems [2, 10] are unable to achieve the above objectives in a cost-effective way. For instance, the sustainable smart water system proposed by MIT [10] adopts large overground solar panels and direct power from lampstands, requiring expensive instrumentation and deployment costs; while other approaches, such as [152], require knowledge of the complete hydraulic models of water networks, which is computationally expensive and unable to adapt to water network system dynamics.

In this chapter, we propose a cost-effective and sustainable sensing system for the autonomous monitoring of system states and abnormal behaviors in water networks. In order to prove our approach, data from the 24 sensor/actuator nodes deployed to the inlet of equal number of DMAs in Bristol Water network were used (see Section 2.1). The sensor nodes record, analyze, and transmit high sample-rate pressure and flow data (up to 128 samples/sec) to a data processing center periodically, and cover an area of approximately 7,544 customer connections and 57km of pipe mains. A low-frequency (868 MHz) and high-power wireless communications was utilized (e.g. around several millijoules per kilobytes [153]) to support the long-range and high-speed sensor data stream transmissions. An additional capability of this large scale smart water distribution networks (WDN) is the remote control of DMA actuator components (i.e. valves) which optimize the water network performance and lifetime over varying demands. Because of underground power limitations, the main trend is the development and deployment of self-powered actuators. In the same system, self-powered multifunction network actuators for dynamically reconfigurable DMAs have been developed which integrate a Cal-Val 99-51 valve [154] and an energy generator that harvests energy (up to 0.7 W) from the water flow [114]. In addition to the large scale smart water system, we exploited the WaterBox infrastructure [155], as presented in Section 3.1, to evaluate the efficiency of our algorithms.

In order to achieve sustainable sensor node operation by utilizing the limited and dynamic harvested energy, we exploit the *in-node* anomaly detection algorithm based on lossless compression¹ of Section 4.1 to analyze and reduce the raw data volumes (and therefore the energy) required for the high-power transmission at the same time. To further reduce energy consumption, our system only requires to transmit a subset of the raw data streams, while other samples are estimated using auto-regressive models [18], based on strong correlations among sensor data streams observed from real data. Fundamentally, these correlations are based on the sound-velocity propagation of pressure signals inside the water network.

In recent years, advanced control methods have been considered in pressure management and pump scheduling tasks (e.g. [156]); all these require reliable information about the system states. However, this information is not always available, due to sensors' hardware malfunction or battery depletion, and data estimation is required. As a result, the unavoidable estimation error not only leads to unreliable sensor data analytics, but also the performance degradation of the smart water control system as a whole. Therefore, this chapter aims to maximize the accuracy of all estimated data streams by fully exploiting the dynamic energy-harvesting opportunities.

The operation of our system is as follows: initially the data processing center receives raw data and energy information from all sensor nodes in the network. After this, the data processing center uses the raw sensor data to establish a correlation graph of the network. This and the energy information are used to determine which nodes are now required to send raw data; this instruction is sent individually to those nodes only. For every subsequent time interval (we set this to 15 minutes in our system) all nodes are required to make note of their behaviours using anomaly detection results and energy metadata to the server. Therefore, unless the data processing center requests it, only the anomaly results and energy information are communicated to the server for each interval.

It can be seen that a fundamental question in our system is: *which are the best subsets of sensor nodes that should be requested to transmit their raw data, given the complex system dynamics*

¹Despite the fact that lossless compression is computationally more intensive than lossy compression, it constitutes a more reliable ground for anomaly detection. Therefore, lossless compression is adopted in this chapter for anomaly detection in order to keep the focus on the scheduling problem. The extension to lossy compression and compressive sensing is left for future work.

regarding the (arbitrary) stochastic processes of energy harvesting, energy consumption, and correlations among sensor data streams? We answer this question by formalizing a stochastic optimization problem to maximize the estimation reliability while ensuring the sustainable operation of systems, and solve the problem by developing a lightweight algorithm with strong theoretical guarantees to compute the best transmission nodes at real-time.

The specific contributions of this chapter are as follows:

- To reduce the energy consumption for long-range high-power data transmissions, we incorporate the *in-node* anomaly-detection algorithm of Section 4.1, which identifies the abnormal behaviours (e.g. bursts and leakages) in water networks and propose an estimation-based transmission solution by exploiting the properties of sound-velocity propagation of pressure signals inside the water network. In addition, our estimation is based on lightweight auto-regressive models, avoiding the use of complex hydraulic models [152].
- We formalize a stochastic optimization problem for the best selection of raw data transmissions, which aims to maximize the aggregated estimation reliabilities while guaranteeing a minimal reliability constraint and the sustainable operation of the smart water sensing system.
- We develop Data Transmission Scheduling (DTS), an asymptotically optimal solution to the formalized problem, based on Lyapunov optimization theory [19] as described in Section 2.4. Guided by the principles of DTS, we then propose FAST-DTS, a lightweight online algorithm that can adapt to Both DTS and FAST-DTS. Both DTS and FAST-DTS *do not* need to predict any future knowledge of the water network. Further, we *do not* make any stochastic/probabilistic assumptions regarding the system dynamics, which means that our approach is adaptive to arbitrary energy and correlation dynamics.
- Our work bridges the gap between data processing and resource allocation of wireless systems. To our knowledge, this is not only the *first* approach that adopts data stream estimation in smart water systems, but also the *first* scheduling approach based on data stream estimation in energy harvesting networks [21].

We evaluated our system by using 170-day water pressure data from the real smart water system as described in Section 2.1. During this process, we examined data reliability, energy waste, node lifetime, and transmission gaps by using three different algorithms where FAST-DTS outperformed them. Further evaluation was conducted by using and extending the hardware infrastructure of the small scale testbed, WaterBox [155] (see Section 3.1), to verify the adaptive behavior of the system in anomalies (i.e. leakages and bursts). The work presented in this chapter is a result of cooperation with Prof. Shusen Yang, who was responsible to formulate the stochastic optimization problem and provide the proofs of sustainability and optimality. My contribution was related to observe and analyze the physics and correlation of pressure streams in a water network; design the communication system, estimation technique, and the DTS algorithm; implement our approach either as simulation by using real data or in the WaterBox testbed and conduct the evaluation; and incorporate the edge anomaly detection of Section 4.1 to the new system.

5.1 Related Work

Currently, many researchers under the context of smart water networks [2, 10] focus on efficient sensing system design, while others [7–9, 152, 157] seek leakage detection solutions by developing new anomaly detection algorithms. However, none of them considers energy optimization nor data stream estimation. Similar to our work, [10] considers energy harvesting systems for water monitoring, but this work require expensive instrumentation and the deployment of solar panels and lampposts (using solar power at daytime, and lampposts at night). In contrast, our system harvests energy from water flows, resulting easy implementation, and continuous power supply (both day and night). More importantly, our scheduling algorithm (including power management) can ensure the sustainable operation of the system. In fact, there is an increasing interest in developing more efficient water flow energy harvester [158–160]. In addition, optimization for energy harvesting networks is an emerging hot research area [21, 79, 161–164]. A fundamental research issue for these approaches is to optimize the network performance (e.g. delay [161] and network utility [79, 164]) while ensuring the sustainable

operation of the network, by developing network algorithms such as routing and flow rate control. However, none of them considers smart water networks, nor the problem of estimation reliability maximization that we study in this chapter.

5.2 Preliminary

Pressure waves are generated at any point in a pipe system where a disturbance occurs because of flow rate change. An external disturbance could be a valve that is opening or closing, a pump that is started up or shut down, a change in reservoir pressure, or in the inflow or outflow for the system. A pressure wave, which represents a rapid pressure and associated flow change, travels at sonic velocity [165] by using the in-pipe liquid as a medium, and the wave is partially transmitted and reflected at all discontinuities in the pipe system (pipe junctions, pumps, open or closed ends, surge tanks, etc.).

The sonic speed, and consequently the delay of the pressure wave propagation, for a liquid flowing within a line are influenced by the elasticity of the line wall, pipe diameter and thickness, and restraint on longitudinal pipe movement. In addition to wave propagation delay, the effect of in-pipe friction causes attenuation in pressure wave, which is related to the length and material of the pipe. Specifically, the equation that describes the pressure wave attenuation/amplification between two individual points in terms of pressure head based on [166] is the following:

$$H_2 - H_1 = \frac{-fLQ^2}{2gDA_L^2} \quad (5.1)$$

Where Q is the water flow rate, L is the pipe length, f is the friction factor, D is the diameter, and A_L is the area of the pipe between the two points. By using well-studied equations like (5.1), the propagation delay and wave attenuation in a water network can be modeled accurately. The modeling normally involves the physical properties of different pipes and network topology and this would allow the estimation of the pressure of each interconnected point in a water network using raw data only from one point. However, the modeling task is unfeasible because of the size of a real water network and impossible because of unpredictable dynamic changes

(e.g. new asset installation and network expansion) or anomalies (leakages and bursts) in the water network. For these reasons, water network monitoring systems have required the installation of battery driven sensor nodes (of which 99% are underground) at least on the main pipes, junctions and valves.

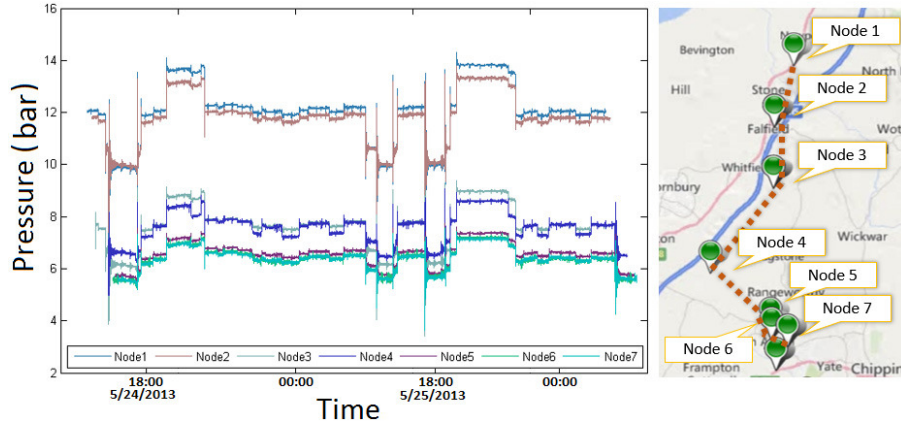


Figure 5.2: Pressure wave propagation delay and attenuation of sequential nodes

One of water network analysts' main interests lie in this system assisting them to predict leakages and network problems. This is focused on the observation of pressure wave behavior and transformations at discontinuities; so called transient events. Because of the high velocity, the analysis of pressure waves requires relatively high sample-rate pressure data from the water network. Figure 5.2 represents high sample-rate pressure data from seven sequential nodes of the water network of Section 2.1, which allows transient event analysis and the observation of propagation delay and the signal attenuation.

5.3 System Overview and Modeling

As shown in Figure 5.3, a smart water sensing system consists of a set of sensor nodes $\mathcal{S}$ and a data processing center. Continuous physical time is divided into discrete intervals $t = \{1, 2, \dots, t_{\text{end}}\}$ (e.g. a default interval is 15 minutes in our system), where the time-horizon of the smart water system t_{end} can be any large but finite value (e.g. several years).

experiments in the small scale testbed that anomalous behaviors also will have a significant impact on the correlation strength among data streams, which further affects estimation reliability. Therefore, the combination of data estimation and anomaly detection are central for our overall system design. We discussed the anomaly detection algorithm in detail in Section 4.1.

Wireless Communication

Water networks cover city scale areas and most of the assets, i.e. sensor nodes, are deployed underground. As a result, long range wireless communication technologies are unavoidable either to cover long distances or to increase signal penetration. For this reasons, the contemporary Low Power Wide Area (LPWA) technologies [167], which are mostly single hop, can be considered as the most appropriate communication approaches. The current chapter considers a single hop communication infrastructure based on LPWA technologies and uses the experimental results of Section 3.2 and [153] to setup the evaluation parameter, such as communication energy consumption. Due the high sample rate and range requirements of the current system, we consider the power hungry XBee868 as the most appropriate technology to stress-test our approach against the communication technique of LoRa or NWave (see Section 3.2). Therefore, the parameters of XBee868 [168] were used which implements the Zigbee protocol for long ranges over 868MHz.

At this point, the timing limitations of LPWA technologies are important to be mentioned. Based on experimental result in [153], the tradeoff of implementing a reliable long range communication based on low frequency, i.e. 868 MHz, is the decrease of data rate. In our system, every sensor node transmits 60 kBytes of water raw data to a data processing center, while the data rate of XBee868 is 24kbps. Thus, the required time is around 15sec. By the fact that data processing centers are highly capable hardware infrastructures, the required time for decision making processes (i.e. scheduling) is at least one to two orders of magnitude less than the data transmission.

At each interval, each sensor node is required to transmit its *Node States* information to the data processing center, which includes (a) the number of anomalies within this particular

interval denoted as the **Anomaly Counter**; (b) its energy related parameters which will be discussed later soon.

According to its transmission information, each sensor node $i \in \mathcal{S}$ can provide two communication states at each interval t : either: (a) transmit both raw data stream and node states, or (b) transmit node states only. We use a binary variable $y_i(t) = 1$ and $y_i(t) = 0$ to denote these respectively.

Power Management

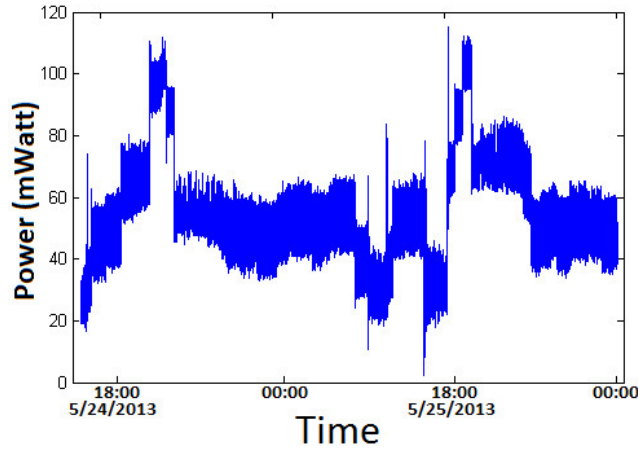


Figure 5.4: Dynamic energy generated by water flow.

Each sensor node i can harvest $h_i(t)$ amount of energy from the water flow [114] at each interval t . As shown in Figure 5.4, $h_i(t)$ is time-varying due to the dynamic water flow rate in the water network. The harvested energy can either be consumed by in-node operations (i.e. sensing and computation) and wireless transmission; or be stored in the battery, which is modelled as an energy queue

$$0 \leq B_i(t) \leq B_{\max} \quad (5.2)$$

where $B_{\max}$ is the battery capacity. In our system, $B_{\max} = 60.2$ KJ (two 3.6V batteries with capacities of 2.4Ah each).

Considering the communication states $y_i(t)$ of each node i , the queueing dynamic of the battery at each sensor node $i \in \mathcal{S}$ is modelled as

$$B_i(t+1) = |B_i(t) - y_i(t)E_i^{tr}(t) - E_i^{in}(t)|_+ + h_i(t) \quad (5.3)$$

where for any real number x , $|x|_+ = 0$ if $x \leq 0$, $|x|_+ = x$ otherwise. $E_i^{tr}(t)$ and $E_i^{in}(t)$ represent the energy costs for transmission and all in-node operations (including sensing, compression, and anomaly detection), which are expected to change over time due to the system dynamics such as time-varying channel quality etc. It is worth noting that since the energy cost for wireless transmission in state $y_i(t) = 0$ is minimal (its several bytes for node states transmission), we only consider energy consumption caused by in-node operations in this state.

5.3.2 Data Processing Center

Based on the node states and raw data streams received from the sensor nodes, the data processing center is responsible for performing the tasks of correlation graph updates, data stream estimation, and raw data transmission scheduling.

Correlation Graph Updates

Due to the water network interconnections, neighboring sensor nodes provide high correlated data (i.e. in Figure 5.2). The data correlation among different nodes may change either for predictable or unexpected reasons within a water network. Changes to the water network topology because of the control of water network assets, i.e. valves, or the physical changes from engineers, such as isolation of pipes or areas for maintenance, can be considered predictable. Due to the slow physics (e.g. demand changes) of water networks, these actions can occur in the range of some hours to days resulting in the rare correlation graph update. On the other hand, anomalies, such as leakages or bursts are unexpected. By exploiting, the anomaly detection at the edge and the a-prior knowledge of water network topology changes, the proposed system can minimize the correlation graph (see Figure 5.5) updates accordingly.

Specifically, the data processing center computes the time-varying correlation graph $G(\mathcal{S}, \mathcal{L}(t))$, where $\mathcal{L}(t)$ represents the set of all current correlation links at each interval t . A symmetric

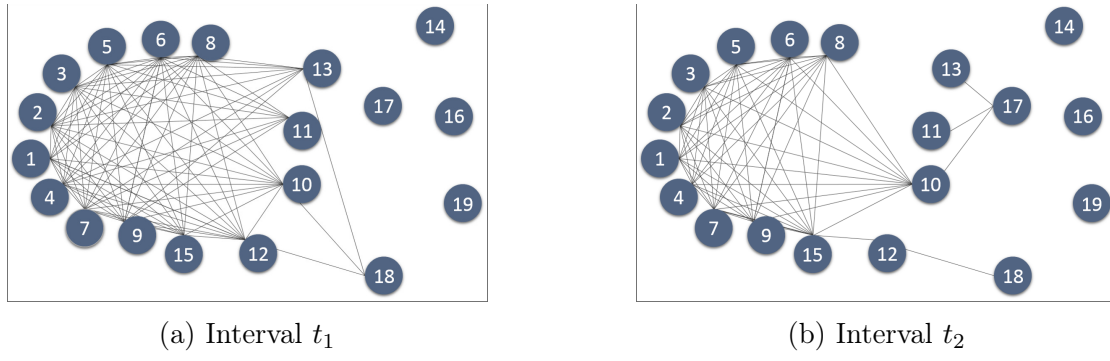


Figure 5.5: The time-varying correlation graph topology at two intervals t_1 and t_2 .

correlation link (i, j) is considered to exist in $\mathcal{L}(t)$, if the current Pearson correlation coefficient $c_{i,j}(t)$ of the raw data streams of sensor nodes i and j are larger than 95%², i.e.

$$\mathcal{L}(t) := \{(i, j) : c_{i,j}(t) \geq 95\%, i, j \in \mathcal{S}\} \quad (5.4)$$

An example of a time-varying correlation graph of our sensor nodes is illustrated in Figure 5.5. For any given sensor $i \in \mathcal{S}$, we define its temporary correlation neighbor set as

$$\mathcal{N}_i(t) := \{j : j \in \mathcal{S}, (i, j) \in \mathcal{L}(t)\} \quad (5.5)$$

i.e. the sets of sensors that currently correlated with i .

The data processing center updates the correlation graph in real-time, *whenever a fresh raw data stream is received*.

Anomaly Counter Outlier Processing

In smart water systems, two streams of pressure data are strongly correlated if their anomaly patterns are similar, observed from real pressure data streams. This is because any change of pressure (e.g. due to the change of the pressure regulator state) is diffused along the water network evenly. In the case when a burst or leakage (or other types of anomalies) occurs, a subpart of the water network is influenced and consequently its correlation. We use the anomaly counter of each node at every interval as an index for anomaly patterns, to detect the sudden

²95% is chosen heuristically based on the reliability results of the estimation models in our system.

change of the correlation graph.

After receiving the anomaly counter values of all sensor node for an interval, the data processing center will compute if a high deviation (outlier) of anomaly counters occurs at each sensor node, based on Chauvenet's criterion [56]. If an outlier occurs at a sensor node, say i , the correlation pattern between this node and all its current correlation neighbors in $\mathcal{N}_i(t)$ may be changed. In this case, the data stream of i cannot be accurately estimated and the correlation graph should be updated as soon as possible. Therefore, the system will **enforce** sensor i to transmit its raw data stream for this interval to update the correlation graph.

Raw Data Transmission Scheduling

At each interval t , the set of all sensors $\mathcal{S}$ can be divided into two subsets $\mathcal{S} = \mathcal{S}_a(t) \cup \mathcal{S}_b(t)$, according whether they are forced to transmit their raw data streams or not:

- $\mathcal{S}_a(t)$ (**Enforce Transmission**): Each sensor $i \in \mathcal{S}_a(t)$ is enforced to transmit its raw data at t in the following three cases: (1) fresh raw data from i is required to update the correlation graph, (2) node i has no correlation neighbor at t , and (3) an anomaly counter outlier occurs at i . This is because when one of three cases happens, neither i 's own data stream nor other data streams can be reliably estimated based on the raw data of i . It is obvious that $y_i(t) = 1, \forall i \in \mathcal{S}_a(t)$.
- $\mathcal{S}_b(t)$ (**Transmission Scheduling**): Each sensor node $i \in \mathcal{S}_b(t)$ is not enforced to transmit its raw data, i.e. $y_i(t) \in \{0, 1\}, \forall i \in \mathcal{S}_b(t)$. Given $\mathcal{S}_b(t)$, our scheduling algorithm DTS and FAST-DTS will optimize the transmission decision, i.e. which subset of sensors in $\mathcal{S}_b(t)$ should transmit their raw data at interval t , based on the estimation reliability (formally defined later) and the energy states of i (including battery level, harvested energy, and energy consumption). This will be discussed in detail in Section 5.5.

Let a set of sensor nodes $\mathcal{Y}(t) \subseteq \mathcal{S}$

$$\mathcal{Y}(t) := \{i : i \in \mathcal{S}, y_i(t) = 1\} \quad (5.6)$$

to represent the *Transmission Scheduling Decision* of the whole system at interval t .

Data Estimation and Estimation Reliability

The data stream of a sensor node i can be estimated by its one or multiple correlation neighbors in $\mathcal{N}_i(t)$. For each sensor i , define its data estimation reliability as a real number between 0 and 1. Specially, if i transmits its raw sensor data stream at interval t , its data reliability is 1. Otherwise, its data should be estimated based on the raw data stream transmitted by the other sensors correlated with it, resulting in a data reliability less than 1. Therefore, we can formalize the data reliability of each sensor node $rlb_i(\mathcal{Y}(t))$ as a function of the scheduling decision $\mathcal{Y}(t)$. The details of data stream estimation and computation of $rlb_i(\mathcal{Y}(t))$ is presented in Section 5.6. It can be seen that $rlb_i(\mathcal{Y}(t)) = 1, \forall i \in \mathcal{Y}(t)$, because every sensor in set $\mathcal{Y}(t)$ is scheduled to transmit its raw data at interval t .

t_{end}	The finite time-horizon of the system.
$\mathcal{S}$	The set of all sensor nodes
$G(\mathcal{S}, \mathcal{L}(t))$	The time-varying correlation graph
$\mathcal{N}_i(t)$	Correlation neighbor set of sensor i at interval t
$\mathcal{S}_a(t)$	Enforced transmission sensor set at interval t
$\mathcal{S}_b(t)$	Non-enforced transmission sensor set at interval t
$y_i(t)$	Sensor i 's raw data transmission state at interval t
$\mathcal{Y}(t)$	Scheduling decision at interval t
$rlb_i(\mathcal{Y}(t))$	Data estimation reliability of sensor i at interval t
$rlb_{\min}$	minimal estimation reliability requirement
$B_i(t)$	Battery level of sensor i at interval t
$B_{\max}$	Battery capacity
$h_i(t)$	Harvested energy by sensor i at interval t
$E_i^{tr}(t)$	Sensor i 's transmission energy cost at interval t
$E_i^{in}(t)$	Sensor i 's in-node operation energy cost at t
B_{exp}	Parameter for sustainability/reliability tradeoff
V	Parameter for sustainability/reliability tradeoff

Table 5.1: Frequently used symbols

Table 5.1 summarizes the frequently used symbols in this chapter.

5.4 Anomaly Definition & Detection

In a water network, an anomaly is defined as an abrupt fluctuation in pressure data. An anomaly can be caused by: (a) a disturbance occurrence in the flow rate in the water network

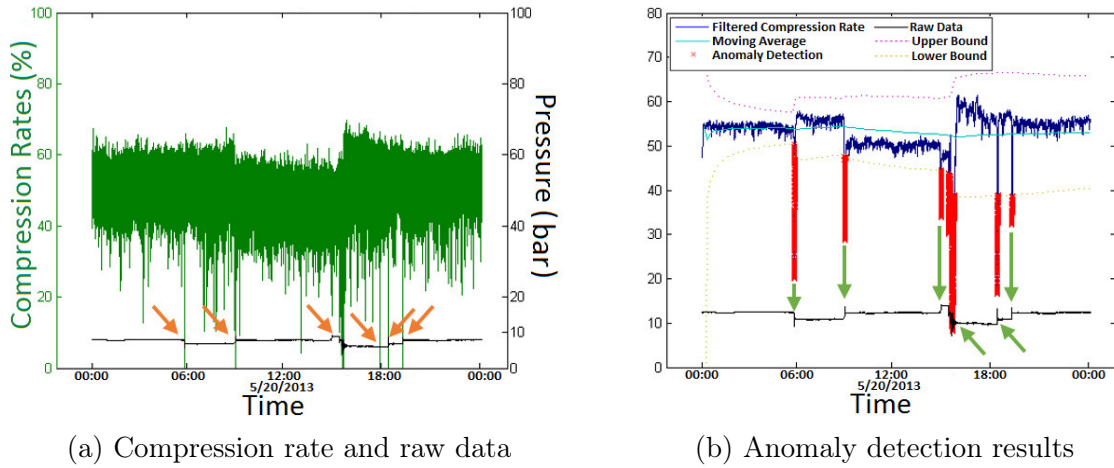


Figure 5.6: Observed correlation between pressure fluctuation and compression rates.

(e.g. leakage or burst), and (b) a hardware failure in the sensor node. In this chapter, we exploit the memory-efficient algorithm of Section 4.1 to perform in-node anomaly detection which is based on the following observation:

Observation 1. *A strong correlation exists between data value fluctuation and the compression rates of several compression algorithms such as S-LZW-MC [129] and miniLZO [128].*

Intuitively, this observation holds because lossless compression algorithms reduce data volume by merging similar bytes within a packet. In the case of a stable system, the retrieved measurements fluctuate within a small range and around a certain value. This increases the similarity of the bytes in a packet, and consequently raises the compression performance. When an anomaly happens, the consecutive measurements deviate dramatically, resulting in the incompressibility of the streams. Figure 5.6(a) shows an example of our observation. Note that the original water pressure data is also overlaid on the same graphs (lower line). It is clear from Figure 5.6, that these traces highlight data anomalies as indicated by the arrows. After multiple dataset evaluations, we formed the hypothesis that we could use the correlation of the compression rate and raw data. From water technician logs, we observed that the anomalies were valve position changes which were used to simulate water bursts, causing significant pressure data fluctuation. At these points, the compression algorithm is unable to compress the data so the compression rate falls to 0%. In Figure 5.6, the drop in compression rate isolates the areas of raw data where

the fluctuation pattern is changeable. We modified the algorithm of Section 4.1 to return not only the timestamps of the observed anomalies but also to count the detected anomalies over a period of time.

5.5 Sustainable Scheduling for Raw Data Transmission

In this section, we develop a theoretically optimal scheduling algorithm, DTS, which maximizes the estimation reliability while ensuring sustainable operation of the smart water sensing system. By exploiting the analytical behaviors of DTS, we then develop FAST-DTS, a lightweight (linear complexity) and adaptive algorithm, to make the transmission scheduling decision at real time in practical smart water sensing systems.

5.5.1 Monotonically Non-Decreasing Estimation Reliability

Based on all data collected from a real smart water sensing system, we observe that the estimation reliability $rlb_i(\mathcal{Y}(t))$ has the following property:

Observation 2. For any given sensor $i \in \mathcal{S}$, consider two scheduling decisions $\mathcal{Y}(t)$ and $\mathcal{Y}'(t)$, where $\mathcal{Y}'(t) = \mathcal{Y}(t) \cup \{i\}$. We have

$$rlb_j(\mathcal{Y}'(t)) \geq rlb_j(\mathcal{Y}(t)), \forall j \in \mathcal{S} \quad (5.7)$$

This observation demonstrates that adding any raw sensor data stream $i \in \mathcal{S}$ will result in a equal or larger (i.e. **a monotonically non-decreasing**) estimation reliability for all sensor data streams $j \in \mathcal{S}$. An example of this observation is illustrated in Figure 5.7, which is based on the sensor node topology shown in Figure 5.2.

Inequality (5.7) is a very important property that will be used in the scheduling decision making.

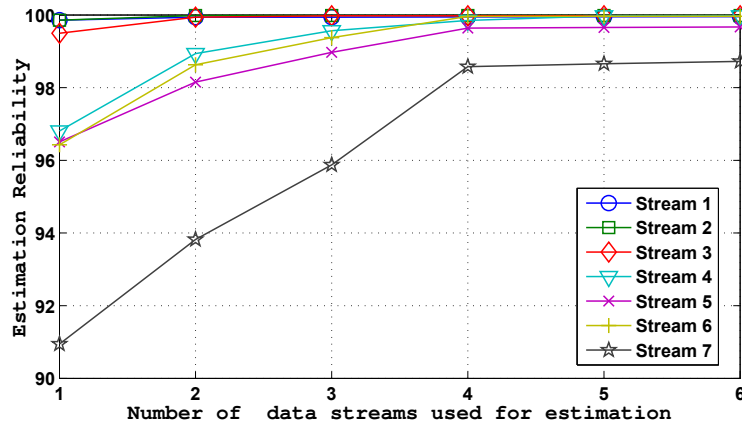


Figure 5.7: results of linear regression for 1 day training set and combination size

5.5.2 Stochastic Scheduling Optimization

The optimal subset of transmission nodes $\mathcal{Y}(t)$ at every interval t can be obtained by solving the following stochastic optimization problem:

$$\max_{\mathcal{Y}(t)} \quad \frac{1}{t_{\text{end}}} \sum_{t=1}^{t_{\text{end}}} \sum_{i \in \mathcal{S}} U_i(\text{rlb}_i(\mathcal{Y}(t))) \quad (5.8)$$

$$\text{subject to} \quad \forall i \in \mathcal{S}_a(t), \forall t, y_i(t) = 1, \text{rlb}_i(\mathcal{Y}(t)) = 1 \quad (5.9)$$

$$\forall i \in \mathcal{S}_b(t), \forall t, y_i(t) \in \{0, 1\}, \text{rlb}_i(\mathcal{Y}(t)) \geq \text{rlb}_{\min} \quad (5.10)$$

$$\forall i \in \mathcal{S}, \forall t, B_{\max} \geq B_i(t) > 0 \quad (5.11)$$

$$\forall i \in \mathcal{S}, \frac{1}{t_{\text{end}}} \sum_{t=1}^{t_{\text{end}}} (h_i(t) - y_i(t) E_i^{\text{tr}}(t) - E_i^{\text{in}}(t)) \geq 0 \quad (5.12)$$

where the utility function $U_i(\text{rlb}_i(\mathcal{Y}(t)))$ in the objective (5.8) can be any convex, non-decreasing, and differentiable function of $\text{rlb}_i(\mathcal{Y}(t))$. For instance, a utility function of

$$U_i(\text{rlb}_i(\mathcal{Y}(t))) = \ln(\text{rlb}_i(\mathcal{Y}(t))) + \varepsilon \quad (5.13)$$

with some small value $\varepsilon > 0$ can achieve approximate proportional fair reliability among sensor data streams [169]. Constraint (5.9) highlights the sets of sensors that must transmit their archived data. Constraint (5.10) states the minimal correlation requirements. Constraint (5.11) ensures that each node is not allowed to run out of battery energy or exceed its finite battery capacity. Finally, constraint (5.12) demonstrates that the long-term average of consumed energy

for each sensor node should not be more than that of the harvested energy.

In summary, we can see that problem (5.8)-(5.12) aims to maximize the long-term average estimation reliability of all sensor data streams with the consideration of fairness, while ensuring the sustainable operation for the long-term sensing of smart water systems.

5.5.3 Data Transmission Scheduling (DTS)

In this subsection, we develop an adaptive scheduling algorithm, Data Transmission Scheduling (DTS), which makes the optimal transmission decision $\mathcal{Y}(t)$ to solve the stochastic optimization problem (5.8)-(5.12) in real time. To derive the benefit of Lyapunov Drift-Plus-Penalty method as described in Section 2.4, we transformed the problem accordingly. Then, at the beginning of each interval $1 \leq t \leq t_{\text{end}}$, the DTS algorithm operates as follows

1. Compute the optimal scheduling decision $\mathcal{Y}(t)$ by solving the following problem:

$$\max_{\mathcal{Y}(t)} \sum_{i \in \mathcal{S}} (V U_i(\text{rlb}_i(\mathcal{Y}(t))) + \beta_i(t) y_i(t)) \quad (5.14)$$

$$\text{subject to } \text{rlb}_i(\mathcal{Y}(t)) \geq \text{rlb}_{\min}, \forall i \in \mathcal{S} \quad (5.15)$$

where $V > 0$ is system parameter that balances the trade-off between the system reliability performance and the risk of battery depletion, and

$$\beta_i(t) = E_i^{\text{tr}}(t)(B_i(t) - B_{\text{exp}}) \quad (5.16)$$

Here, the *expected battery level* $0 < B_{\text{exp}} < B_{\text{max}}$ is a system parameter that aims to balance the harvested energy utilization and risk of battery depletion. We will discuss the practical settings of V and B_{exp} in detail in the next subsection.

2. Update the battery of each sensor node i using Eq.(5.3). If a battery overflow occurs, i.e. $B_i(t+1) > B_{\text{max}}$, then set $B_i(t+1) = B_{\text{max}}$.

Due to the Lyapunov Drift-Plus-Penalty formulation, it is worth noting that the DTS algorithm *only requires* current system knowledge, which can be obtained easily at real time, and *does not* need to predict the information of future intervals, such as future energy harvesting opportunities and data stream correlations.

5.5.4 Sustainability & Optimality Proof

The analytical results summarized by Theorem 5.1 below will demonstrate that the DTS algorithm has strong performance guarantees, and provides guiding principles for setting system parameters V and B_{exp} in practice.

Theorem 5.1. *DTS algorithm has the following performance guarantees.*

1. **Sustainable Operation.** *By setting*

$$V \leq V_{\text{threshold}} = \frac{\sum_{i \in \mathcal{S}} (U_i(1) - U_i(0))}{E_{\text{max}} B_{\text{exp}}} \quad (5.17)$$

DTS will make a real-time scheduling decision $\mathcal{Y}(t)$ such that every node i will not transmit its raw data stream when its battery level $B_i(t) < B_{\text{exp}}$, unless its reliability constraint is not satisfied, i.e. $\text{rlb}_i(\mathcal{Y}(t)) < \text{rlb}_{\text{min}}$. Here E_{max} is the upper bound of energy costs for transmission for all intervals (e.g. $\forall t, i, E^{\text{tr}}(t) \leq E_{\text{max}}$), which is a constant value depending on the wireless transceiver adopted by the sensing device.

2. **Asymptotic Optimality.** *As $V \rightarrow +\infty$, DTS asymptotically achieves the optimal solution to the stochastic optimization problem (5.8)-(5.12).*

Proof. Theorem 5.1 can be proved by using sample-path based Lyapunov optimization theory, which can be found in the Appendix B. □

We first discuss the impact of V on the behaviors of DTS algorithm. With a fixed B_{exp} , a larger V will result in better reliability performance, but more aggressive energy consumption behavior. Especially, when V is larger than the threshold $V_{\text{threshold}}$ defined by the right-hand

side of inequality (5.17), a sensing device would be forced to transmit its raw data, even when it has a low battery energy level i.e. smaller than the B_{exp} . In order to achieve sustainable operation, we can set $V = V_{\text{threshold}}$ in practice.

The expected battery level B_{exp} also has a significant impact on the DTS performance. By setting a larger B_{exp} , the risk of battery depletion will be lower, but the probability of battery overflow will be higher, resulting in lower harvested energy utilization. In addition, larger B_{exp} will lead to a small $V_{\text{threshold}}$, and therefore, lower reliability performance. In practice, we can set B_{exp} according to the long-term energy harvesting and consumption behaviors of the smart water system. For instance, if the correlation graph is always dense, overtime (good estimation opportunity and low energy consumption), we can use a small B_{exp} to improve the system performance. However, if the correlation graph is sparse or evenly disconnected within some durations (i.e. poor estimation opportunity and large energy consumption to ensure constraint (5.15) within such durations), we need to set a large B_{exp} to reduce battery depletion risk within such durations.

5.5.5 NP Completeness of the DTS

At each interval t , DTS needs to compute the optimal $\mathcal{Y}(t)$ to solve the deterministic optimization problem (5.14)-(5.15), which looks very simple but may introduce extremely intensive computation. In the worst case, when $\mathcal{S}_b(t) = \mathcal{S}$ and the correlation graph $G(\mathcal{S}, \mathcal{L}(t))$ is a complete graph (i.e. all sensor data streams are strongly correlated with each other), the number of all possible scheduling decisions is $2^{|\mathcal{S}|} - 1$. This is because that $\mathcal{Y}(t) \subseteq \mathcal{S}$, and all possible non-empty subsets of $\mathcal{S}$ should be considered as the best one that solves problem (5.14)-(5.15).

Theoretically, we have

Theorem 5.2. *Problem (5.14)-(5.15) is NP complete.*

Proof. Since the variable $y_i(t)$ is a binary variable for each sensor node i , i.e. $y_i(t) \in \{0, 1\}, \forall i$, Problem (5.14)-(5.15) is a mixed 0-1 programming problem, which is well-known to be NP complete in general [170]. $\square$

As a result, although the DTS algorithm can guarantee an asymptotically optimal solution to problem (5.8)-(5.12), computing the optimal scheduling policy at each interval is prohibitive for large-scale smart water sensing systems. To address this issue, we develop FAST-DTS, an approximate algorithm of DTS.

5.5.6 FAST-DTS A Linear Approximation of DTS

ALGORITHM 3: Pseudocode of FAST-DTS at every interval t .

Input : $B_{\text{exp}}, \mathcal{S}_a(t), \mathcal{S}_b(t)$

Output: Scheduling Decision $\mathcal{Y}(t)$

A: Transmission Scheduling

```

1  $\forall i \in \mathcal{S}_a(t), y_i(t) \leftarrow 1;$ 
2  $\forall i \in \mathcal{S}_b(t), y_i(t) \leftarrow 0;$ 
3  $\mathcal{Y}(t) \leftarrow \mathcal{S}_a(t);$ 
4 forall  $i \in \mathcal{S}_b(t)$  do
5   if  $B_i(t) > B_{\text{exp}}$  then
6      $y_i(t) \leftarrow 1; \mathcal{Y}(t) \leftarrow \mathcal{Y}(t) \cup \{i\};$ 
7 forall  $i \in \mathcal{S}_b(t)$  do
8   if  $rlb_i(\mathcal{Y}(t)) < rlb_{\min}$  then
9      $y_i(t) \leftarrow 1; \mathcal{Y}(t) \leftarrow \mathcal{Y}(t) \cup \{i\};$ 
10 return  $\mathcal{Y}(t);$ 

```

B: Battery Energy Level Update

```

11 forall  $i \in \mathcal{S}$  do
12    $B_i(t+1) \leftarrow |B_i(t) - y_i(t)E_i^{tr}(t) - E_i^{in}(t)|_+ + h_i(t);$ 
13    $B_i(t+1) \leftarrow \min(B_{\max}, B_i(t+1));$ 

```

FAST-DTS has a linear complexity of $O(|\mathcal{S}|)$, based on the discussions of practical parameter settings presented in the last subsection. The pseudo code of FAST-DTS is shown in Algorithm 3. From Algorithm 3, we can see that FAST-DTS implicitly sets the parameter $V = V_{\text{threshold}}$: If the battery level of a node is higher than B_{exp} , it will be scheduled to transmit its data stream at current interval (lines 04-06, part A); otherwise, it will not transmit unless its reliability constraint is not satisfied (lines 07-09, part A).

The operations of DTS and FAST-DTS are different in the case where the reliability constraints for the nodes with battery levels lower than B_{exp} are not satisfied. In this case, DTS will maximize objective (5.14) by considering all possible scheduling policies (which introduces

exponential worst-case complexity), while FAST-DTS simply forces these nodes to transmit one by one until all of their reliability constraints are satisfied, according to the observed property (5.7). Although this aggressive energy consumption behavior of FAST-DTS would result in slightly higher risk of battery depletion, the system reliability performance will be better. More importantly, the computational complexity would be dramatically reduced, which enables the real-time application of FAST-DTS for large-scale smart water sensing systems.

5.6 Data Stream Estimation

In this section, we discuss how to estimate the data stream of a given sensor node i based on the a given set of sensors $\mathcal{Y}(t) \subseteq \mathcal{S}$, and compute the estimation reliability $rlb_i(\mathcal{Y}(t))$.

Our estimation is based on multiple auto-regressive models [18], such as linear, quadratic, pure-quadratic, and polynomial model. For simplicity, we use linear multiple regression models to discuss our estimation approach. Let two vectors $\mathbf{x}_i(t)$ and $\hat{\mathbf{x}}_i(t)$ be the raw and estimated data streams of sensor i at interval t respectively. The fitted multiple linear regression models have the form:

$$\hat{\mathbf{x}}_i(t) = b_0(w) + b_1(w)\mathbf{x}_1(t) + \dots + b_n(w)\mathbf{x}_n(t) \quad (5.18)$$

where $n = |\mathcal{Y} \cap \mathcal{N}_i(t)|$ is the number of raw data streams that are used for the estimation, because only the correlated neighbors of sensor i can be used to estimate its data; $b_0(w), \dots, b_n(w)$ are the regression coefficients trained based on the received data streams with a window size of w ; $\mathbf{x}_1(t), \dots, \mathbf{x}_n(t)$ are raw data streams of i 's correlation neighbors in $\mathcal{Y} \cap \mathcal{N}_i(t)$.

Let a $n+1$ dimensional vector $\mathbf{b}(w) = (b_0(w), \dots, b_n(w))$ represent all the regression coefficients. The Sum of Squares of the Error (SSE)

$$SSE(\mathbf{b}(w)) = (\|\mathbf{x}_i(t) - \hat{\mathbf{x}}_i(t)\|_2)^2 \quad (5.19)$$

is a function of $\mathbf{b}(w)$, where $\|\cdot\|$ is the 2-norm operator. Therefore, it is easy to compute the

optimal regression coefficients

$$\mathbf{b}^{opt}(w) = \arg \min_{\mathbf{b}(w)} SSE(\mathbf{b}(w)) \quad (5.20)$$

We use the coefficient of determination to measure the reliability of the estimated data stream $\hat{\mathbf{x}}_i(t)$ based on $\mathcal{Y}$:

$$r_{lb_i}(\mathcal{Y}(t)) = 1 - \frac{SSE}{SST} = 1 - \frac{(\|\mathbf{x}_i(t) - \hat{\mathbf{x}}_i(t)\|_2)^2}{(\|\mathbf{x}_i(t) - \bar{\mathbf{x}}_i(t)\|_2)^2} \quad (5.21)$$

Where the total corrected sum of squares $SST = (\|\mathbf{x}_i(t) - \bar{\mathbf{x}}_i(t)\|_2)^2$ and $\bar{\mathbf{x}}_i(t)$ is the mean of all values in vector $\mathbf{x}_i(t)$.

5.7 Evaluation

In this section, we evaluate the performance of our FAST-DTS algorithm by using 170-day water pressure (128 samples per second) and energy harvesting data from our real smart water system, which consists of 24 sensor nodes. Additionally, we verify the adaptive behaviour of the system in anomalies by using and extending a small-scale testbed, WaterBox [155]. The hardware infrastructure of WaterBox is based on Intel Edison which retrieves high sample rate pressure, flow, temperature, and energy consumption data. In spite of the capability of battery connection, the sensor nodes are directly connected to power and the battery is virtualized to ensure the stability of experiments.

5.7.1 Evaluation Set-up

Since no prior related algorithm exists, we compare the performance of our FAST-DTS (FDTS) with the following three naive scheduling algorithms:

- **Reliability-Greedy(RG):** All the sensor nodes transmit data at every interval to achieve the highest reliability. Aforementioned state of the art solutions [10, 13, 14] exploits this algorithm.

- **Energy-Greedy (EG m):** The m sensor nodes with the highest battery levels transmit raw data.
- **Round Robin (RR m):** The selection of sensor nodes is shifted by m nodes at every interval.

In all evaluations, we set the duration of an interval and training window size as 15 (commonly used by utility companies) and 60 minutes (heuristically found) respectively. In order to perform a strict benchmark in terms of energy consumption, we decide to use smaller batteries with $B_{\max} = 500\text{J}$ instead of the real battery used in our smart water system (i.e. 60.2 kJ). Based on WaterBox sensing device hardware [155] and the real experiments of LPWA communication technologies as conducted in [153], the energy consumption for in-node operations $E_i^{\text{in}}(t)$ and $E_i^{\text{tr}}(t)$ for each node i at every interval t are set as random variables with averages of 6 J and 30 J respectively.

5.7.2 A Case Study

In this subsection, we present a simple case study of 7 highly-correlated sensor data streams with fixed system parameter settings ($B_{\text{exp}} = 150$ and $\text{rlb}_{\min} = 0.98$) to illustrate the behaviour of the above algorithms. For brevity, the following results refer to one set of highly correlated sensor nodes³, while $m = 3, 6$ for Energy-Greedy (EG3, EG6) and Round Robin (RR3, RR6).

To examine data reliability results, we introduce a new metric which was applied only to test datasets for every interval and not during training. Similar to Eq.(5.21) in Section 5.6, instead of using SSE, we calculate $(\|\mathbf{x}_i(t) - \hat{\mathbf{x}}_{i,-i}(t)\|_2)^2$, where $\hat{\mathbf{x}}_{i,-i}(t)$ are the estimation of $\mathbf{x}_i(t)$ in the test set.

To provide an easier interpretation of the reliability and to define the notion and relationship with transmission gaps, we present detailed results from one and a half day. As shown in

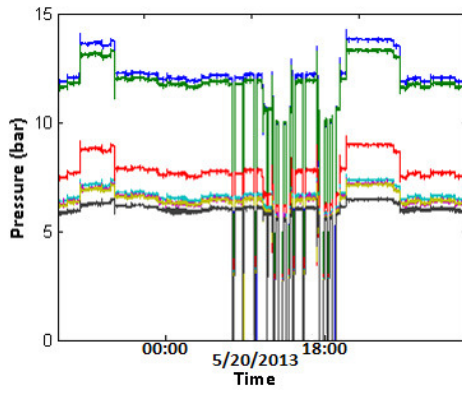
³The largest correlated set of our infrastructure consists of seven nodes, which is selected to represent our simulation results. In addition, four more sets of correlated nodes were considered, which consisted of 3, 3, 5, and 6 sensor nodes respectively.

Table 5.2: Evaluation results of the case study.

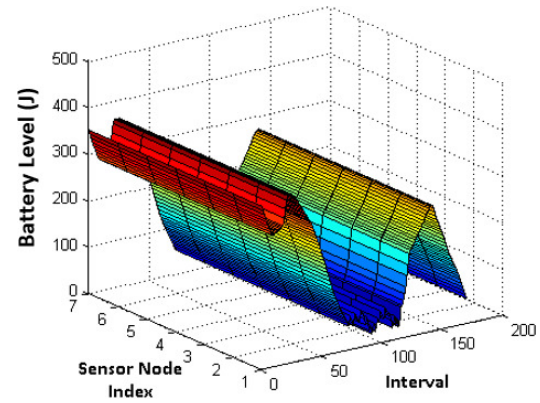
Algorithm	Reliability (%)	Energy Waste (kJ)	Transmission Gaps
RG	92.8	0.5	84
EG3	31.9	15.3	0
EG6	99.1	1.4	0
RR3	69.3	15.4	0
RR6	99.3	1.4	0
FDTS	99.8	0.5	0

Table 5.2, FAST-DTS achieves the highest estimation reliability over all algorithms. Furthermore, although the expected reliability of estimation models is above 95% for all nodes, the actual estimation results of some algorithms are less. This is caused by two reasons: (1) the non-optimal selection of a small set of nodes to transmit raw data, and (2) the transmission gaps. For example, EG3 and RR3 belong to the first case and produce the lowest estimation reliability results, 31.9% and 69.3% respectively. For the second case, the RG algorithm performs 7% less in terms of estimation reliability, in spite of the continuous transmission from all the nodes. The reason is that in some intervals the nodes are unable to transmit data, because of battery depletion and transmission gaps shown in last column of Table 5.2. These gaps equate data reliability to 0% for these intervals (since no data can be transmitted). Figure 5.8a and Figure 5.8c illustrate the transmitted data from RG and FAST-DTS respectively and emphasize the transmission gaps of RG (zero values in Figure 5.8a).

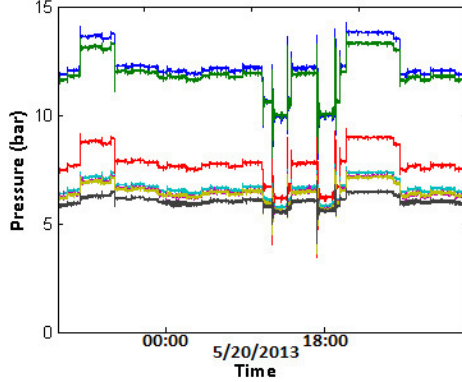
During transmission gaps, RG aggressively utilize energy and forced each node's battery level to zero (Figure 5.8b), leading to unsustainability of the system and frequent transmission gaps (Figure 5.8a). In contrast, FAST-DTS tries to keep the battery level above B_{exp} , as shown in Figure 5.8d respectively. This successfully avoids transmission gaps and therefore system sustainability. In addition, FAST-DTS also achieves the best performance in harvested energy utilization. In some cases, the energy harvesting system can produce more energy than the sensor nodes' need and the system wastes energy. For example, the EG3 and RR3 algorithms which select only three nodes per interval keeps the energy close to B_{max} and at every interval waste energy that could be used to schedule more nodes to transmit raw data. In summary, FAST-DTS performs the best battery management by minimizing energy waste and avoiding



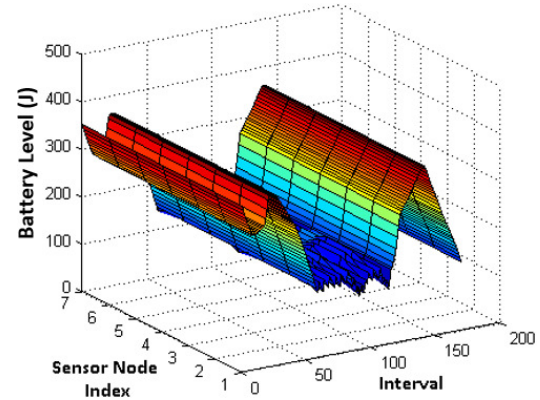
(a) Pressure data transmission using RG



(b) Battery level evolution using RG



(c) Pressure data transmission using FAST-DTS



(d) Battery level evolution using FAST-DTS

Figure 5.8: Reliability-Greedy and FAST-DTS comparison based on reliability and battery level per node

transmission gap, as shown in Table 5.2.

5.7.3 System Adaptation in Anomalies

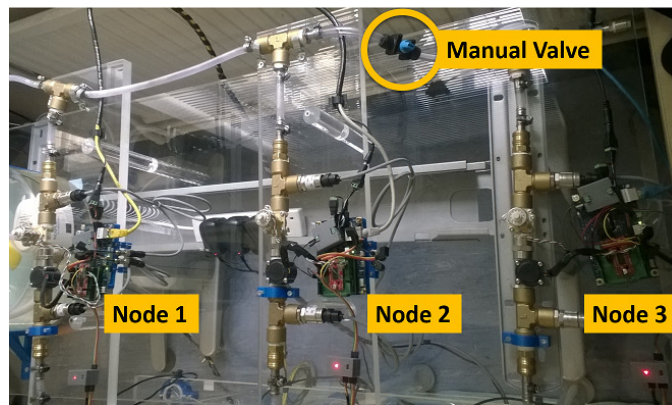


Figure 5.9: Testbed modification for anomaly emulation.

In this subsection, we present the impact of an observed anomaly on the behaviour of our system

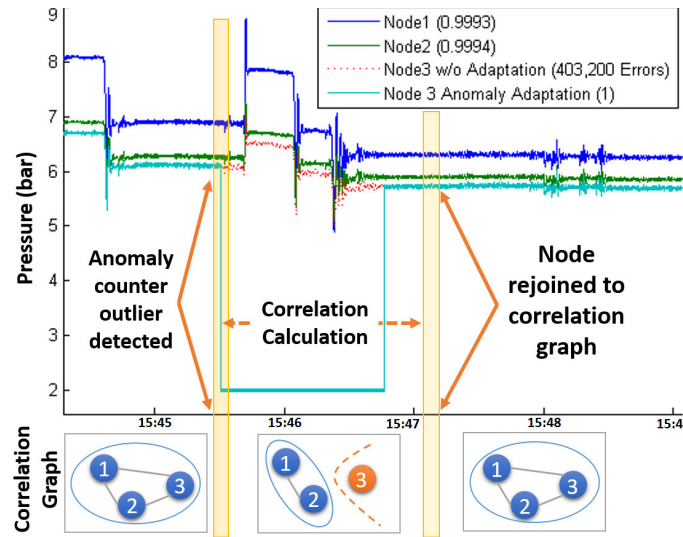


Figure 5.10: Correlation adaptation based on anomaly variance.

and the correlation graph update. As shown in Figure 5.9, we modify WaterBox to emulate the anomaly behaviors of three DMAs' sensor nodes (Nodes 1-3). Specifically, a manual valve was installed between the second and third DMAs, and bursts were emulated by opening/closing the valve abruptly.

Figure 5.10 presents an example of these three sensor nodes and the system's behavior in an anomaly which was occurred due to abrupt closing of the valve before Node 3. During the closing of the valve (Figure 5.10 first vertical line), the three sensor nodes transmit to the data processing center anomaly counters 3, 2, and 100 respectively due to sudden pressure drop in Node 3. Then, the data center detects the Node 3 anomaly counter as an outlier and forces Node 3 to transmit raw data. After receiving raw data, the data processing center calculates the correlation and decides to remove Node 3 from the correlation graph. Without this functionality, Node 3's data could continue to be estimated by other nodes' data and to produce inaccurate data (horizontal dashed line arrows - 403200 wrong estimation values to the data center). After some intervals (second vertical line), we open the valve and Node 3 returns to normal hardware operation. Then, the data processing center observes high correlation and rejoins Node 3 to the correlation graph. Due to the fact that a processing center is a highly capable computer system (i.e. server or base station), the graph update is performed instantly compared to the slow LPWA communication process.

5.7.4 The Impacts of System Parameter Settings

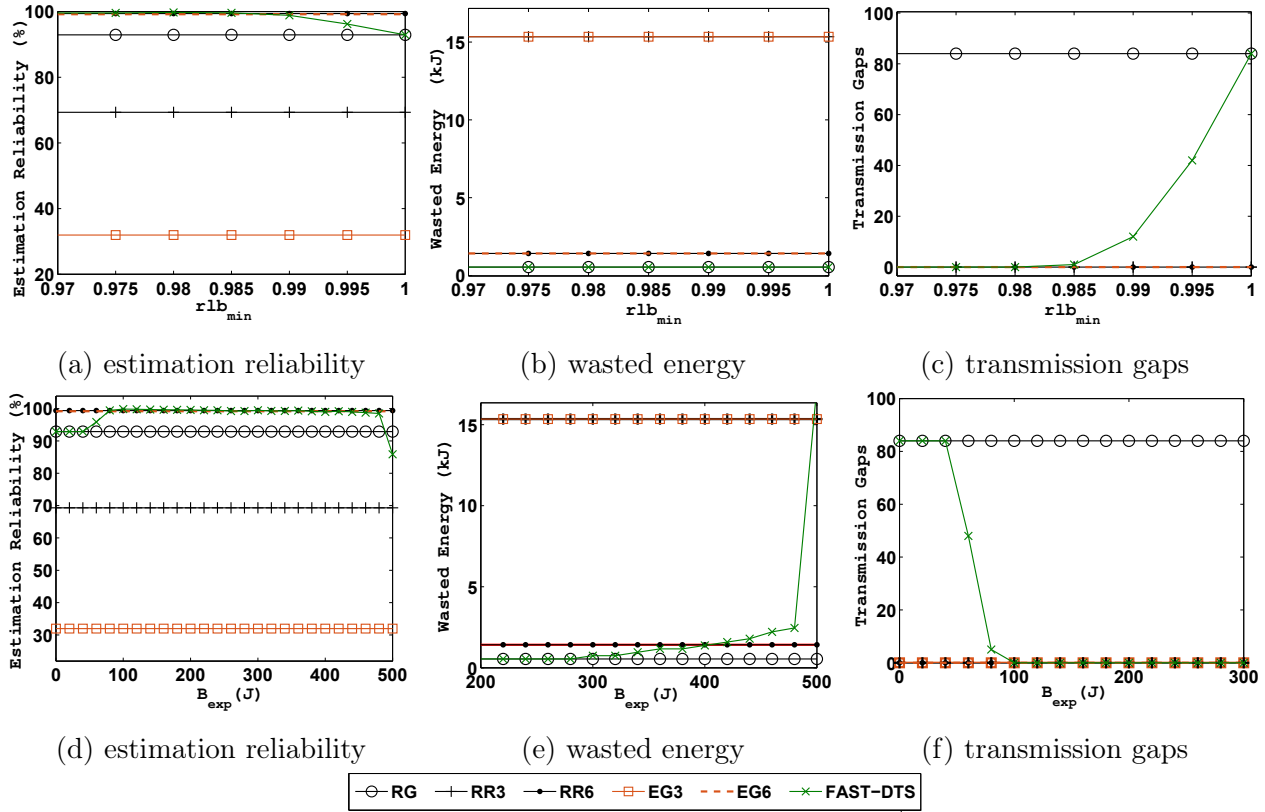


Figure 5.11: Reliability, energy waste, and transmission gaps with different rlb_{min} and B_{exp} .

The system parameters B_{exp} and rlb_{min} have significant impacts on our smart water system, as shown in Figure 5.11. As the rlb_{min} tends to 1, the FAST-DTS behaves like RG. This is because more nodes are forced to transmit raw data to achieve a 100% reliability, resulting in the increase of transmission gaps (Figure 5.11c) and consequently the decrease of reliability (Figure 5.11a). As the B_{exp} parameter tends to 0, the scheduling algorithm drains the battery of the nodes and enforce their battery levels close to 0. This leads to the increase of transmission gaps (Figure 5.11f) and consequently the decrease of reliability (Figure 5.11d). As the B_{exp} tends to B_{max} , the system wastes more energy (Figure 5.11e) and the reliability drops (Figure 5.11d), because less nodes are being selected to maintain the battery level close to the high B_{exp} . Based on Figure 5.11b, the best configuration for the evaluation system parameters is $100 \leq B_{exp} \leq 300$ and $rlb_{min} \leq 0.995$ where 0.995 is the highest value of estimation reliability.

Table 5.3: Average results of 80 sensor nodes for 30 days ($B_{\text{exp}} = 150$ and $\text{rlb}_{\text{min}} = 0.98$).

Algorithm	Reliability (%)	Wasted Energy (kJ)	Transmission Gaps
RG	84.9	50.3	1428
EG40	61.5	153.7	21
EG70	80.1	64.6	725
RR40	28.9	154.6	23
RR70	90.1	65.4	837
FDTS	95.9	52.8	335

5.7.5 Scalability Study

Real water networks can cover a large area with a corresponding large number of nodes, which requires the scalability of the algorithm design. In order to examine a larger scale smart water network, we exploit the raw data from the 24 sensor nodes and generate synthetic data of 56 virtual nodes, by virtually creating 2 neighbors for each real node (48 virtual nodes) and adding 8 additional virtual nodes in areas where the distance between two nodes was relatively large to the other nodes of the network. Then, the virtual nodes were distributed in equal distances between two real nodes. Hydraulic equations (i.e. Equation (5.1)), were used to generate the 30-day virtual pressure measurements. This process enable us to exploit the water network physics and avoids random data synthesis, resulting in realistic simulation studies.

The evaluation results of this scalability study are summarized in Table 5.3. RG algorithm waste less energy by 4.7% than FAST-DTS, however it occurs 4.3 times more transmission gaps. EG40 and RR40 produce 314 less transmission gaps, nevertheless these algorithms are inappropriate for deployment because of the poor data reliability results (61.5% and 28.9% respectively). Overall, FAST-DTS managed to maintain the battery level by keeping energy waste and transmission gaps low, and it achieved the highest data reliability.

5.8 Exploiting Distributed Compressive Sensing

In this chapter, we consider auto-regressive models to estimate the missing data. In our on-going research, we are examining the use of Distributed Compressive Sensing (DCS - see Section 2.3)

to achieve sustainability and high estimation reliability. Recalling Equation 2.7, the unknown vector of the i -th sensor node $\mathbf{x}_i$ is recovered from an underdetermined system of linear equation

$$\mathbf{y}_i = \Phi_i \mathbf{x}_i = \Phi_i \Psi_i^T \mathbf{w}_i,$$

where $\mathbf{y}_i \in \mathbb{R}^M$ is the vector of the i -th sensor node compressive measurements. Based on Equation (2.2) of Section 2.3, the selection of measurement matrix Φ_i determines the number of samples (i.e. sample rate SR_i) that the i -th sensor node transmits to a processing center (or base station). In the DCS applications, the selection of SR_i is not trivial and researcher (i.e. [77, 78]) consider the mean squared error (MSE) as appropriate metric to decide the lower bound of SR_i (i.e. Equation (2.11)).

However, to the best of our knowledge, none of the state-of-the-art solutions involves the sensor nodes' energy state of a network in the selection of Φ_i and consequently SR_i to guarantee sustainable operation. At this point, we propose the combination of our current Lyapunov Drift-Plus-Penalty approach and the MSE constraint in [78] to optimize both the selection of SR_i for all the nodes and guarantee sustainable operation, and minimize the network energy consumption. Formally, the problem can be formulated as an infinity-horizon stochastic optimization problem, by deciding the appropriate sample rate $SR_i(t), \forall i \in \mathcal{S}$ at each communication period t :

$$\underset{a \leq SR_i(t) \leq b, \forall i}{\text{minimize}} \quad \limsup_{t \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{i \in \mathcal{S}} U_i(EC_i(SR_i(t))) \quad (5.22)$$

$$\text{subject to} \quad \forall i \in \mathcal{S}, \limsup_{t \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T (EC_i(SR_i(t)) - H_i(t)) \leq 0 \quad (5.23)$$

$$\forall i \in \mathcal{S}, \sum_{\mathbf{J} \in \mathbf{\bar{S}}} Pr(\mathbf{J}) Tr((\mathbf{D}^T \mathbf{D})^{-1}) \leq \gamma \quad (5.24)$$

$$\forall i \in \mathcal{S}, SR_i(t) = Tr((D)^T(D)) \quad (5.25)$$

$$\forall i \in \mathcal{S}, B_i(t) = B_i(t-1) - EC_i(SR_i(t)) + H_i(t) \quad (5.26)$$

$$\forall i \in \mathcal{S}, B_{\max} \geq B_i(t) > 0 \quad (5.27)$$

where $EC_i(SR_i(t))$ is the function of energy consumption of each node for every communication

period t and it can be calculated as follows:

$$EC_i(SR_i(t)) = \frac{1}{C_e}(SR_i(t)(E_i^{tr}(t) + E_i^{se}(t)) + E_i^{id}(t)) - E_i^{sd} \quad (5.28)$$

where C_e is columbic efficiency⁴, E_i^{tr} and E_i^{se} represent the energy consumption due to transmission and sensing of one sample, E_i^{id} is the energy consumption during idle mode, and E_i^{sd} is the self discharge of the sensor node. Similar to our current problem, the utility function $U_i(EC_i(SR_i(t)))$ in the objective (5.22) can be any convex, non-decreasing, and differential function of $EC_i(SR_i(t))$.

Constraint (5.23) demonstrates that the long-term average of consumed energy for each sensor node should not be more than that of the harvested energy. Constraint (5.24) presents the minimum combined sampling rate of sensor nodes as was defined by DCS with Equation (2.11). Constraint (5.25) describes how SR_i is related to inequality of constraint (5.24). Constraint (5.26), represents the battery update based on a selected $SR_i(t)$. Finally, constraint (5.27) ensures that each node is not allowed to run out of battery energy or exceed its finite battery capacity.

In summary, we can see that problem (5.22)-(5.27) aims to minimize the long-term average energy consumption for each node $i \in \mathcal{S}$, while ensuring the sustainable operation for the long-term data acquisition. By exploiting Lyapunov Drift-Plus-Penalty method the above problem can be simplified and provide the necessary guarantees as follows:

$$\underset{a \leq SR_i(t) \leq b, \forall i}{\text{minimize}} \quad \sum_{i \in \mathcal{S}} ((B^{max} - B_i(t))(EC_i(SR_i(t)) - H_i(t))) \quad (5.29)$$

$$\text{subject to} \quad \forall i \in \mathcal{S}, \sum_{\mathbf{J} \in \mathbf{S}} Pr(\mathbf{J}) Tr((\mathbf{D}^T \mathbf{D})^{-1}) \leq \gamma \quad (5.30)$$

$$\forall i \in \mathcal{S}, SR_i(t) = Tr(\mathbf{D}^T \mathbf{D}) \quad (5.31)$$

$$\forall i \in \mathcal{S}, B_i(t) = B_i(t-1) - EC_i(SR_i(t)) + H_i(t) \quad (5.32)$$

$$\forall i \in \mathcal{S}, B_{max} \geq B_i(t) > 0 \quad (5.33)$$

⁴columbic efficiency (also called Faraday efficiency) describes the efficiency with which charge (electrons) is transferred in a system facilitating an electrochemical reaction.

where $V > 0$ is a system parameter that balances the trade-off between the system energy performance and the risk of battery depletion. The new formulation derives the benefits of Lyapunov Drift-Plus-Penalty approach and optimize the selection of SR_i , $\forall i \in \mathcal{S}$ in DCS.

5.9 Summary

In this chapter, we develop a reliable and sustainable wireless sensing system for high-sample-rate water pressure monitoring and abnormal behavior detection in smart water networks, powered by an energy harvesting system based on water flow. In order to achieve sustainable sensor node operation by utilizing the dynamic and limited harvested energy, we use lossless compression and the in-node anomaly detection algorithm of Section 4.1 to reduce the raw data volumes (and therefore the energy) required for the high-power and long-range wireless transmission.

To further reduce energy consumption, our system only requires the transmission of a subset of the raw data streams, while other data streams are estimated using auto-regressive models instead of complex hydraulic models. The estimation is based on strong correlations among sensor data streams caused by the sound-velocity propagation of pressure signals inside the water network.

We formalize a stochastic optimization problem for the best selection of raw data transmissions that aim to maximize the aggregated estimation reliabilities, while guaranteeing a minimal reliability constraint and the sustainable operation of the smart water sensing system. We develop DTS, a theoretically-proven asymptotically optimal solution to the formalized problem, based on Lyapunov optimization theory. Guided by the principles of DTS, we then propose FAST-DTS, a lightweight online algorithm that can adapt to arbitrary energy and correlation dynamics.

We evaluate our approach by using 170-day high sample rate data (128 samples per second) from the real smart water system of Section 2.1 and we compare our scheduling algorithm with three other algorithms. FAST-DTS outperforms these algorithms in terms of data reliability and

sustainable operation, by achieving 99% estimation accuracy and at the same time the lowest energy waste and transmission gaps. Currently, we are planning to replace auto-regressive models and to establish distributed compressive sensing as estimation method. Further, we are working to extend our approach by examining the impact of data estimation with regard to the automatic control process. In the following chapter, we introduce the automatic control notion to smart water networks.

Chapter 6

Event Trigger Control & Communication Framework

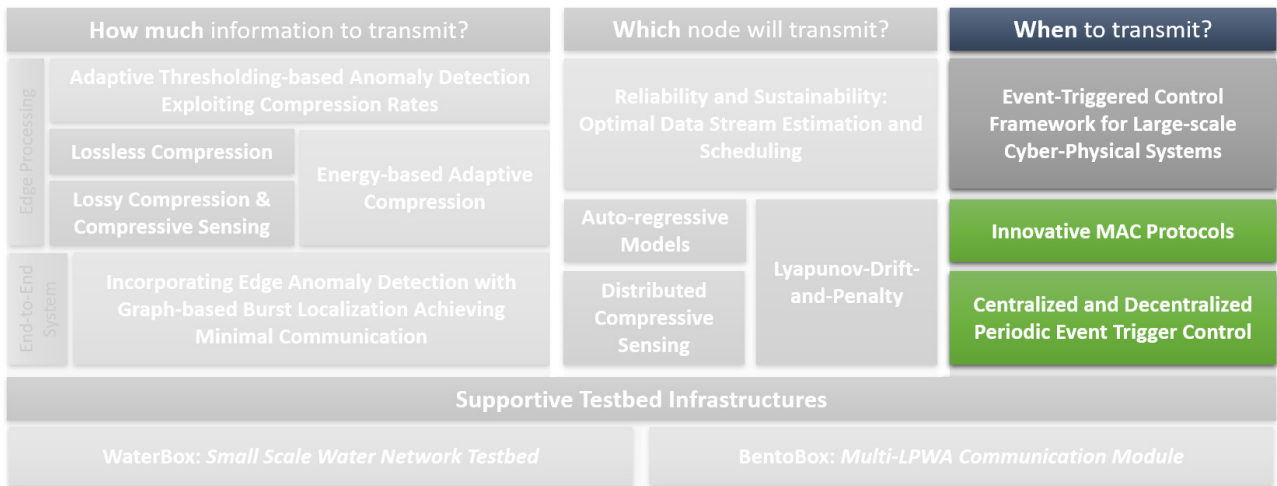


Figure 6.1: Chapter 6: Event Trigger Control & Communication Framework.

In large scale CPS, sensor and actuator nodes are usually energy constrained and installed in harsh environments. For example, in smart water network more than 97% of sensing and actuation assets are located underground and powered by batteries [1]. To transmit the required information through long-range (several kilometres) wireless communications, high transmission power is required that leads to fast battery depletion. In addition, the periodic sensing, transmission, and actuation, regardless the state of the plant, decrease network bandwidth and increase actuations and consequently the energy consumption. Recent control theory studies

propose to solve these problems by introducing aperiodic strategies, such as Event-Triggered Control (ETC) strategies, e.g. [89,94,95,171–173]. These strategies guarantee stability and performance while minimizing energy consumption and network bandwidth usage. Specifically, the sensors and actuators communicate only when the stability or a predefined control performance are about to diminish.

In spite of the potential of significant savings, ETC techniques have only been partially examined and implemented on real systems, i.e. [22,174–184]. In [176], the authors propose a system based on the Diddyborg robot and examine the strategy presented in [173]. However, this system is first-order and therefore unable to test complex event-triggered strategies. In [184], an experimental evaluation was made for time-triggered control and event-triggered control from [95]. However, this work requires state monitoring continuously to check event conditions. Additionally, the results in [95] can only be used for system with collocated sensors. To the best of our knowledge, there is no experiment that validates and compares different decentralized event-triggered mechanisms under the same conditions.

ETC systems have been studied extensively in order to guarantee convergence of plants under reduced communication schemes. However, the design and implementation of a communication protocol, which fully exploits the ETC behavior and ensures optimal communication, is still missing [185]. State of the art ETC approaches that are focused on communication, i.e. [186–194], have been limited to simulate or analyze theoretically the impact of network states on system performance. CSMA [195], TDMA [196] and ALOHA [197] were the three communication protocols which have been used in the above approaches. Specifically, [186,187] provide useful insights and comparison of all the above communication protocols. The authors in [188–190] present a Markov model that captures the joint interactions of the event-triggering policy and a contention resolution mechanism over CSMA communication. In [191,192], the ALOHA protocol, which has been applied in Long Term Evolution (LTE) Random Access (RA) procedure, was combined with ETC, with [192] to introduce the impact of collisions into the system performance. TDMA-based communication protocols, i.e. Time Triggered Network-on-a-Chip (TTNoC), Time Triggered Controller Area Network (TTCAN), were analysed in [193], which discusses their application to ETC systems. The earliest practical work is [184], contin-

ued in [194], by proposing the extension of the TDMA-based IEEE 802.15.4 MAC layer [198] which has been used in communication protocols for network control via wireless, i.e. WirelessHART [199]. However, the main drawback of this approach is the assumption that the actuator nodes listen continuously to network messages. Furthermore, none of the prior work has considered the cost of sensing. For example, in our evaluation platform [155], the sensing costs almost half the energy consumption of communication. In industrial systems with more energy hungry sensors, e.g. laser based turbidity sensors for water quality, the energy cost may surpass communications.

In this chapter, we not only propose an automatic and real time control system for large scale CPS, but also merge the event trigger control theory with real industrial implementations by introducing innovative communication schemes. Uniquely, in this chapter, the proposed system duty cycle the sensing and actuator listening activities and at the same time enable decentralized ETC techniques and introducing innovative communication schemes. Specifically, the contributions of this chapter are listed as follows:

- Practical combination of sensor duty-cycling with three decentralized aperiodic control approaches.
- A new MAC protocol that facilitates decentralized synchronous ETC without the requirement of continuous actuator communication.
- A novel flexible MAC protocol that can also accommodate two decentralized and asynchronous ETC approaches, communicating firstly absolute states and alternatively relative states only.

By using an extended version of the WaterBox testbed environment [155], we provide experimental results from Time-Triggered Control (TTC) and four different ETC techniques: Periodic centralized ETC (PETC) [171], Periodic Synchronous Decentralized ETC (PSDETC) [94], and Periodic Asynchronous Decentralized ETC (PADETC) by transmitting absolute or relative state [200]. To the best of our knowledge, this is the first real deployment of most of the implemented ETC techniques to a real plant. The work presented in this chapter is a result

of cooperation with Manuel Mazo Jr, Assistant Professor at TU Delft, and his PhD student Anqi Fu, who were responsible to provide the theory of decentralized event-triggered control, model WaterBox by applying system identification, and design the hybrid controller. Further, Anqi Fu was actively participated in the conduction of experiments and data analysis. My contribution was related to design the innovative MAC layers to accommodate the ETC techniques; understand the theory and solve all the existing unrealistic assumptions; design and optimize the system (both edge and server-side) by considering both ETC and communication limitations; develop the proposed solutions to WaterBox; and design and conduct the necessary experiments for both system identification and evaluation process.

6.1 Event-Trigger Control Techniques

We denote the positive real numbers by $\mathbb{R}^+$, the natural numbers including zero by $\mathbb{N}$. $|\cdot|$ denotes the Euclidean norm in the appropriate vector space, when applied to a matrix $|\cdot|$ denotes the l_2 induced matrix norm. A matrix $P \in \mathbb{R}^{n \times n}$ is said to be positive definite, denoted by $P \succ 0$, whenever $x^T P x > 0$ for all $x \neq 0$, $x \in \mathbb{R}^n$. For the sake of brevity, we write symmetric matrices of the form $\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$ as $\begin{bmatrix} A & B \\ \star & C \end{bmatrix}$.

6.1.1 Periodic control

Consider a linear time-invariant (LTI) plant and controller

$$\dot{\xi}(t) = A\xi(t) + Bv(t), \quad (6.1)$$

$$v(t) = K\xi(t), \quad (6.2)$$

where $\xi(t) \in \mathbb{R}^n$ is the state vector and $v(t) \in \mathbb{R}^m$ is the input vector at time t . Assume $A+BK$ is Hurwitz [201], the system is completely observable, and each sensor can access only one of the system states.

A sample-and-hold mechanism is implemented for the controller (6.2):

$$v(t) = K\hat{\xi}(t), \quad (6.3)$$

where

$$\hat{\xi}(t) := \xi(t_b), t \in [t_b, t_{b+1}[. \quad (6.4)$$

and $\{t_b\}_{b \in \mathbb{N}}$ is the sequence of the update time of the state. Representing the sample-and-hold effect as a measurement error, we have:

$$\varepsilon(t) := \hat{\xi}(t) - \xi(t). \quad (6.5)$$

Define T as the sample period. In a periodic time-triggered control strategy, t_b is determined by

$$\{t_b | t_b = bT, b \in \mathbb{N}, T > 0\}. \quad (6.6)$$

6.1.2 Periodic centralized event-triggered control

In event-triggered control strategies, the control input update time is determined by some pre-designed conditions. These conditions are always a relation between system state and sample-and-hold error (6.5). Therefore, control executions happen only when necessary. However, the centralized event-triggered condition presented in [95] requires the continuous monitoring and transmission of the current state to check the event conditions. If the state cannot be measured continuously, we can either compute a stricter event condition considering measurement delays; or apply the PETC strategy from [171], which combines periodic sampled-data control and event-triggered control:

Consider system (6.1), (6.3), (6.5), and a sample sequence (6.6). At each sampling time t_b , the

controller updates its state by

$$\hat{\xi}(t_b) = \begin{cases} \xi(t_b), & \text{when } \xi_p^T(t_b)Q\xi_p(t_b) > 0 \\ \hat{\xi}(t_{b-1}), & \text{when } \xi_p^T(t_b)Q\xi_p(t_b) \leq 0, \end{cases} \quad (6.7)$$

where $\xi_p(t) = \begin{bmatrix} \xi^T(t) & \hat{\xi}^T(t) \end{bmatrix}^T$, Q satisfies $Q := \begin{bmatrix} (1-\sigma)I & -I \\ -I & I \end{bmatrix}$, and $\sigma > 0$.

For the system (6.1), (6.3), (6.5), and (6.6), if $\exists c > 0$ and $\rho > 0$ such that for any initial condition $\xi(0) \in \mathbb{R}^n$, $\forall t \in \mathbb{R}^+$, $|\xi(t)| \leq ce^{-\rho t}|\xi(0)|$ is satisfied, then the system is said to be globally exponential stable, we call ρ the decay rate [202].

According to Corollary III.3 in [171], given a decay rate $\rho > 0$, if there exist a matrix $P \succ 0$ and scalars $\mu_i \geq 0$, $i \in \{1, 2\}$, such that

$$\begin{bmatrix} e^{-2\rho T}P + (-1)^i\mu_iQ & J_i^T e^{\bar{A}^T T}P \\ \star & P \end{bmatrix} \succ 0, i \in \{1, 2\}, \quad (6.8)$$

where

$$\bar{A} := \begin{bmatrix} A & BK \\ 0 & 0 \end{bmatrix}, J_1 := \begin{bmatrix} I & 0 \\ I & 0 \end{bmatrix}, J_2 := \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix},$$

then the system is globally exponential stable with a decay rate ρ .

6.1.3 Periodic synchronous decentralized event-triggered control

The event-triggered strategies presented in (6.7) are centralized event-triggered strategies, since the event conditions require the whole vector of current state and current error. When the sensors are not co-located, decentralized event conditions are preferred. We introduce the PSDETC strategy based on [94].

For system (6.1), (6.3), and (6.5), a decentralized event-triggering condition with periodic

sampling (6.6) is given by:

$$\hat{\xi}(t_b) = \begin{cases} \xi(t_b), & \text{when } \exists i : \varepsilon_i^2(t_b) - \sigma \xi_i^2(t_b) > \theta_i \\ \hat{\xi}(t_{b-1}), & \text{when } \forall i : \varepsilon_i^2(t_b) - \sigma \xi_i^2(t_b) \leq \theta_i, \end{cases} \quad (6.9)$$

where $\varepsilon_i(t)$ and $\xi_i(t)$ denote the i -th coordinates of $\varepsilon(t)$ and $\xi(t)$ respectively, and $\{\theta_i\}_{i \leq n}$ is a set of parameters. Define $\{t_k\} := \{t_b | \exists i : \varepsilon_i^2(t_b) - \sigma \xi_i^2(t_b) > \theta_i\}$ the sequence of event times. The sequence $\{\theta_i\}_{i \leq n}$ is obtained solving at each event time t_k :

$$\begin{cases} \hat{G}_i(t_k + t_e) = \hat{\varepsilon}_i^2(t_k + t_e) - \sigma \hat{\xi}_i^2(t_k + t_e) - \theta_i(k) \\ \hat{G}_i(t_k + t_e) = \hat{G}_j(t_k + t_e), \forall i, j \in \{1, 2, \dots, n\} \\ \sum_{i=1}^n \theta_i(k) = 0, \end{cases} \quad (6.10)$$

where for $t \in [t_k, t_{k+1}[$

$$\begin{aligned} \hat{\xi}_i(t) &= \xi_i(t_k) + \dot{\xi}_i(t_k)(t - t_k) + \frac{1}{2} \ddot{\xi}_i(t_k)(t - t_k)^2 + \dots \\ &\quad + \frac{1}{q!} \xi_i^{(q)}(t_k)(t - t_k)^q \\ \hat{\varepsilon}_i(t) &= 0 - \dot{\xi}_i(t_k)(t - t_k) - \frac{1}{2} \ddot{\xi}_i(t_k)(t - t_k)^2 - \dots \\ &\quad - \frac{1}{q!} \xi_i^{(q)}(t_k)(t - t_k)^q. \end{aligned}$$

The map $t_e : \mathbb{N} \rightarrow \mathbb{R}^+$ can be set to either $t_e(k) = T$ or $t_e(k) = t_k - t_{k-1}$. We apply Algorithm 1 in [94] to determine $t_e(k)$ in the experiments.

Thus, with the current $\theta_i(k)$ being calculated and transmitted from the controller to each sensor node, the sensor node can locally determine the occurrence of local events. When there is an event, the corresponding sensor node notifies the controller, and then the controller requests fresh measurements from all sensors to compute and update the control input.

Proposition 1. *For system (6.1), (6.3), (6.5), and (6.9), given a decay rate $\rho > 0$, if there*

exist a matrix $P \succ 0$ and scalars $\mu_1, \mu_2, \mu_3 \geq 0$, such that

$$\left\{ \begin{array}{l} \begin{bmatrix} e^{-2\rho T}P - \mu_1 Q & J_1^T e^{\bar{A}^T T} P \\ \star & P \end{bmatrix} \succ 0 \\ \begin{bmatrix} e^{-2\rho T}P + \mu_2 Q & J_2^T e^{\bar{A}^T T} P \\ \star & P \end{bmatrix} \succ 0 \\ \begin{bmatrix} e^{-2\rho T}P + \mu_3 Q & J_1^T e^{\bar{A}^T T} P \\ \star & P \end{bmatrix} \succ 0, \end{array} \right. \quad (6.11)$$

hold, then the system is globally exponential stable with a decay rate ρ .

Proof. According to [94], $\forall i : \varepsilon_i^2(t) - \sigma \xi_i^2(t) \leq \theta_i$ implies $\varepsilon^T(t)\varepsilon(t) \leq \sigma \xi^T(t)\xi(t)$, which is equivalent to $\xi_p^T(t_b)Q\xi_p(t_b) \leq 0$. However, $\exists i : \varepsilon_i^2(t) - \sigma \xi_i^2(t) > \theta_i$ may indicate $\xi_p^T(t_b)Q\xi_p(t_b) > 0$ or $\xi_p^T(t_b)Q\xi_p(t_b) \leq 0$. From the proof of Corollary III.3 in [171], if the hypothesis in Proposition 1 hold, by applying the S-procedure (see e.g. [203]), one obtains

$$x^T Q x > 0 \text{ holds, then } W(J_1 x, 0) \leq W(x, T),$$

$$x^T Q x \leq 0 \text{ holds, then } W(J_2 x, 0) \leq W(x, T),$$

$$x^T Q x \leq 0 \text{ holds, then } W(J_1 x, 0) \leq W(x, T).$$

in which $W(x, \tau)$ is a Lyapunov function defined as (18) in [171]. Therefore, W does not increase during samplings. Together with the results from Theorem III.2 in [171] that, such a P can guarantee $\frac{d}{dt}W \leq -2\rho W$ between samplings, the system is globally exponential stable with a decay rate ρ . $\square$

6.1.4 Periodic asynchronous decentralized event-triggered control

A periodic asynchronous event-triggered control strategy is presented in [200]. In this strategy, again the triggering condition is distributed to each sensor node. When there is an event, in contrast with PSDETC, only the corresponding sensor node data is used to update the controller. The updated control input is then calculated and transmitted to the actuators.

Consider system (6.1), (6.3), and (6.5), the current sampled state in the controller is updated as:

$$\hat{\xi}_i(t_b) = \begin{cases} q(\xi_i(t_b)), & \text{if } i \in \mathcal{J} \\ \hat{\xi}_i(t_{b-1}), & \text{if } i \in \mathcal{J}_c, \end{cases} \quad (6.12)$$

where $q(\xi_i(t))$ denotes the quantized signal of $\xi_i(t)$, $\mathcal{J}$ is an index set: $\mathcal{J} \subseteq \{1, \dots, n\}$ for $\xi(t)$, indicating the occurrence of events, $\mathcal{J}_c := \{1, \dots, n\} \setminus \mathcal{J}$.

Define $\Gamma_{\mathcal{J}} := \text{diag}(\gamma_{\mathcal{J}}^1 \dots, \gamma_{\mathcal{J}}^n)$. The element $\gamma_{\mathcal{J}}^l$, with $l \in \{1, \dots, n\}$ is equal to 1, if $l \in \mathcal{J}$, and 0 otherwise. Furthermore, we use the notation $\Gamma_j = \Gamma_{\{j\}}$. The local event-triggering condition is:

$$i \in \mathcal{J} \text{ iff } \xi_p^T(t_b) Q_i \xi_p(t_b) \geq \eta_i(t_b), \quad (6.13)$$

where $Q_i := \begin{bmatrix} \Gamma_i & -\Gamma_i \\ -\Gamma_i & \Gamma_i \end{bmatrix}$, $\eta_i(t) := \omega_i^2 \eta^2(t)$ is a local threshold, ω is a pre-designed distributed parameter satisfying $|\omega| = 1$, $\eta(t)$ is a global threshold, determined by

$$\eta(t_b^+) = \begin{cases} \mu \eta(t_b), & \text{if } |\hat{\xi}(t_b^+)| \leq \varrho \eta(t_b) \wedge \eta(t_b) > \mu^{-1} \eta_{\min}, \\ \eta_{\min}, & \text{if } |\hat{\xi}(t_b^+)| \leq \varrho \eta(t_b) \wedge \eta(t_b) \leq \mu^{-1} \eta_{\min}, \\ \mu^{-1} \eta(t_b), & \text{if } |\hat{\xi}(t_b^+)| \geq \mu^{-1} \varrho \eta(t_b), \\ \eta(t_b), & \text{otherwise,} \end{cases} \quad (6.14)$$

where $\eta_{\min} > 0$ is a pre-specified minimum threshold, and $\mu \in]0, 1[$ is a pre-designed parameter.

Consider the Hamiltonian matrix:

$$H := \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}, \quad (6.15)$$

where $H_{11} := \bar{A} + \rho I$, $H_{12} := 0$, $H_{21} := -(\gamma^2 I - I)^{-1}$, $H_{22} := -(\bar{A} + \rho I)^T$, for some $\gamma > 1$, and some given $\rho > 0$. And introduce the matrix exponential

$$F(\tau) := e^{-H\tau} = \begin{bmatrix} F_{11}(\tau) & F_{12}(\tau) \\ F_{21}(\tau) & F_{22}(\tau) \end{bmatrix}. \quad (6.16)$$

Define the matrix $\bar{S}$ satisfying $\bar{S}\bar{S}^T := -F_{11}^{-1}(T)F_{12}(T)$.

According to Theorem IV.4 in [200], consider the system (6.1), (6.3), (6.5), (6.12), and (6.14), given a decay rate $\rho > 0$. Assume $F_{11}(\tau)$ is invertible $\forall \tau \in [0, T]$. If there exist matrix $P \succ 0$, scalars $\varrho > 0$, $\beta_1 > 0$, $\beta_2 > 0$ and $\epsilon_{\mathcal{J}_i} > 0$, $\mathcal{J} \subseteq \{1, \dots, n\}$, $i \in \{1, \dots, n\}$ such that the bilinear matrix inequality

$$\begin{bmatrix} \beta_2 I & F_{11}^{-T}(T)P\bar{S} & \tilde{F} & -\beta_2 J_{\mathcal{J}} & 0 \\ \star & I - \bar{S}^T P \bar{S} & 0 & 0 & 0 \\ \star & \star & \tilde{F} & 0 & 0 \\ \star & \star & \star & P + \tilde{H}_1 & 0 \\ \star & \star & \star & \star & \tilde{H}_2 \end{bmatrix} \succ 0, \quad (6.17)$$

holds, where

$$\begin{aligned} \tilde{F} &:= F_{11}^{-T}(T)PF_{11}^{-1}(T) + F_{21}(T)F_{11}^{-1}(T) \\ \tilde{H}_1 &:= -\beta_1 I + \beta_2 J_{\mathcal{J}}^T J_{\mathcal{J}} - \sum_{i \in \mathcal{J}} \epsilon_{\mathcal{J}_i} Q_i + \sum_{i \in \mathcal{J}_c} \epsilon_{\mathcal{J}_i} Q_i \\ \tilde{H}_2 &:= \beta_1 \varrho^2 I - \beta_2 \bar{\Delta}_{\mathcal{J}}^T \bar{\Delta}_{\mathcal{J}} + \sum_{i \in \mathcal{J}} \epsilon_{\mathcal{J}_i} \Theta^T \Gamma_i \Theta \\ &\quad - \sum_{i \in \mathcal{J}_c} \epsilon_{\mathcal{J}_i} \Theta^T \Gamma_i \Theta \\ \bar{\Delta}_{\mathcal{J}} &:= \begin{bmatrix} 0 \\ \Gamma_{\mathcal{J}} \Theta \end{bmatrix}, \quad J_{\mathcal{J}} = \begin{bmatrix} I & 0 \\ \Gamma_{\mathcal{J}} & I - \Gamma_{\mathcal{J}} \end{bmatrix} \\ \Theta &= \begin{bmatrix} \omega_1 & \dots & \omega_n \end{bmatrix}^T, \end{aligned}$$

then $\mathcal{A} := \{x | |x| \leq \bar{\varrho} \eta_{\min}\}$ is a globally pre-asymptotically stable set for the system (see e.g. [204]), where $\bar{\varrho} := \max\{|J_{\mathcal{J}}|\varrho + |\bar{\Delta}_{\mathcal{J}}|, \forall \mathcal{J}\}$.

In [200], the update of the signals $\hat{\xi}_i(t_b)$ is given by:

$$\hat{\xi}_i(t_b) = \hat{\xi}_i(t_{b-1}) + \text{sign}(\hat{\xi}_i(t_{b-1}) - \xi_i(t_b)) m_i(t_b) \sqrt{\eta_i(t_b)},$$

in which $m_i(t_b) := \lfloor \frac{|\hat{\xi}_i(t_{b-1}) - \xi_i(t_b)|}{\sqrt{\eta_i(t_b)}} \rfloor$, $i \in \{1, \dots, n\}$. Thus, in practise one only needs to send $\text{sign}(\hat{\xi}_i(t_{b-1}) - \xi_i(t_b))$ and $m_i(t_b)$ from a sensor to the controller. We call PAETC with this update mechanism as PADETCrel, since we transmit the relative value (increment and its sign). A

dynamic quantizer can be applied with maximum quantization error $\sqrt{\eta_i(t_b)}$ for each sensor. However, since the static quantizers we install in our experimental setting have quantization error neglectable compared with the noise, and to compare with the rest of strategies fairly, we can instead update signals $\hat{\xi}_i(t_b)$ by $\hat{\xi}_i(t_b) = \xi_i(t_b)$. We call PAETC with this update mechanism as PADETCabs, since we transmit the absolute value.

6.2 Incorporating ETC with the MAC Layer

The MAC network layer provides addressing and channel access control mechanisms to enable several network nodes to communicate within a shared medium. These mechanisms are critical to control systems which require time guarantees and reliable packet transmission. Therefore, the incorporate of ETC in the MAC layer is vital. In this section, we present the design and implementation of three innovative TDMA-based MAC protocols which enable the deployment of TTC, PETC, PSDETC, and PADETC approaches accordingly: Control-TDMA (C-TDMA), Synchronous Decentralized-CTDMA (SDC-TDMA), and Asynchronous Decentralized-CTDMA (ADC-TDMA). The main benefits of these ETC-specific MAC protocols are: the optimization of communications by fully exploiting the behaviour and needs of ETC; the minimization of actuator node listening through duty cycling; and the off-load of the local controller node (base station) by minimizing the interaction with sensor and actuator nodes. For the proposed TDMA-based communication schemes, we assume a CPS network architecture, in which the sensor/actuator nodes communicate with a based station and retrieve acknowledgement messages per transmission. Further, the controller which computes the control signals is executed in the base station. In this chapter, the terms controller and base station are used interchangeably.

6.2.1 Simplistic TDMA Protocol

TDMA is a channel access method for shared medium networks, which allows several users to share the same frequency channel. Specifically, time is divided into intervals T_i with length T , so-called super-frames. Each interval is split into smaller time slots S_j , with $\sum_{j=1}^N S_j \leq T$,

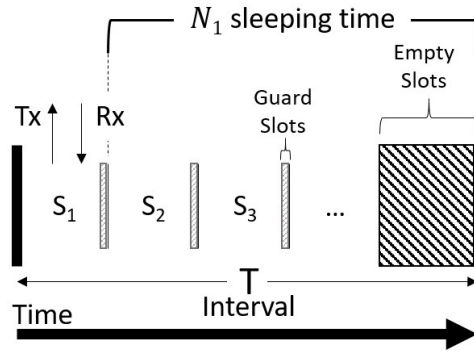


Figure 6.2: Simplistic TDMA MAC Protocol.

where N is the number of sensor/actuator nodes which share the same channel¹. In each time slot, only one predefined sensor/actuator node N_j can transmit (T_x) or receive (R_x) a burst of messages to and from a base station. Outside the timeslot S_j , N_j sleeps or executes other tasks depending on the hardware infrastructure and the provided ability to duty cycle. To avoid time violations of time slot bounds due to N_j possible clock drift, a guard slot forces the termination of communication before the end of each S_j . Figure 6.2 illustrates the communication scheme of a simplistic TDMA protocol.

Due to synchronous operation, a TDMA-based protocol (e.g. [199]) can guarantee tight bounds on delays which are critical for network control systems. On the other hand, synchronizing sensor/actuator nodes is considered as the main drawback of TDMA-based systems. However, state of the art solutions, i.e. GPS clock synchronization technologies [205], ensure typical accuracy better than 1 microsecond by consuming ultra low power.

6.2.2 Control-TDMA Protocol

Control-TDMA (C-TDMA) is designed to enable TTC and PETC approaches (see Figure 6.3). In order to ensure the simultaneous state sampling, in the beginning of every interval T_i at time t_i , every node N_j has to retrieve a state measurement x_j from the available sensors. Then, the channel bandwidth is divided into two sets of time slots which are separated by a time delay:

Measurement Slots S_j^x (X-slots): Every sensor/actuator node N_j transmits x_j within the

¹A super-frame can be divided into equal time slots that fully utilize the channel or to minimal slots which cover the application requirements and allow the communication to new nodes into the same channel.

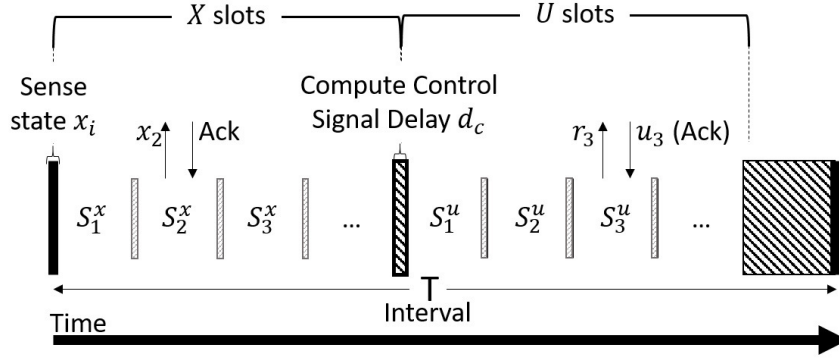


Figure 6.3: CTDMA protocol.

time slot S_j^x to the controller. Within each time slot only one successful message is required. Thus, the size selection of S_j^x is application specific and depends on the size of x_j (e.g. 2 Bytes per sensor) and the number of re-transmissions to achieve high reliability based on the selected hardware.

Delay d_c : After receiving the complete sampled state by receiving $x_j, \forall j \in N$, a time delay is required to allow the computation of appropriate control signals u_j for every sensor/actuator node N_j . The length of this delay depends on the controller infrastructure and the complexity of the control model.

Actuation Slots S_j^u (U-slots): The last set of time slots is related to the control message retrieval by the sensor/ actuator nodes N_j . In each time slot S_j^u , node N_j transmits a request r_j for a control signal u_j to the controller. Then, the u_j is piggybacked to the acknowledgement message. The benefit of the r_j request is two-fold: (a) off-loads the controller side and (b) reduces N_j listening time. Otherwise, the controller has to transmit or broadcast u_j continuously by blocking other tasks, while N_j has to be active in receive mode during the full length of the S_j^u slot until a successful control message retrieval. Further, this approach causes more energy savings for nodes with communication modules that consume the same amount of energy for transmission and listening, i.e. [206]. The length of S_j^u depends on the size of u_j signals.

Based on the above, the minimum interval size T_{min} can be defined by the length of X-slots, U-slots and delay d_c , and the number of the nodes. Further, the T_{min} can be considered as the maximum time delay of the system.

TTC and PETC are centralized approaches and are executed in the controller. In both cases, the system requires the transmission of the current state to the controller at every T_i during the X-slots. Then, in the TTC technique, the controller replies back in the U-slots of every interval T_i with a new u_j control signal. On the other hand, in PETC, the controller evaluates the event condition, as has been described in Equation (6.7), and transmits the new u_j only if there is a violation. This behaviour allows PETC to save energy due to actuation reduction.

6.2.3 Synchronous Decentralized Control-TDMA Protocol

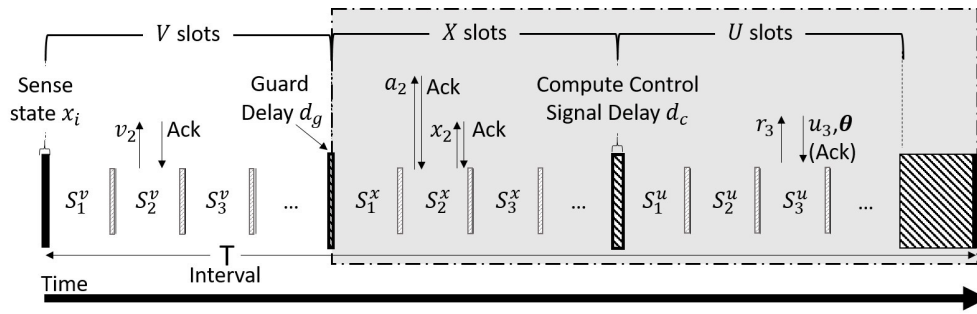


Figure 6.4: SDC-TDMA Protocol for PSDETC.

PSDETC is a distributed technique and each sensor node is responsible for the state transmission decision in every T_i . The computation of control signals u_j and θ_j parameters require the **complete** knowledge of the system's state for the interval T_i in which a threshold violation has happened. For example, consider a system with three nodes, $\{N_1, N_2, N_3\}$, in which only two of the nodes, i.e. $\{N_2, N_3\}$, observe threshold violations. The controller requires the state from all the three nodes to compute the control signals. Using the same example, in a TDMA-based communication scheme in which each node is assigned to a pre-defined time slot S_j , node N_1 is precluded from transmitting its state by being unable to be informed about the threshold violations of N_2 and N_3 . To overcome these limitations, SDC-TDMA introduces a new set of time slots S_j^v , the V-slots (see Figure 6.4).

Violations Slots S_j^v (V-slots): In the beginning of every T_i , each node retrieves a measurement and evaluates Equation (6.9). Then, the result of each threshold v_j is transmitted by the corresponding node N_j to the controller at time slot S_j^v .

Measurement Slots S_j^x (X-slots): In the beginning of every S_j^x in X-slots, each node N_j asks the controller, by sending an a_j request, whether a threshold violation was observed in the V-slots. If no threshold violation occurred, the sensor node sleeps immediately until the next interval T_{i+1} (gray box in Figure 6.4). Otherwise, each node transmits each state x_j to controller, wait for the delay d_c and actuation slots, U-slots, follow.

Delay d_c & Actuation Slots S_j^u (U-slots): Similar to the C-TDMA approach, after a delay d_c , each node requests the new control signal u_j from the controller. The u_j and the new threshold parameters θ_j , which is being calculated based on Equation (6.10), are being piggybacked into the acknowledgement messages of U-slots.

Based on the above, SDC-TDMA sacrifices channel availability and increases the minimum interval length, T_{min} and consequently the system's maximum delay, by adapting V-slots into the TDMA scheme. However, in the case of threshold violation absence, the communication is being minimized significantly by avoiding the transmission of state x_j and the entire execution of U-slots.

6.2.4 Asynchronous Decentralized Control-TDMA Protocol

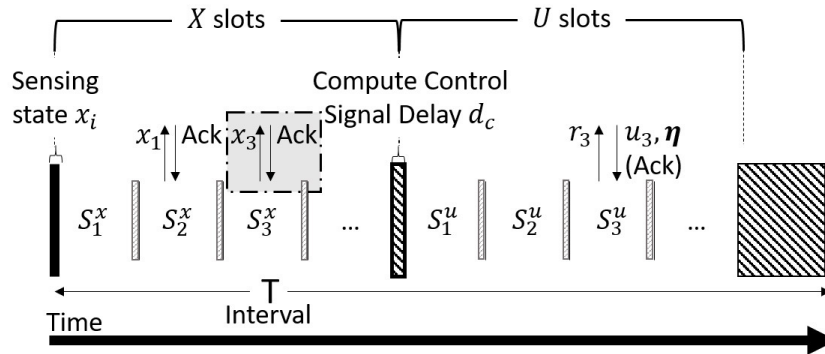


Figure 6.5: ADC-TDMA Protocol for PADETC.

Similar to PSDETC, the PADETC approach transfers the communication decision making from the controller down to the sensor/actuator nodes. Additionally, due to its asynchronous feature, this approach increases the communication savings by avoiding the state transmission x_j from every node N_j in every interval T_i . The only overhead in the communication is the η_j

update based on Equation (6.14) and transmission to N_j . The value of η_j is being piggybacked with the control signal u_j in the acknowledgement message.

Specifically, the architecture of ADC-TDMA is similar to C-TDMA and consists of sensing task, X-slots, d_c delay, and U-slot (see Figure 6.5). In a S_j^x slot, the node N_j evaluates the threshold of Equation (6.13). In the case of no threshold violation (i.e. gray box of Figure 6.5), the node N_j skips the communication and returns to sleep mode until S_j^u . Otherwise, N_j transmits to the controller: x_j in PADETCabs or the increment m_j in PADETCrel (see Section 6.1.4). Then, the controller computes the appropriate control signals and updates the local and global η based on Equation (6.14) by using **only** the available x_j states.

In the U-slots, every node has to send a r_j request message to the controller, in order to be notified about a possible threshold violation from another node. Therefore, the violation of at least one threshold causes the update of u_j and η_j to be sent to every actuator node. The values of u_j and η_j are piggybacked to the acknowledgement message.

6.3 WaterBox Infrastructure & Modelling

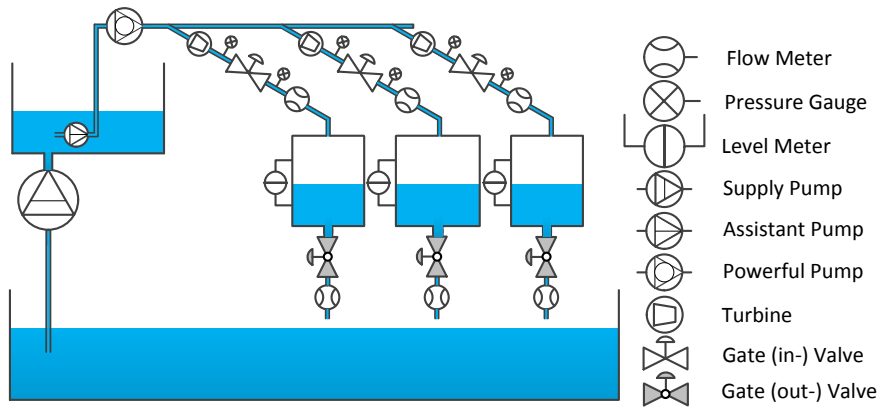


Figure 6.6: The schematic structure of the WaterBox.

Smart water networks have been used as a proof of concept for our proposed framework. The WaterBox platform (see Figure 3.4) is a scaled version of such water network [155] and developed to demonstrate real time monitoring and control by adapting innovative communication theories

and control methodologies. WaterBox was used as evaluation platform for our proposed ETC techniques and communication schemes.

6.3.1 WaterBox infrastructure

A Water supply network structure consists of three individual layers: (a) storage and pumping, (b) water supply zones and District Meter Areas (DMAs), and (c) end users (water demand). While valves control flows and pressures at fixed points in the water network, pumps pressurise water to overcome gravity and frictional losses along supply zones, which are divided into smaller fixed network topologies (in average 1500 customer connections) with permanent boundaries, DMAs. The DMAs are continuously monitored with the aim to enable proactive leakage management, simplistic pressure management, and efficient network maintenance. WaterBox was designed to support this architecture as follows:

Water Storage and Pumping

The structure of the WaterBox is shown as Figure 6.6. The WaterBox has a lower tank (i.e. ground, soil), an upper tank (i.e. reservoir, lake) and three small tanks (i.e. DMAs). The lower tank collects water from small tanks, and supplies water to the upper tank by an underwater bilge supply pump. This supply pump can supply enough water as the system requires. An assistant bilge pump and a new powerful pump are installed in series inside and after the upper tank respectively, and supply water to the small tanks. When the small tanks require more water supply, the assistant pump and powerful pump work together. When the small tanks require less water supply, only the assistant weak pump works. This behaviour emulates the day and night pumping operation of a water network in which the demand changes dramatically.

Water Supply & Sensor/Actuator Node

The water from the powerful pump flows into three small 'DMA' tanks via a pipe network. For the inlet each tank, a sensor/actuator node (see Figure 3.5), based on the Intel Edison

development board [207], controls the water flow through a motorized gate valve, so-called in-valve, and monitors the water flow, pressure (before and after in-valve) and the in-tank water level. Further, a turbine (flow-based energy harvester) is installed before each in-valve to harvest energy. To capture the total energy consumption of each sensor/actuator node, a custom made sensor module was created.

Remark. *In the WaterBox infrastructure, the sensors and actuator of each inlet are connected to the same node. However, our proposed communication schemes can be applied to non-collocated infrastructures.*

Water Demand Emulation

At the bottom of each small tank, there is an opening which enables the emulation of water consumption. A gate valve, so-called out-valve, is installed after each opening and can be controlled by a sensor/actuator node. The control of out-valves changes the outlet flow rate and facilitates the emulation of user's random water consumptions (i.e. external disturbance).

Base Station (Controller)

Every sensor node is connected to a local isolated WiFi network². A laptop is used as a controller or base station and hosts a visualization application (see Figure 3.9) which presents at real time the current state of the system, allows the manual control of actuators, logs the retrieved messages, and enables our proposed communication schemes and ETC scenarios per experiment. Additionally, due to lack of an indoor GPS time synchronization, a NTP server is running in the local controller and ensures less than millisecond time synchronization accuracy among the nodes.

²The isolation was achieved by disabling SSID operation (broadcast of WiFi availability to new users) from the router and selecting the low occupied communication channel for the nodes based on spectrometer experiments

6.3.2 System Identification & Modelling

We apply Grey-Box identification [208] to generate the system model and to find the uncertain parameters. A first principle model³ is obtained following [209]. We identify independently models under both mode 1: only the assistant pump works and mode 2: both pumps work. Using the Matlab cftool, we generate fitting curves for the gate valve coefficient of each in-valve, the turbine efficiency, and the pump efficiency. These curves are used to compute the first principles model. Since our aim is to stabilize the water level at the desired height h'_i , the model is linearised around this height. In this process, in order to simplify the simulation of the user water consumption, we keep the out-valves open, thus, constant out flow rates can be assumed.

6.4 Switched Controller Design

To evaluate the proposed ETC-based communication schemes, the following **control scenario** was used: *"Control in-valves to stabilize the water level of the small 'DMA' tanks to a certain level ensuring pressure and flow bounds. Enable mode 2 (weak and powerful pump) only if the system is away from the target levels. Switch to mode 1 (weak pump) only when the system is close to the reference state to guarantee efficient low pressure and flow operation."*

In the design of the controller, the following limitations need to be considered:

1. *Saturation of the actuators:* The maximum open level of one in-valve is 360° , while the minimum is 0° .
2. *Actuator quantization:* Due to the limitation of the valve's control components, their open levels can only be changed in steps of 10° . Therefore, small disturbances may result in dramatic changes of actuations.

³For further details, Appendix C provides the first-principle dynamical model of WaterBox as has been designed by Anqi Fu

3. *Over-pressure protection:* Due to mechanical limitations of the pipe network, there is a maximum allowable pressure for the pipe network.

Since the height of the water levels have a direct effect on the Quality of Service (QoS), the closed-loop system requires a fast response; however, since the size of the small tanks are limited, the overshoot should simultaneously be constrained. Experiments show that, in mode 2, the pipe network may be over pressured, when the open level of the in-valves, defined by α_j^{in} , cannot satisfy:

$$\sum_{j=1}^3 \alpha_j^{in} \geq 180^\circ. \quad (6.18)$$

Overshoot and disturbances could make condition (6.18) be violated. While in mode 1, there is no such constrains, that is, even all three in-valves are totally closed, the pipe network will not be over pressured. Therefore, filling the small tanks in mode 2 and switching to mode 1 when (6.18) is violated is required. Also experimentally, we observe that, when the system is in mode 1, the pump may not provide enough water supply to the small tanks even at the maximum open level, i.e. $\alpha_j^{in} = 360^\circ$, $\forall j \in \{1, 2, 3\}$. Therefore, switching back to mode 2 when the water in the tanks reaches some pre-defined low levels is necessary. To support this mode switching, we define $\underline{h} := \begin{bmatrix} \underline{h}_1 & \underline{h}_2 & \underline{h}_3 \end{bmatrix}^T$, $\underline{h}_j < h'_j$, $j \in \{1, 2, 3\}$, as the lower water levels. If $\exists j \in \{1, 2, 3\}$ such that $h_j(t) \leq \underline{h}_j$, the system switches from mode 1 to mode 2. With carefully chosen $\underline{h}_j$ and properly designed controllers, this violation can only happen in mode 1. Further analysis shows that, (6.18) can only be violated when $h_j(t) > h'_j$, which together with the fact that $\underline{h}_j < h'_j$ precludes Zeno behavior. Overall, Zeno behavior is avoided due to the use of periodic ETC and the design of system parameters.

Let $\vartheta \in \{1, 2\}$ represents the corresponding system mode. The linearized version of the first principle switched model (see Appendix C) and switched controller of WaterBox are described by:

$$\dot{\xi}(t) = A\xi(t) + B_\vartheta v(t), \vartheta = 1, 2, \quad (6.19)$$

$$v(t) = S(-K_\vartheta \xi(t) + \bar{\alpha}_\vartheta^{in}), \vartheta = 1, 2, \quad (6.20)$$

where $\xi(t) = \begin{bmatrix} \xi_1(t) & \xi_2(t) & \xi_3(t) \end{bmatrix}^T$, $\xi_j(t) := h_j(t) - h'_j$, $j = 1, 2, 3$ are the system states, $h_j(t) \in \mathbb{R}$ are the water levels, and $h'_j \in \mathbb{R}$ are the reference water levels; $v(t) = \begin{bmatrix} v_1(t) & v_2(t) & v_3(t) \end{bmatrix}^T$, $v_j(t) := \alpha_j^{in}$, $j = 1, 2, 3$ are the system control inputs, and $\bar{\alpha}_\vartheta^{in}$ are the equilibrium open levels of the in-valves per operation mode ϑ ; S is a map $\mathbb{R}^m \rightarrow \mathbb{R}^m$ representing actuator saturation and quantization, that is

$$S(s_j) := \max\{\min\{10\lfloor 10^{-1}s_j \rfloor, 360^\circ\}, 0^\circ\}$$

Then, the WaterBox hybrid controller state automaton is illustrated in Figure 6.7.

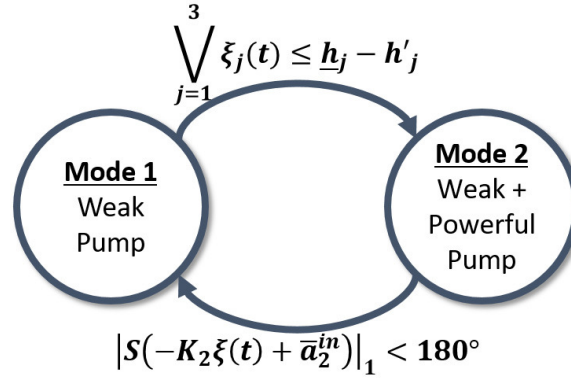


Figure 6.7: Hybrid controller state automaton.

From Grey-Box identification procedure, the system parameters are: A is a zero 3×3 matrix and

$$B_1 = 10^{-5} \times \begin{bmatrix} 0.1436 & -0.0170 & -0.0164 \\ -0.0098 & 0.1060 & -0.0100 \\ -0.0139 & -0.0139 & 0.1492 \end{bmatrix}, \quad B_2 = 10^{-5} \times \begin{bmatrix} 0.7666 & -0.0493 & -0.0457 \\ -0.0274 & 0.5848 & -0.0279 \\ -0.0393 & -0.0432 & 0.7865 \end{bmatrix}.$$

The controllers designed are given by:

$$K_1 = \begin{bmatrix} 99950 & 3029 & 872 \\ -3014 & 99940 & -1679 \\ -922 & 1652 & 99982 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 9998.5 & 167.1 & 41.0 \\ -166.6 & 9997.9 & -116.0 \\ -43.0 & 115.3 & 9999.2 \end{bmatrix}.$$

The designed controller is stable in both mode 1 and 2 because $-B_1 K_1$ and $-B_2 K_2$ are Hurwitz

matrices. Further, due to the long dwell time⁴ of the system, the closed loop retains stability. Given $h'_j = 0.06$ and $\underline{h}_j = 0.03$, $\forall j \in \{1, 2, 3\}$, $\bar{\alpha}_1^{in}$ and $\bar{\alpha}_2^{in}$ are computed:

$$\bar{\alpha}_1^{in} = \begin{bmatrix} 503.5950 \\ 422.4378 \\ 428.5839 \end{bmatrix}, \quad \bar{\alpha}_2^{in} = \begin{bmatrix} 84.5099 \\ 68.2069 \\ 72.8442 \end{bmatrix}.$$

6.5 Evaluation

This section summarizes the experimental results of more than 400 experiments conducted in WaterBox to evaluate our proposed communication schemes for the different ETC strategies. Each experiment executes the same control scenario (as described in Section 6.4) and the total process lasts between 7 and 10 minutes, including the water state initialization, the execution of experiments, and data logging from sensor/actuator nodes and local controller.

6.5.1 Evaluation Setup

Table 6.1: Communication parameter evaluation setup.

	Parameter	Value	Description
Packet Size	x_j	36 Bytes	Node ID Timestamp (in msec) Inlet pressure Outlet pressure Flow rate Total water volume Distance from surface Energy consumption Energy harvesting
	$Ack \mid r_j, a_j$	1 Byte	0 or 1 Node ID
	u_j, η_j, θ_j	2 Bytes	Control signal and DETC parameters
	m_j	4 Bytes	State delta
Time Duration	S_j^x	80 msec	X slot size
	S_j^u, S_j^v	50 msec	U and V slot size
	d_c	10 msec	Control decision delay
	d_g	5 msec	Threshold violation decision delay
	Guard slot	1 msec	Forced task termination time

⁴Dwell time is the time between two consecutive mode switches [210].

The proposed communication protocols of Section 6.2 were deployed to the WaterBox sensor/actuator nodes by wrapping the functionality of the Intel Edison WiFi module. Table 6.1 presents the configuration of the communication parameters. Based on the predefined packet sizes of the specific hardware infrastructure, a set of experiments was conducted to determine reliable time slot and guard delay sizes.

Table 6.2: Parameters of triggering strategies.

Method	T (sec)	Parameter	Value
TTC	0.5, 1, 2	-	-
PETC	0.5, 1, 2	σ	0.05, 0.1, 0.2
PSDETC	1, 2		
PADETC (abs & rel)	0.5, 1, 2	μ	0.75, 0.95
		ϱ	85 , 120

Based on the Table 6.1 timing parameters and the Section 6.2 time slot analysis, the minimum interval size T_{min} for C-TDMA and ADC-TDMA has to be more than 321 msec while for SDC-TDMA more than 564 msec (because of the V-slots). Thus, we evaluate TTC, PETC, and PADETC (with absolute or reference value) with interval size $T = 0.5, 1, 2$ and $T = 1, 2$ sec for PSDETC. The selected interval sizes and the rest parameters of the ETC strategies are listed in Table 6.2. The σ and ρ ETC parameters are chosen by finding feasible solutions of the corresponding Algorithms (6.8) and (6.17), while μ is tuned experimentally.

In the first set of experiments, we examine the impact of the ETC parameters (σ , μ , and ϱ) to the performance of the system. A fixed interval size $T = 1$ was used with all the different combinations of ETC parameter values of Table 6.2. Another set of experiments was conducted to explore the effect of T in the behaviour of the system. Keeping $\sigma = 0.2$, $\mu = 0.95$, and $\varrho = 85$ constant, all the experiments were re-executed with $T = 0.5$ and $T = 2$ (Table 6.2 bold text). To ensure the validity of the evaluation results, each experiment was repeated 10 times⁵ for each different combination of ETC parameters and T . Mean values are used to illustrate the evaluation results. The data was captured from the nodes and controller for the period of time between the beginning of each experiment ($t_0 = 0$) until a fixed end time ($t_{end} = 110s$), which guarantees the system turns to mode 1 and converges to steady state, denoted T_{exp} .

⁵The number of experiment repetitions was selected experimentally by analysing the variance of the results (i.e. under 2% of mean).

6.5.2 Experimental Results

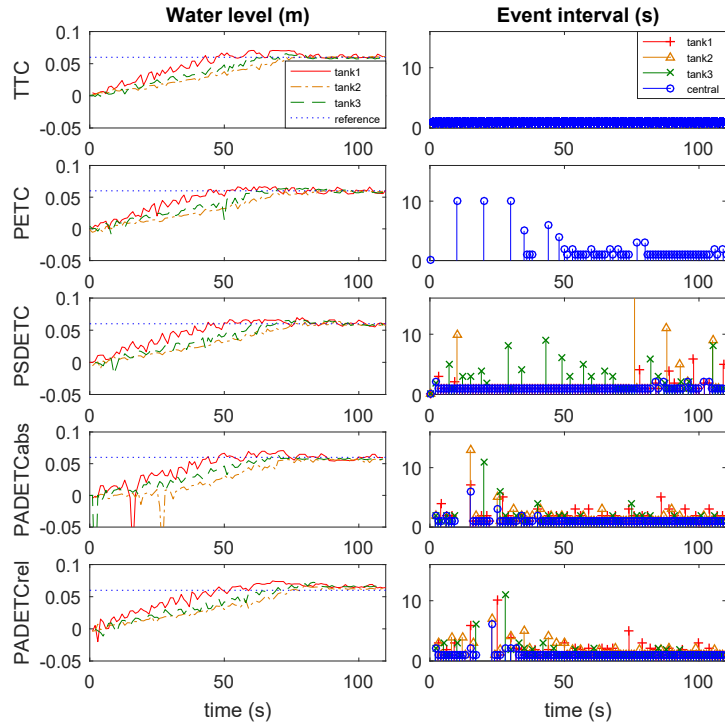


Figure 6.8: Experimental results. Note that, for SETC, the maximum event-interval of small tank 2 is 66 sec.

In this section we compare TTC, PETC, PSDET, PADETCabs and PADETCrel experimental results, in terms of:

- *Water Level Overshoot*: the maximum water level during the experiment. This parameter indicates the system's maximum state overshoot which is critical for water network asset safe operation.
- *Switching Time (t_{sm})*: the duration between experiment start time t_0 and first switch mode time t_{sm} , as described in condition (6.18). The time to mode switching is employed as an estimate of the system settling time (due to its ease of detection in our setup).
- *Sleep Time*: the total sleep time of all the nodes. This parameter evaluates the use of the bandwidth and CPU in the sensor/actuator node.
- *Discharge (Energy Consumption)*: our custom made sensor module retrieves current measurements c_n in mA at a fixed frequency of 10kHz. The energy consumption of a specific

time period Δt in seconds and with average current measurements over this period $\langle C \rangle_{\Delta t}$ can be derived from $E(\Delta t) = \langle C \rangle_{\Delta t} \cdot \frac{\Delta t}{3600}$. We used a hardware average to ensure the continuity of the current measurements and validated our instrument against a calibrated reference [211]. The energy consumption includes the consumption because of the communication, sensing, actuation, and idle mode. We present two discharge values, i.e. the whole discharge and discharge without sleep time. Based on these parameters, the battery lifetime of different hardware infrastructures can be implied.

- *Actuations*: the number of valves' changes, i.e. $\sum_{\forall T_i \in [t_0, t_{end}]} \sum_{j=1}^3 \alpha_j^{in}(T_i)$. The amount of actuations indicates the lifetime of actuators which is vital for industrial deployments.
- *Valve Movement*: the sum of valves' movement in degrees between two consecutive changes, i.e. $\sum_{\forall T_i \in [t_0, t_{end}]} \sum_{j=1}^3 |\alpha_j^{in}(T_i) - \alpha_j^{in}(T_{i-1})|$. Combined with the number of actuations, the valve movement can be used to estimate physical system lifetime.
- *Violations*: the sum of event condition violations. For each violation the local controller transmits a control signal u_j to each node i.e. three times. Therefore, the total transmitted control signals are equal to 3 times the violations. This metric indicates the communication requirements of actuators. Violations and actuations are different values because the local controller can produce the same consecutive control signal.
- *State Transmissions*: the sum of state x_j transmissions to local controller. This metric indicates the communication requirements of sensors. Both violations and state transmissions represent the total communication requirements of the system.

Figure 6.8 shows an example of raw data as captured from the nodes and the controller. Next, we analyse the energy consumption trends compared to sleep time for different hardware infrastructures, the effect of ETC parameter setup and interval length T to the performance of the system, and we aggregate the savings of ETC approaches against the vanilla scenario of TTC.

Energy Consumption and Sleep Time

The hardware infrastructure of a WaterBox node consumes more energy in sleep mode than operation mode. During sleep mode, our process yields priority to the background tasks of the operating system which are more energy hungry. In order to generalize the results to different hardware infrastructures which support lower energy consumption during sleep mode (i.e. deep sleep), we provide the upper and a lower bounds of energy consumption. The upper bound presents the real experimental results based on our node while the lower bound represents an estimation of energy consumption of a node which supports deep sleep⁶. The need of energy consumption range can be clearly seen in Figures 6.10c and 6.10d. In spite of the sleep time increase in all cases, the upper bound of energy consumption increases proportionally (the opposite holds for the lower bound). Additionally, PSDETC is expected to consume more energy than the others because of the V-slots. However, Figures 6.9c and 6.9d illustrate the opposite trend for the upper bound (opposite for lower bound) due to the energy hungry sleep mode. Overall, PADETCabs and PADETCrel consume the least energy compared to the other approaches. In spite of the uses of the same communication scheme, PETC performs slightly better than TTC of actuation reduction (quantitative results will be presented later on).

Effect of ETC Parameters

Figure 6.9 presents the effect of different parameters, e.g. σ , ρ , μ with the same interval length $T = 1$. The experiment results follow the trends shown in the theory. In PETC and PSDETC, a smaller σ forces the system to be more conservative and leads to more event condition violations (Figure 6.9g) and consequently to more actuation (Figure 6.9e) and energy consumption (Figure 6.9d). For the same reason, in the decentralized PSDETC, the state transmission reduces with bigger σ (Figure 6.9h).

In PADETC, a bigger ϱ has similar effect as a smaller σ in PETC. This can be clearly seen in Figure 6.9e and 6.9g, in which bigger ϱ causes more actuations and violations respectively.

⁶we calculated the lower energy consumption by subtracting the energy consumption during sleep mode from the total

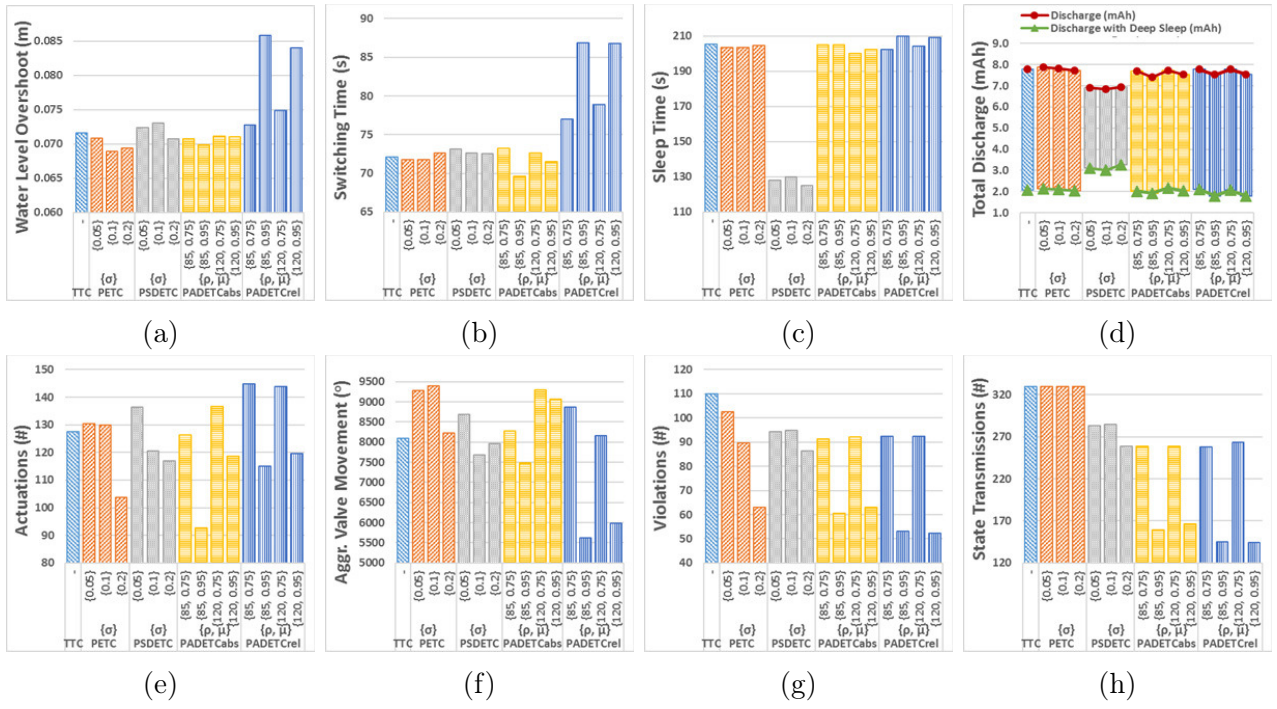


Figure 6.9: Impact of ETC parameters (σ, μ, ρ) in: (a) water level overshoot, (b) switching time, (c) sleep time, (d) discharge, (e) actuations, (f) valve movement, (g) violation, and (h) state transmissions.

A bigger μ can result in more frequent threshold updates, but maintains the threshold less conservative, and thus, the sampling errors can be enlarged. Additionally, Figure 6.9g shows that μ has greater impact on violations than σ and ρ parameters.

Impact of Interval Length Selection

Figure 6.10 illustrates the impact of different interval lengths, in which the same pre-designed Lyapunov converge rate can be guaranteed, for the same set of rest of the parameters, e.g. σ, ρ, μ . It can be clearly seen in Figure 6.10 that smaller interval T results in better performance but worse energy consumptions. The water level overshoots are almost the same because of the actuator quantization. Larger sampling times always result in longer convergence time and longer sleeping times. Similarly, the upper bound discharge indicates this trend; longer sleep time leads to higher energy consumption due to the energy hungry operating system background tasks. Oppositely, the lower bound of discharge shows that hardware infrastructures with deep sleep consume significant lower energy for larger interval T due to long sleeping time.

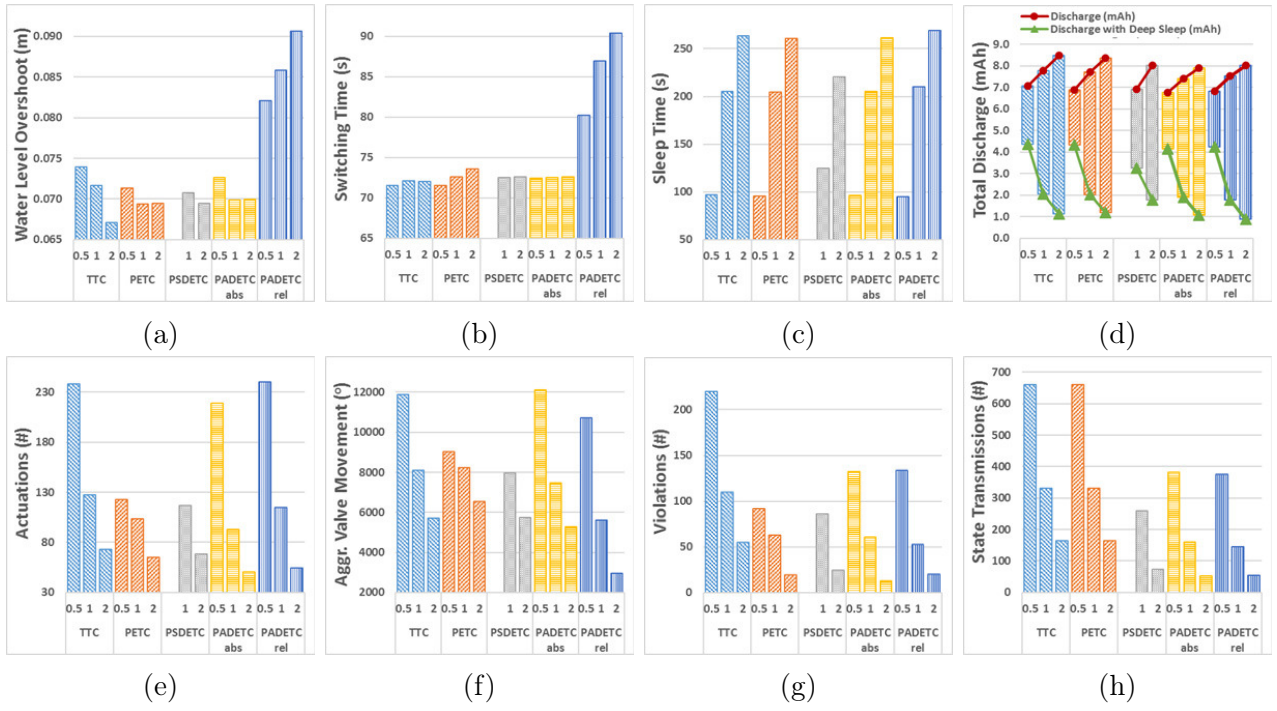


Figure 6.10: Impact of interval length T in: (a) water level overshoot, (b) switching time, (c) sleep time, (d) discharge, (e) actuations, (f) valve movement, (g) violation, and (h) state transmissions.

Savings Compared to TTC

Table 6.3: Savings compared to TTC (%) total experiment time, i.e. $[0, t_{end}]$, with $\sigma = 0.2$, $\rho = 85$, and $\mu = 0.95$

Approach	Water Level Overshoot	Switching Time	Discharge	Discharge with Deep Sleep	Actuations	Aggr. Valve Movement	Violations	State Transmissions
PETC	3.18	-0.69	0.97	1.1	18.6	-1.7	42.8	0
PSDETC	1.24	-0.55	11.03	-57.6	8.2	1.5	21.5	21.5
PADETC (abs)	2.44	3.47	4.74	6.9	27.2	7.5	44.8	51.6
PADETC (rel)	-19.72	-20.53	3.28	13.2	9.7	30.5	51.8	56.0

Table 6.3 and 6.4 show the total savings of different ETC techniques against TTC for the time period $period_1 = [0, t_{end}]$ (total experiment time) and $period_2 = [0, t_{sm}]$ (time until switching mode) respectively. We provide this data separately due to the existence of the switched controller and the different behaviour of the two modes.

In $period_1$ PETC performs similarly to TTC with the difference of reduced actuations and violations by 18.6% and 42.8% respectively. In spite the saving, PETC causes more valve movements than TTC. The PSDETC is more conservative than the centralized PETC, with a result, the lower savings in terms of violations. However, PSDETC reduces the valve movements and the state transmissions due to the decentralized architecture. PADETCabs outperforms all the

Table 6.4: Savings compared to TTC (%) until mode switching time, i.e. $[0, t_{sm}]$, with $\sigma = 0.2$, $\rho = 85$, and $\mu = 0.95$

Approach	Water Level Overshoot	Switching Time	Discharge	Discharge with Deep Sleep	Actuations	Aggr. Valve Movement	Violations	State Transmissions
PETC	3.15	-0.69	1.34	2.4	30.3	17.3	55.2	-0.7
PSDETC	1.67	-0.55	10.97	-66.9	10.4	8.3	14.6	14.6
PADETC (abs)	3.02	3.47	9.22	11.9	34.7	24.1	57.0	63.9
PADETC (rel)	-19.72	-20.53	-15.23	-4.5	14.4	22.6	49.1	54.5

other approaches because of the asynchronous behaviour, reducing significantly the violations, state transmissions and actuations by achieving 44.8%, 51.6%, and 27.2% savings respectively. PADETCrel reveals similar actuation and communication saving with PADETCabs but with the trade-off of lower performance in terms of water level overshoots and switching time. As has been described in Section 6.1, this happens because the PADETC with reference value updates introduces an extra error, known as maximum dynamic quantization error. However, this extra error allows this triggering mechanism to be more robust against noise than any of the other mechanisms with pre-designed maximum dynamic error.

In $period_2$, some ETC approaches deviate compared to the total savings. For example, in PETC approach, $period_2$ reveals higher actuation savings than $period_1$. The reason is that in mode 1, the weak pump is unable to supply the tanks with enough water and the system deviates from steady state continuously. Thus, event condition violations are being increased and often large valve movements are required. PSDETC and PADETCabs have a more stable behaviour than the other ETC approaches. Again PADETCabs outperforms the other ETC approaches achieving outstanding violation (57%) and actuation (35%) savings.

6.6 Summary

In this chapter, we have proposed duty-cycling of the sensing and actuator listening activities and enabled decentralized ETC techniques introducing innovative communication schemes. Specifically, we designed and implemented three new MAC layers, which enable the application of four different periodic centralized and decentralized event triggered control approaches. By implementing our proposed communication schemes in the WaterBox testbed [155], we provided experimental results from more than 300 experiments.

Based on the experimental results, ETC approaches can introduce considerable benefits into industrial deployments. Due to the outstanding decrease of actuations either in number (up to 35%) or size (i.e. for valve movement up to 24%), the ETC techniques can increase the robustness, resilience, and lifetime of physical plants and actuators significantly. This increase can lead to significant maintenance cost reduction by postponing expensive replacements of plant assets.

WaterBox consists of energy hungry sensor/ actuator nodes to allow computational intensive algorithm deployments. An optimal hardware infrastructure will reduce the energy consumption even more than the evaluation results. Intuitively, the level of energy reduction will be closer to threshold violations (up to 57%) and state transmission (up to 64%) savings which indicate the actuator and sensors communication requirement respectively.

An additional benefit of applying periodic centralized or decentralized ETC approaches is the reduction of sensing rate. Continuous measurement retrieval from high energy demanding sensors (e.g. the water content sensor [212] which consumes 570 mJ per measurement) may lead to higher energy consumption than the communication process (e.g. low power wide area communication modules in [153] which consumes 1.5 to 42 J per 10 bytes). Further, based on our experimental results, higher sensing rates do not guarantee higher control performance. As future work, we will examine the aperiodic sensing scheduling over ETC techniques and the impact to the co-existed high sample rate algorithms for anomaly detection and validation. While in this chapter we focus on smart water networks, our proposed framework can be applied to a variety of Cyber-Physical Systems such as Smart Grids, Smart Transportation Systems and Automated Agriculture.

Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

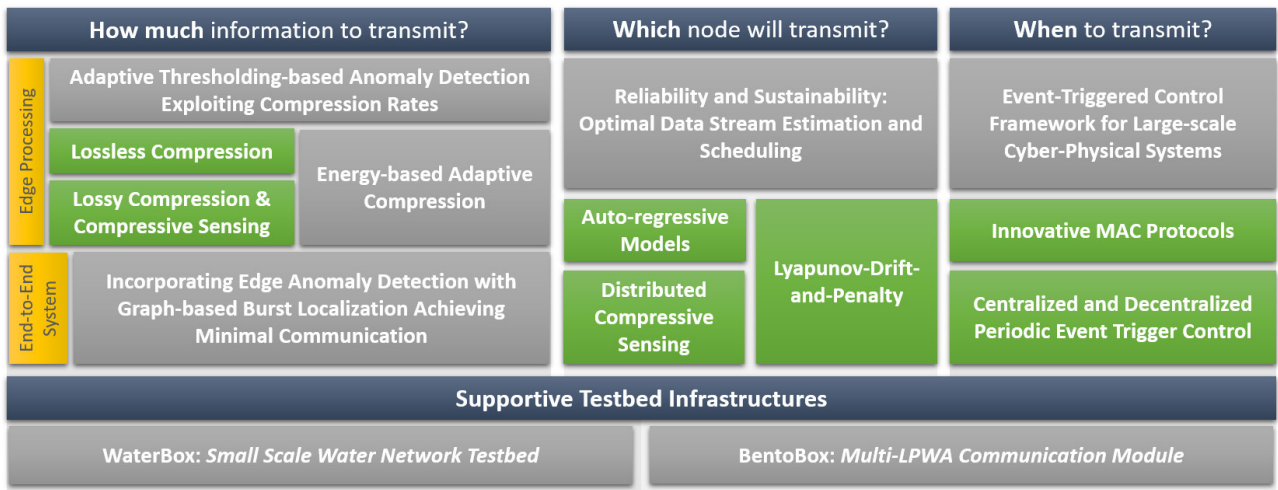


Figure 7.1: Summary of Thesis Achievements.

This thesis investigates communication optimization techniques over energy-constrained large-scale CPSs using as proof of concept smart water networks. The aim of this research is not only to extend the lifetime of sensor networks but also to introduce next generation Cyber-Physical Water Distribution Systems that enable automatic control and sustainable operation. To achieve this goal, this thesis introduces the concepts of edge data processing and reduction, communication scheduling adaptation, and event-driven communication based on control theory while answers the following three question:

- **How much information** must be transmitted by each node?
- **Which node** should transmit at every communication period?
- **When** has each sensor to transmit the local state of the physical layer?

7.1.1 How much information to transmit?

There is a movement in many CPS applications to transform acquired raw data from sensors to higher level of information by pushing processing to the edge. The main benefits of edge processing are the reduction in the amount of data that has to be transmitted wirelessly over the network and the decrease in latency between the decision making and control message reception. In Chapter 4, we introduce a variety of data reduction techniques by exploiting edge processing and the cooperation between the edge and the offline systems.

In our first attempt to reduce transmitted data, we applied lossless compression. During this process, a high correlation between high raw data fluctuation and achieved compression rates was observed. Based on this observation, we propose an edge lightweight anomaly detection algorithm based on adaptive thresholding which examines the compression rates instead of raw data. By applying our system mainly to water pressure data, the achieved benefits can be summarized as follows: (a) the execution of the algorithm requires only 10KB of memory and can be applied in embedded systems, (b) the raw data and consequently the communication is reduced by 55%, (c) the computation decreased by 98%, and (d) due to the normalized form of the compression rate metric, we show that the algorithm can be applied to any kind of raw data. To increase the precision of our anomaly detection approach by minimizing the false-positives and true-negatives of detection, a self-adaptive configuration mechanism based on the notion of active learning is introduced as well. Based on evaluation results, the mechanism improves the performance of anomaly detection compared to manual tuning by a further 15%.

Having the ability to precisely detect the anomalies to the edge, the next step was to examine ways to exploit this feature by incorporating it with off-line algorithm. Under this context, we combine the lightweight compression and anomaly detection with off-line graph topology analytics for water distribution networks to localize bursts. By evaluating our proposed system

to a medium scale test rig at Welsh Water, we show that our approach not only significantly reduces the amount of communications between sensor devices and the back end servers, but also can effectively localize water burst events by using the difference in the arrival times of the vibration anomalies detected at sensor locations. Transmitting only anomaly detection arrival time, our sensor nodes can save up to 99% communications compared with traditional periodical reporting situations. At the same time, our localization method can find the position of the anomaly for our particular scenario within 0.5m error. This data driven approach is significantly better than many hydraulic modelling approaches that at best identify a leak to the length of a given pipe which can be 10s of meters [149].

In spite of the significant data reduction provided by the above techniques, the power of Compressive Sensing (CS) was exploited to achieve higher compression rates, and at the same time maintain high reconstruction accuracy, of real high sample rate data. Applying CS on real pressure data, we achieved 35% improved performance compared with well-established lossless compression algorithms by introducing a minor reconstruction error (i.e. 0.0025 relative error). This approach employs a fixed sparsifying transformation of the sensor data, however, an adaptive degree of sparsity can increase the efficiency of the CS. To this end, distributed compressive sensing can be exploited, in conjunction with a joint sparsity assumption, to account for correlations either between the windows of each individual data stream or between distinct data streams.

Both data reduction techniques of lossless compression and CS have trade-offs. In lossless compression, the initial stream can be reconstructed without losing information, while CS, as a lossy approach, introduces reconstruction error. On the other hand, lossless compression achieves lower compression than CS. In order to balance these trade-offs, we proposed an adaptive compression scheme that optimally selects the compression technique minimizing the reconstruction error, given the sensor node battery state. The evaluation results over real pressure data reveal up to 66% communication reduction, in conjunction with an extension of the battery life by 46% compared the traditional periodic communication approaches.

Based on the above, data reduction and edge processing reduce the communication significantly.

However, the aforementioned techniques extends nodes' lifetime by making decisions locally to the edge without considering the state of other sensor nodes of the same network. Next, by answering the question of "Which node will transmit?", we consider a scheduling system that both optimizes the decision making of communication along the network and guarantees sustainable operation exploiting energy harvesting within smart water networks.

7.1.2 Which node will transmit?

As a typical CPS, smart water distribution networks require monitoring of underground water pipes with high sample rates for precise data analysis and water network control. Due to poor underground wireless channel quality and long-range communication requirements, high transmission power is typically adopted to communicate high-speed sensor data streams; posing challenges for long term sustainable monitoring.

In Chapter 5, we develop the first sustainable proven water sensing system that exploits energy harvesting opportunities from water flows and highly correlated data estimation. Our system does this by scheduling the transmission of a subset of the data streams, while other correlated streams are estimated using auto-regressive models based on the sound-velocity propagation of pressure signals inside water networks. In addition, Distributed Compressive Sensing (DCS) [20] is being examined as alternative to auto-regressive model-based estimation.

To compute the optimal scheduling policy, we formalize a stochastic optimization problem to maximize the estimation reliability, while ensuring the system's sustainable operation under dynamic conditions. We develop Data Transmission Scheduling (DTS), an asymptotically optimal scheme; and FAST-DTS, a lightweight online algorithm that can adapt to arbitrary energy and correlation dynamics.

Using over 170 days of real data from the Bristol Waters smart water system deployment and conducting in-vitro experiments to our small-scale testbed, WaterBox (see Chapter 3); our evaluation demonstrates that Fast-DTS significantly outperforms three alternatives, considering data reliability, energy utilization, and sustainable operation by achieving 99% estimation

accuracy and at the same time the lowest energy waste and transmission gaps.

In order to answer the question of "Which node will transmit?", we exploit data reduction and edge processing (i.e. anomaly detection) combined with a smart scheduling system to optimize the communication and ensure sustainable operation. However, a complete CPS system includes automatic control of the physical layer of the plant. By answering the following question of "When to transmit?", we minimize the communication by considering the state of the physical plant and, at the same time, ensure stability by involving control theory.

7.1.3 When to transmit?

Energy constraint long-range wireless sensor/ actuator based solutions are theoretically the perfect choice to support the next generation of city-scale cyber-physical systems. Traditional systems adopt periodic control which increases network congestion and actuations while burdens the energy consumption. Recent control theory studies overcome these problems by introducing aperiodic strategies, such as event trigger control. In spite of the potential savings, these strategies assume actuator continuous listening while ignoring the sensing energy costs.

In Chapter 6, we fill this gap, by enabling sensing and actuator listening duty-cycling and proposing two innovative MAC protocols for three decentralized event trigger control approaches. The WaterBox testbed, which emulates a smart water network, was modelled and extended to evaluate the impact of system parameters and the performance of each approach.

Results from more than 300 experiments reveal the predominance of the decentralized event-triggered control against the classic periodic control either in terms of actuation or communication by promising significant system lifetime extension. Specifically, the examined decentralized ETC strategies reduce the actuations by up to 35% and the size of actuations (i.e. valve movements) by up to 24% compared to traditional periodic control. The reduction to the number and size of actuations has as main impact the extension of the lifetime of physical plants and consequently the significant maintenance cost reduction by postponing expensive replacements of plant actuator assets.

Additionally, the proposed system reduces the communication requirement significantly for both actuator and sensor nodes. Based on the experimental results, by applying decentralized ETC, the actuators reduce required received information by up to 57%, while sensor node avoid the state transmission by up to 64%. This high level of communication reduction can enable the use of low-cost sensor/actuator nodes, which consume high energy to achieve underground long-range communication to city scale deployments.

By applying periodic sampling to centralized and decentralized ETC strategies, we also achieved the reduction of sensing rate. Continuous sampling from high energy demanding sensors may lead to higher energy consumption than the communication process. While in this thesis focuses on smart water networks, our proposed system can be applied to a variety of CPSs such as Smart Grids, Smart Transportation Systems and Automated Agriculture.

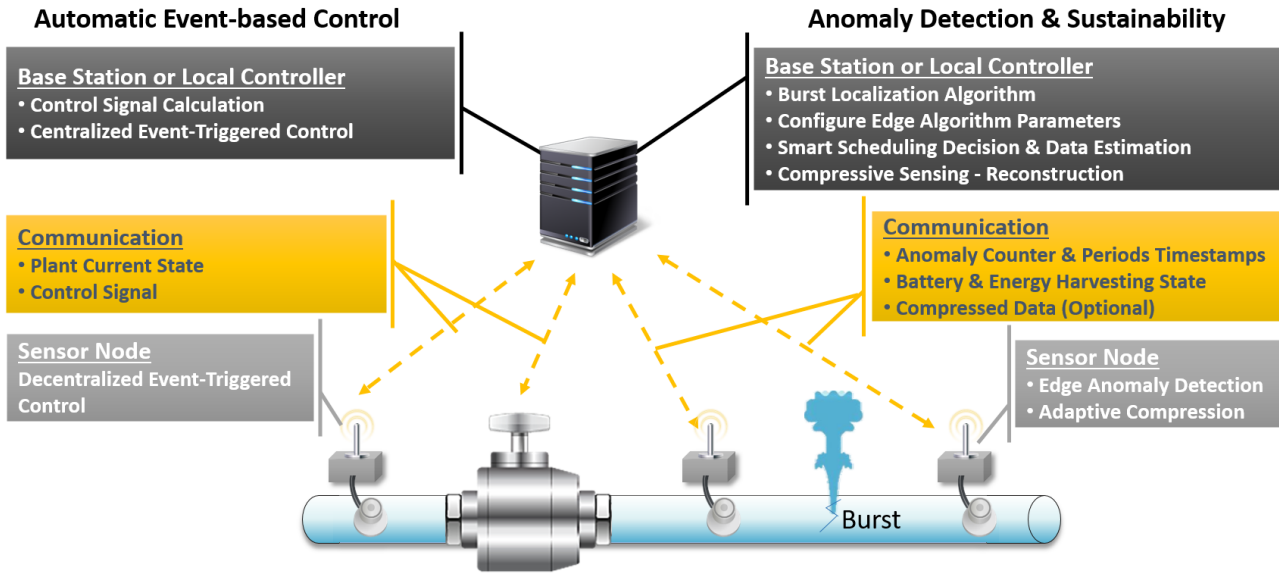


Figure 7.2: Thesis achievements as a system architecture.

once the questions "how much?, which?, and when?" have been answered, the next important question is: *"How can they be unified into a single system architecture to optimize the communication and ensure stability of the physical layer of the next generation CPSs?"* Figure 7.2 illustrates the unified system diagram that consists of two concurrent processes: (a) automatic event-based control and (b) anomaly detection and sustainability. In this system, the sensor nodes have to communicate with a local controller or base station only when a state threshold violation has occurred to ensure control stability or an anomaly detection has been observed

to enable burst localization detection. In both cases, the transmitted information is limited to the current state of the plant and the anomaly time stamp followed by the sensor node energy status respectively. Then, the base station is responsible to compute the control signals, localize bursts, schedule the sensor nodes, and recover or estimate the missing data. Optionally, the nodes can transmit compressed data for further analysis i.e. security or validation. However, the decision of additional data transmission for further analysis constitutes part of our future work.

7.2 Future Work

This thesis investigates ways to optimize the communication and propose next generation sustainable Cyber-Physical Systems (CPS) deployed in harsh environments, i.e. water networks, exploiting limited resources by combining Information and Control theory, Communication, and Computation theory (3C). This section outlines promising extensions of the work presented in this thesis which are related to the exploitation of distributed compressive sensing, the incorporation of event-triggered control with off-line processes and the implementation of our proposed algorithms in large scale systems.

7.2.1 Smart Scheduling and Distributed Compressive Sensing

The support of sustainable operation is critical in CPS applications. In Chapter 5, we exploit the high correlation of water pressure streams to estimate the data instead of performing transmission. The estimation achieved by using lightweight auto-regressive models. Our preliminary experiments show that distributed compressive sensing can be a candidate to replace auto-regressive models and achieve better estimation reliability. As the same chapter describes, our intention is not just to replace the estimation method but both to improve distributed compressive sensing operation and ensure sustainable operation at the same time. This will be achieved by dynamically adapt the measurement matrices Φ_i by considering the available resources of all the sensor nodes along the network. Further, we are working to extend our

approach by examining the impact of data estimation with regard to the automatic control process.

7.2.2 Incorporate Validation, Security and Event-Triggered Control

Automatic control constitutes the main ingredient in CPSs. In Chapter 6, we minimize the communication by transmitting the state of the plant only when it is necessary and covering the requirements of stability concepts. However, this approach does not consider the data requirements of other off-line algorithms of the system i.e. security and verification algorithms. As future work, we will examine the aperiodic sensing scheduling over ETC techniques and the impact to the co-existed high sample rate algorithms for security and validation. Under the same context, our future system will be adapted between the minimum communication mode based on ETC theory to maximum transmission rate covering security enhancement requirements and ensuring quality of service and sustainability. Another improvement toward sustainability will be the incorporation of the harvesting and battery levels as state parameters of the plant. With that way, the control theory will stabilize both the state of the plant and the energy state of wireless sensor/actuator network.

7.2.3 Large Scale Implementation

In this thesis, we have evaluated our proposed systems by using real data gathered from existing water networks or emulating CPS scenarios in small to medium scale testbeds. Our ultimate goal is to apply our research to real-time city-scale systems. Based on the experience we have gained by creating WaterBox and BentoBox (see Chapter 3), this goal is not trivial and requires multidisciplinary collaborations between industry and academia.

Additionally, in a large scale system, an innovative hierarchy of CPS assets will be vital. For example, a large scale system can be divided into smaller controllable areas which will constitute the first layer of the hierarchy. Then, a second layer will be responsible to orchestrate the controllers of each area in order to achieve global optimal operations. This layer-based hierarchy

will introduce further complexity to the communication and the current star network topology will be inappropriate. However, a more complex large-scale network topology, i.e. mesh network topology, will reveal other critical challenges, such as the inability of delay guarantees for control systems.

Overall, our research can be applied to different CPSs which share similar behavior. For example, a smart irrigation system can be considered as the most potential candidate. Other applications include: heating and cooling control for large building, oil pipe network control in deserted areas, smart traffic systems, and smart grids.

Appendix A

Proof in Chapter 2

This appendix provides the proof of Theorem 2.1 in Chapter 2.

A.1 Proof of Rate Stability

Proof. To prove that if the queue length is rate stable, i.e. $\lim_{t \rightarrow \infty} \frac{Q_i(t)}{t} = 0$, then the time average $\bar{y}_i(t) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} y(\tau) \leq 0$, we use the telescoping sum. Based on theory, the telescopic sum is defined as follows:

$$\sum_{i=1}^{n-1} (a_i - a_{i+1}) = (a_1 - a_2) + (a_2 - a_3) + \cdots + (a_{n-1} - a_n) = (a_1 - a_n) \quad (\text{A.1})$$

By converting Equation (2.12) to inequality, we have

$$Q_i(\tau + 1) \geq Q_i(\tau) + y_i(\tau) \Leftrightarrow Q_i(\tau + 1) - Q_i(\tau) \geq y_i(\tau) \quad (\text{A.2})$$

By time averaging both side of Inequality (A.2), then applying (A.1) and finding the limit to infinity, we derive

$$\begin{aligned}
\sum_{\tau=0}^{t-1} (Q_i(\tau+1) - Q_i(\tau)) &\geq \sum_{\tau=0}^{t-1} y_i(\tau) \\
Q_i(t) - Q_i(0) &\geq \sum_{\tau=0}^{t-1} y_i(\tau) \\
\frac{1}{t} (Q_i(t) - Q_i(0)) &\geq \frac{1}{t} \sum_{\tau=0}^{t-1} y_i(\tau) \\
\lim_{t \rightarrow \infty} \frac{(Q_i(t) - Q_i(0))}{t} &\geq \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} y_i(\tau) \tag{A.3}
\end{aligned}$$

Then, $\frac{Q_i(t)}{t} \rightarrow 0$ and implies that $\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} y_i(\tau) \leq 0$

□

Appendix B

Proofs in Chapter 5

This appendix provides the proof of Prof. Shusen Yang for Theorem 5.1 in Chapter 5.

B.1 Proof of Sustainable Operation

Proof. Consider a scheduling decision $\mathcal{Y}(t)$ and an arbitrary node $j \in \mathcal{S}_b(t)$, $j \notin \mathcal{Y}(t)$. Let $\mathcal{Y}'(t) = \mathcal{Y}(t) \cup \{j\}$. Let $\varphi_j(t)$ be the difference of (5.14) when scheduling decisions $\mathcal{Y}(t)$ and $\mathcal{Y}'(t)$ are adopted, we have

$$\begin{aligned}\varphi_j(t) &= V \sum_{i \in \mathcal{S}} (U_i(rlb_i(\mathcal{Y}(t))) + \sum_{i \in \mathcal{Y}(t)} a_i(t) - (V \sum_{i \in \mathcal{S}} (U_i(rlb_i(\mathcal{Y}'(t))) + \sum_{i \in \mathcal{Y}'(t)} a_i(t))) \\ &= V \sum_{i \in \mathcal{S}} (U_i(rlb_i(\mathcal{Y}(t))) - U_i(rlb_i(\mathcal{Y}'(t)))) - a_j(t)\end{aligned}$$

In order to ensure $\varphi_j(t) > 0$ when $B_j(t) < B_{\text{exp}}$, we have $\forall E_j^{tr}(t), rlb_i(\mathcal{Y}(t)), rlb_i(\mathcal{Y}'(t))$

$$V \sum_{i \in \mathcal{S}} (U_i(rlb_i(\mathcal{Y}(t))) - U_i(rlb_i(\mathcal{Y}'(t)))) > E_j^{tr}(t)(B_j(t) - B_{\text{exp}}),$$

Consider the facts that $0 \leq rlb_i() \leq 1$, $E_j^{tr}(t) \leq E_{\text{max}}$, and $U_i()$ is a non-decreasing function, above inequality will always hold when $V < \frac{\sum_{i \in \mathcal{S}} (U_i(1) - U_i(0))}{E_{\text{max}} B_{\text{exp}}}$. $\square$

B.2 Proof of Asymptotic Optimality

Proof. To prove the optimality of DTS, we divide the system time horizon, $[1, t_{\text{end}}]$, into K successive frames with size T intervals (i.e. $t_{\text{end}} = KT$). We assume that there exists an *ideal* algorithm operating at the first interval of each frame $t = (k - 1)T + 1, 1 \leq k \leq K$, which can obtain full information regarding the dynamics of the smart water system for the future T slots (which is impossible in practice). Based on future knowledge, the ideal algorithm solves the following problem:

$$\begin{aligned} & \max_{\mathcal{Y}(t)} \quad \frac{1}{T} \sum_{t=kT-T+1}^{kT} \sum_{i \in \mathcal{S}} U_i(rlb_i(\mathcal{Y}(t))) \\ & \text{subject to} \end{aligned} \quad (\text{B.1})$$

Constraints(5.9) – (5.11)

$$\frac{1}{T} \sum_{t=kT-T+1}^{kT} (h_i(t) - y_i(t)E_i^{\text{tr}}(t) - E_i^{\text{in}}(t)) \geq 0, \forall i \quad (\text{B.2})$$

Let $\Phi(t) = \sum_{i \in \mathcal{S}} U_i(rlb_i(\mathcal{Y}(t)))$ achieved by the DTS algorithm, and $\Phi^{\text{ideal}}(k, T)$ denote the utility achieved by the ideal algorithm over each frame $1 \leq k \leq K$. We aim to prove the following inequality

$$\frac{1}{t_{\text{end}}} \sum_1^{t_{\text{end}}} \Phi(t) \geq \frac{1}{K} \sum_{k=1}^K \Phi^{\text{ideal}}(k, T) - \frac{MT - N}{V} \quad (\text{B.3})$$

where $M = \frac{1}{2}|\mathcal{S}| \max(h_{\text{max}}^2, E_{\text{max}}'^2)$ and $N = \frac{1}{2t_{\text{end}}}|\mathcal{S}|B_{\text{max}}^2$. is a constant value and $E_{\text{max}}'^2$ represents the upper bound energy costs for transmission and in-node operation for each interval, i.e. $E_i^{\text{tr}}(t) + E_i^{\text{in}}(t) \leq E_{\text{max}}'^2, \forall i, \forall t$.

Inequality (B.3) shows that parameter V can be set as large as desired to force MT/V to be arbitrarily small. Specifically, Inequality (B.3) also demonstrates that when $T = t_{\text{end}}$, the optimal average aggregated utilities of reliability can be asymptotically achieved by DTS, as $V \rightarrow \infty$. Now we prove the Inequality (B.3). Let $\Delta B_i(t) = B_i(t + 1) - B_i(t)$ and define the

Lyapunov function

$$L(t) = \frac{1}{2} \sum_{i \in \mathcal{S}} (B_i(t) - B_{\text{exp}})^2 \quad (\text{B.4})$$

Consider its one-interval drift plus penalty

$$\begin{aligned} \Delta_1 L(t) &= L(t+1) - L(t) - \Phi(t) \\ &= \frac{1}{2} \sum_{i \in \mathcal{S}} ((B_i(t+1) - B_{\text{exp}})^2 - (B_i(t) - B_{\text{exp}})^2) - \Phi(t) \\ &= \frac{1}{2} \sum_{i \in \mathcal{S}} (\Delta B_i^2(t) + 2\Delta B_i(t)(B_i(t) - B_{\text{exp}})) - \Phi(t) \\ &\leq_a M - V\Phi(t) - \sum_{i \in \mathcal{S}} (B_i(t) - B_{\text{exp}})(y_i(t)E_i^{tr}(t) + E_i^{in}(t) - h_i(t)) \end{aligned} \quad (\text{B.5})$$

Inequality (B.5) is because of the fact that

$$M = \frac{1}{2} |\mathcal{S}| \max(h_{\text{max}}^2, E_{\text{max}}^2) \geq \frac{1}{2} \sum_{i \in \mathcal{S}} \Delta B_i^2(t)$$

It can be easily seen that the DTS algorithm greedily minimizes the right-hand-side of inequality (B.5) at every interval t . Now we consider the T -interval drift of the Lyapunov function plus T -interval penalty

$$\begin{aligned} \Delta_T L(t) - V \sum_{t=kT-T+1}^{kT} \Phi(t) \\ &= L(kT) - L(kT - T + 1) - V \sum_{t=kT-T+1}^{kT} \Phi(t) \\ &= \sum_{t=kT-T+1}^{kT} (\Delta_1 L(t) - \Phi(t)) \\ &\leq_a MT + \sqrt{2M} \frac{T(T-1)}{2} \\ &\quad + \sum_{t=kT-T+1}^{kT} \left(\sum_{i \in \mathcal{S}} (B_i(t) - B_{\text{exp}})(y_i(t)E_i^{tr}(t) + E_i^{in}(t) - h_i(t)) - V\Phi(t) \right) \\ &\leq_b MT^2 - VT\Phi^{ideal}(k, T) \\ &\quad + \sum_{t=kT-T+1}^{kT} \sum_{i \in \mathcal{S}} (B_i(t) - B_{\text{exp}})(y_i^{ideal}(t)E_i^{tr}(t) + E_i^{in}(t) - h_i(t)) \end{aligned} \quad (\text{B.6})$$

where the inequality $\leq_a$ is based on inequality (B.5), the sum of $\Delta_1 L(t) - V\Phi(t)$ over T slots of the k th frame, and the fact that each queue backlog does not change by more than $(t - (kT - T + 1)) \max(h_{\max}, E'_{\max})$ for any slot $kT - T + 1 \leq t \leq kT$; the inequality $\leq_b$ follows from $M \geq \sqrt{M}, \forall M \geq 1$, and the fact that our DTS algorithm minimizes the right-hand side of the inequality $\leq_a$ over all possible transmission scheduling decisions, including the decisions of the ideal algorithm $y_i^{ideal}(t), i \in \mathcal{S}$, which achieves the optimal reliability utility of the ideal algorithm $\Phi^{ideal}(k, T)$. Consider (B.6) and the fact that $y_i^{ideal}(t)$ satisfies constraint (B.2), we have

$$\Delta_T L(t) - V \sum_{t=kT-T+1}^{kT} \Phi(t) \leq MT^2 - VT\Phi^{ideal}(k, T) \quad (\text{B.7})$$

Taking a telescopic sum of the inequality (B.7) over $k \in \{1, \dots, K\}$ and dividing both side by VKT , we get

$$\frac{L(KT + T) - L(1)}{KVT} - \frac{1}{KT} \sum_{t=1}^{KT} \Phi(t) \leq \frac{MT}{V} - \frac{1}{K} \sum_{k=1}^K \Phi^{ideal}(k, T)$$

Consider $\frac{L(KT+T)-L(1)}{KT} \geq N = \frac{1}{2t_{\text{end}}} |\mathcal{S}| B_{\max}^2$, we have

$$\frac{1}{KT} \sum_{t=1}^{KT} \Phi(t) \geq \frac{1}{K} \sum_{k=1}^K \Phi^{ideal}(k, T) - \frac{MT - N}{V}$$

Since $t_{\text{end}} = KT$, Inequality (B.3) obviously holds. $\square$

Appendix C

WaterBox dynamical model in Chapter 6

This appendix provides the WaterBox dynamical model of Anqi Fu in Chapter 6.

C.1 Hydraulic component definitions

Figure C.1 illustrates the important indexes and component positions: 10 locates right after the powerful pump; 21, 22, and 23 locate right before the tee joint; 31, 32, and 33 locate right after the tee joint, on the branch; 41 and 42 locate right after the tee joint, on the line; 5i and 6i locate right before and after in-valves; 7i locate right after the pipe ports; 8i and 9i locate right before and after out-valves.

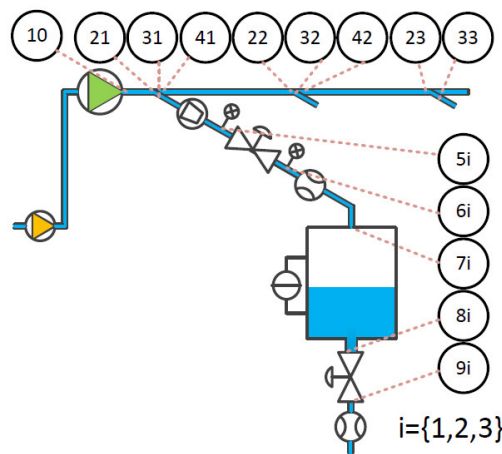


Figure C.1: WaterBox hydraulic components: indexes and positions.

The important parameters and their descriptions are shown in Table C.1. Note that, index j and k have two digits while i have a single digit.

Table C.1: WaterBox hydraulic component definition

Symbol	Unit	Description
h_a	m	head of the flow from assistant pump
q_a	m^3/s	rate of the flow from assistant pump
p_a	Pa	pressure of the flow from assistant pump
h_p	m	head of the flow from powerful pump
q_p	m^3/s	rate of the flow from powerful pump
p_p	Pa	pressure of the flow from powerful pump
Δh_{ap}	m	height difference between assistant pump and powerful pump
g	m/s^2	local gravity acceleration
d_p	m	diameter of the pipe
a_p	m^2	cross section of the pipe, $a_p = \pi d_p^2/4$
ρ	kg/m^3	density of the fluid, for water $\rho = 1000kg/m^3$
α_i^i	rad	angle of in-valve i
$\alpha_{i\max}^i$	rad	angle of in-valve i when it is widely open, $\alpha_{i\max}^i = 2\pi$
$\alpha_{i\min}^i$	rad	angle of in-valve i when it is minimum open
α_o^i	rad	angle of out-valve i
$\alpha_{o\max}^i$	rad	angle of out-valve i when it is widely open, $\alpha_{o\max}^i = 2\pi$
$\alpha_{o\min}^i$	rad	angle of out-valve i when it is minimum open
r_i^i	m	diameter of the in-valve i , we assume $r_i^i = a_p$
r_o^i	m	diameter of the out-valve i , we assume $r_o^i = a_p$
q_i^i	m^3/s	in flow rate of the small tank i
q_o^i	m^3/s	out flow rate of the small tank i
SG	—	relative specific gravity of fluid with reference to water (for water $SG=1$)
K_L	—	minor loss coefficient
K_v	—	flow coefficient
C_v	—	flow coefficient, $C_v = 1.156K_v$
K_{vi}^i	—	flow coefficient of i -th in-valve
K_{vo}^i	—	flow coefficient of i -th out-valve
H_i	m	height of small tank i
D_i	m	diameter of small tank i
A_i	m^2	cross section of small tank i , $A_i = \pi D_i^2/4$
d_i	m	diameter of the hole in small tank i
a_i	m^2	cross section of the hole in small tank i , $a_i = \pi d_i^2/4$
Δh_i	m	height from bottom of small tank i to the core of out-valve i
h_i	m	height of water level in small tank i
h_i'	m	desired height of water level in small tank i
Re	—	Reynolds number
μ	$Pa.s$	dynamic viscosity of the fluid, for water with 101.325kpa, 20°C, $\mu = 1.01 \times 10^{-3} Pa.s$
η	—	efficiency of the turbine
p_j	Pa	pressure at point j
h_j	m	head at point j
v_j	m/s	flow rate at point j
Δh_j	m	height difference between point j and powerful pump

C.2 WaterBox Model

Having defined the main parameters and components, in this section, we give the mathematic model of the WaterBox.

C.2.1 Dynamics at different points

The dynamic at each point and the relations between different points are shown below:

At 10:

$$\begin{cases} h_{10}(t) = \frac{v_{10}^2(t)}{2g} + \frac{p_{10}(t)}{\rho g} \\ q_{10}(t) = v_{10}(t)a_p \end{cases} \quad (C.1)$$

From 10 to 21:

$$\begin{cases} h_{10}(t) = h_{21}(t) \\ q_{10}(t) = q_{21}(t) \end{cases} \quad (C.2)$$

At 21:

$$\begin{cases} h_{21}(t) = \frac{v_{21}^2(t)}{2g} + \frac{p_{21}(t)}{\rho g} - \Delta h_{31} \\ q_{21}(t) = v_{21}(t)a_p \end{cases} \quad (C.3)$$

From 21 to 31 and 41:

$$\begin{cases} p_{21}(t) = p_{31}(t) = p_{41}(t) \\ q_{21}(t) = q_{31}(t) + q_{41}(t) \end{cases} \quad (C.4)$$

At 31:

$$\begin{cases} h_{31}(t) = \frac{v_{31}^2(t)}{2g} + \frac{p_{31}(t)}{\rho g} - \Delta h_{31} \\ q_{31}(t) = v_{31}(t)a_p \end{cases} \quad (C.5)$$

At 41:

$$\begin{cases} h_{41}(t) = \frac{v_{41}^2(t)}{2g} + \frac{p_{41}(t)}{\rho g} - \Delta h_{31} \\ q_{41}(t) = v_{41}(t)a_p \end{cases} \quad (C.6)$$

From 31 to 22:

$$\begin{cases} h_{31}(t) = h_{22}(t) \\ q_{31}(t) = q_{22}(t) \end{cases} \quad (\text{C.7})$$

At 22:

$$\begin{cases} h_{22}(t) = \frac{v_{22}^2(t)}{2g} + \frac{p_{22}(t)}{\rho g} - \Delta h_{32} \\ q_{22}(t) = v_{22}(t)a_p \end{cases} \quad (\text{C.8})$$

From 22 to 32 and 42:

$$\begin{cases} p_{22}(t) = p_{32}(t) = p_{42}(t) \\ q_{22}(t) = q_{32}(t) + q_{42}(t) \end{cases} \quad (\text{C.9})$$

At 32:

$$\begin{cases} h_{32}(t) = \frac{v_{32}^2(t)}{2g} + \frac{p_{32}(t)}{\rho g} - \Delta h_{32} \\ q_{32}(t) = v_{32}(t)a_p \end{cases} \quad (\text{C.10})$$

At 42:

$$\begin{cases} h_{42}(t) = \frac{v_{42}^2(t)}{2g} + \frac{p_{42}(t)}{\rho g} - \Delta h_{32} \\ q_{42}(t) = v_{42}(t)a_p \end{cases} \quad (\text{C.11})$$

From 42 to 23:

$$\begin{cases} h_{42}(t) = h_{23}(t) \\ q_{42}(t) = q_{23}(t) \end{cases} \quad (\text{C.12})$$

At 23:

$$\begin{cases} h_{23}(t) = \frac{v_{23}^2(t)}{2g} + \frac{p_{23}(t)}{\rho g} - \Delta h_{33} \\ q_{23}(t) = v_{23}(t)a_p \end{cases} \quad (\text{C.13})$$

From 23 to 33:

$$\begin{cases} p_{23}(t) = p_{33}(t) \\ q_{23}(t) = q_{33}(t) \end{cases} \quad (\text{C.14})$$

At 33:

$$\begin{cases} h_{33}(t) = \frac{v_{33}^2(t)}{2g} + \frac{p_{33}(t)}{\rho g} - \Delta h_{33} \\ q_{33}(t) = v_{33}(t)a_p \end{cases} \quad (\text{C.15})$$

From 3i to 5i:

$$\begin{cases} q_{3i}(t)\rho gh_{3i}(t) = q_{5i}(t)\rho gh_{5i}(t) + \eta(q_{3i}, h_{3i})q_{3i}\rho gh_{3i}(t) \\ q_{3i}(t) = q_{5i}(t) \end{cases} \quad (C.16)$$

At 5i:

$$\begin{cases} h_{5i}(t) = \frac{v_{5i}^2(t)}{2g} + \frac{p_{5i}(t)}{\rho g} - \Delta h_{3i} \\ q_{5i}(t) = v_{5i}(t)a_p \end{cases} \quad (C.17)$$

From 5i to 7i:

$$\begin{cases} q_{5i}(t) = \frac{K_{vi}^i(\alpha_i^i(t))}{3600} \sqrt{p_{5i}(t)} \end{cases} \quad (C.18)$$

C.2.2 System model

Based on the dynamic analysis of different points, together with the relation between $q_{10}(t)$ and $p_{10}(t)$ obtained from the pump curves, we obtain the system model of supply layer as:

$$\begin{cases} q_{10}(t) = f(p_{10}(t)) \\ p_{10}(t) + \rho g \Delta h_{31} = \frac{\rho g}{1 - \eta(q_{51}, p_{51})} (\eta(q_{51}, p_{51}) \frac{q_{51}(t)^2}{2a_p^2 g} + \frac{p_{51}(t)}{\rho g} - \eta(q_{51}, p_{51}) \Delta h_{31}) \\ p_{10}(t) + \rho g \Delta h_{32} = \frac{\rho g}{1 - \eta(q_{52}, p_{52})} (\eta(q_{52}, p_{52}) \frac{q_{52}(t)^2}{2a_p^2 g} + \frac{p_{52}(t)}{\rho g} - \eta(q_{52}, p_{52}) \Delta h_{32}) \\ p_{10}(t) + \rho g \Delta h_{33} = \frac{\rho g}{1 - \eta(q_{53}, p_{53})} (\eta(q_{53}, p_{53}) \frac{q_{53}(t)^2}{2a_p^2 g} + \frac{p_{53}(t)}{\rho g} - \eta(q_{53}, p_{53}) \Delta h_{33}) \\ q_{10}(t) = q_{51}(t) + q_{52}(t) + q_{53}(t) \\ q_{51}(t) = \frac{K_{vi}^1(\alpha_i^1(t))}{3600} \sqrt{P_{51}(t)} \\ q_{52}(t) = \frac{K_{vi}^2(\alpha_i^2(t))}{3600} \sqrt{P_{52}(t)} \\ q_{53}(t) = \frac{K_{vi}^3(\alpha_i^3(t))}{3600} \sqrt{P_{53}(t)} \end{cases} \quad (C.19)$$

Therefore, we have 8 parameters with 8 functions. If there is no turbine, that is $\eta = 0$, we have a simplified model:

$$\left\{ \begin{array}{l} q_{10}(t) = f(p_{10}(t)) \\ p_{10}(t) + \rho g \Delta h_{31} = p_{51}(t) \\ p_{10}(t) + \rho g \Delta h_{32} = p_{52}(t) \\ p_{10}(t) + \rho g \Delta h_{33} = p_{53}(t) \\ q_{10}(t) = q_{51}(t) + q_{52}(t) + q_{53}(t) \\ q_{51}(t) = \frac{K_{vi}^1(\alpha_i^1(t))}{3600} \sqrt{P_{51}(t)} \\ q_{52}(t) = \frac{K_{vi}^2(\alpha_i^2(t))}{3600} \sqrt{P_{52}(t)} \\ q_{53}(t) = \frac{K_{vi}^3(\alpha_i^3(t))}{3600} \sqrt{P_{53}(t)} \end{array} \right. \quad (\text{C.20})$$

Note that, the mapping $q_{10}(t) = f(p_{10}(t))$ is a relation got from the pump curves. There are two curves, with powerful pump on and off. More specifically, we define $q_{10}(t) = f_1(p_{10}(t))$ and $q_{10}(t) = f_2(p_{10}(t))$ separately, with $f_1(\cdot)$ being the mapping for the situation when both weak pump and powerful pump are working; while $f_2(\cdot)$ being the mapping for the situation when only the weak pump is working. Thus, there are 8 curves to be fitted: $f_1(\cdot)$, $f_2(\cdot)$, K_{vi}^i , $\eta(q_{5i}, p_{5i})$.

The system model of the DMA layer is:

$$\dot{h}_i(t) = \frac{1}{A_i} (q_i^i(t) - \frac{K_{vo}^i(\alpha_o^i(t))}{3600} \sqrt{\rho g (\Delta h_i + h_i(t))}) \quad (\text{C.21})$$

or

$$\dot{h}_i(t) = \frac{1}{A_i} (q_i^i(t) - a_i \sqrt{2gh_i(t)}) + w(t) \quad (\text{C.22})$$

When the out-valves are widely open, a simplified model of the DMA layer is:

$$\dot{h}_i(t) = \frac{1}{A_i} (q_i^i(t) - a_i \sqrt{2gh_i(t)}) \quad (\text{C.23})$$

Bibliography

- [1] SWIG, “Data standards and protocols for real-time communications in the water industry,” <http://www.swig.org.uk/past-swig-events/2014-2/>, 2014, online; accessed 10-October-2014.
- [2] I. Stoianov, L. Nachman, S. Madden, T. Tokmouline, and M. Csail, “Pipenet: A wireless sensor network for pipeline monitoring,” in *Proc. IPSN*, 2007, pp. 264–273.
- [3] M. Allen, A. Preis, M. Iqbal, and A. J. Whittle, “Water distribution system monitoring and decision support using a wireless sensor network,” in *Proc. IEEE SNPD*, 2013, pp. 641–646.
- [4] A. Johnson and J. Burton, “Water torture: 3,300,000,000 litres are lost every single day through leakage,” *The Independent*, 2010.
- [5] T. Guardians, “Water leakages: which company is the worst?” <https://www.theguardian.com/news/datablog/2012/may/08/water-companies-leaks-target-drought#data>, 2015.
- [6] Ofwat, “The economic regulator of the water sector in england and wales,” <http://www.ofwat.gov.uk/>, 2015.
- [7] B. Aghaei, “Using wireless sensor network in water, electricity and gas industry,” in *Proc. IEEE ICECT*, vol. 2, 2011, pp. 14–17.
- [8] A. Santos and M. Younis, “A sensor network for non-intrusive and efficient leak detection in long pipelines,” in *Proc. IEEE WD*, 2011, pp. 1–6.

- [9] Z. Wang, X. Hao, and D. Wei, “Remote water quality monitoring system based on WSN and GPRS,” *Instrument Technique and Sensor*, vol. 1, p. 018, 2010.
- [10] M. Allen, A. Preis, M. Iqbal, and A. J. Whittle, “Water distribution system monitoring and decision support using a wireless sensor network,” in *Proc. IEEE SNPD*, 2013, pp. 641–646.
- [11] “NEC and Imperial College London Smart Water Project,” <https://wp.doc.ic.ac.uk/aese/project/smart-water/>, 2016, online; accessed 10-February-2016.
- [12] R. Wright, I. Stoianov, P. Parpas, K. Henderson, and J. King, “Adaptive water distribution networks with dynamically reconfigurable topology,” *Journal of Hydroinformatics*, 2014.
- [13] I2O, “I2O automatic PRV optimization sytem,” <http://www.i2owater.com>, 2015, online; accessed 14-September-2016.
- [14] Derceto, “Aquadapt valve and pump scheduling optimization system,” <http://www.derceto.com>, 2015, online; accessed 14-September-2016.
- [15] B. Moyer, “Low power, wide area: A survey of longer-range IoT wireless protocols,” *Electronic Engineering Journal*, September 2015.
- [16] K. C. Barr and K. Asanovi, “Energy-aware lossless data compression,” *ACM Trans. Comput. Syst.*, no. 24, pp. 250–291, 2006.
- [17] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [18] S. Papadimitriou, J. Sun, C. Faloutsos, and S. Y. Philip, “Dimensionality reduction and filtering on time series sensor streams,” in *Managing and Mining Sensor Data*. Springer, 2013, pp. 103–141.

- [19] M. J. Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [20] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, “Distributed compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [21] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 2011.
- [22] M. Lemmon, “Event-triggered feedback in control, estimation, and optimization,” in *Networked Control Systems*. Springer, 2010, pp. 293–358.
- [23] L. E. Ormsbee, T. M. Walski, D. V. Chase, and W. W. Sharp, “Methodology for improving pump operation efficiency,” *Journal of water resources planning and management*, vol. 115, no. 2, pp. 148–164, 1989.
- [24] B. Ulanicki, J. Kahler, and H. See, “Dynamic optimization approach for solving an optimal scheduling problem in water distribution systems,” *Journal of Water Resources Planning and Management*, vol. 133, no. 1, pp. 23–32, 2007.
- [25] B. Ulanicki, H. AbdelMeguid, P. Bounds, and R. Patel, “Pressure control in district metering areas with boundary and internal pressure reducing valves,” in *The 10th International Water Distribution System Analysis Conference*, 2008, pp. 17–20.
- [26] E. Abraham and E. C. Kerrigan, “Optimal active control and optimization of a wave energy converter,” *IEEE Trans. Sustain. Energy*, vol. 4, no. 2, pp. 324–332, 2013.
- [27] I. Stoianov, “InfraSense Labs,” <https://infrasense.net/aboutUs.html>, 2016, online; accessed 14-October-2016.
- [28] A. Hoskins and I. Stoianov, “Infrasense: A distributed system for the continuous analysis of hydraulic transients,” *Procedia Engineering*, vol. 70, pp. 823–832, 2014.
- [29] Semtech, “LoRa: The Ultimate Long Range Wireless Solution,” <http://www.semtech.com/wireless-rf/lora.html>, 2017, [Online; accessed 26-January-2017].

- [30] NWave, “Nwave radio modules,” <http://www.nwave.io/>, 2017, [Online; accessed 26-January-2017].
- [31] Sigfox, “Sigfox radio modules,” <https://www.sigfox.com/>, 2017, [Online; accessed 26-January-2017].
- [32] Maxim Integrated, “An Introduction to Spread-Spectrum Communications,” <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1890>, 2016, [Online; accessed 12-October-2016].
- [33] TI, “Long-range RF communication,” www.ti.com/lit/wp/swry006/swry006.pdf, 2016.
- [34] Antenna Theory, “Beamwidths and Sidelobe Levels,” <http://www.antenna-theory.com/basics/radPatDefs.php>, 2016, [Online; accessed 12-October-2016].
- [35] Microwaves&rf, “Understanding UNB modulation,” mwrf.com/markets/understanding-ultra/-narrowband-modulation, 2016.
- [36] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, “Data collection, storage, and retrieval with an underwater sensor network,” in *Proc. ACM SenSys*, 2005, pp. 154–165.
- [37] Z. A. Eu, H.-P. Tan, and W. K. Seah, “Routing and relay node placement in wireless sensor networks powered by ambient energy harvesting,” in *IEEE WCNC*, 2009, pp. 1–6.
- [38] K. C. Barr and K. Asanović, “Energy-aware lossless data compression,” *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 250–291, 2006.
- [39] E. F. Nakamura, A. A. Loureiro, and A. C. Frery, “Information fusion for wireless sensor networks: Methods, models, and classifications,” *ACM CSUR*, vol. 39, no. 3, p. 9, 2007.
- [40] A. Abdelgawad and M. Bayoumi, *Resource-Aware data fusion algorithms for wireless sensor networks*. Springer Science & Business Media, 2012, vol. 118.
- [41] A. Sharaf, J. Beaver, A. Labrinidis, and K. Chrysanthis, “Balancing energy efficiency and quality of aggregate data in sensor networks,” *VLDB*, vol. 13, no. 4, pp. 384–403, 2004.

- [42] E. Cayirci, “Data aggregation and dilution by modulus addressing in wireless sensor networks,” *IEEE Comm. Letters*, vol. 7, no. 8, pp. 355–357, 2003.
- [43] T. He, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Aida: Adaptive application-independent data aggregation in wireless sensor networks,” *ACM Trans. Embed. Comput. S.*, vol. 3, no. 2, pp. 426–457, 2004.
- [44] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, “Synopsis diffusion for robust aggregation in sensor networks,” in *Proc. ACM SenSys*, 2004, pp. 250–262.
- [45] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, “Medians and beyond: new aggregation techniques for sensor networks,” in *Proc. ACM SenSys*, 2004, pp. 239–249.
- [46] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, 1977.
- [47] ———, “Compression of individual sequences via variable-rate coding,” *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, 1978.
- [48] J. Cleary and I. Witten, “Data compression using adaptive coding and partial string matching,” *IEEE Trans. Comm.*, vol. 32, no. 4, pp. 396–402, 1984.
- [49] M. Burrows and D. J. Wheeler, “A block-sorting lossless data compression algorithm,” 1994.
- [50] Atmel, “Atmel microcontroller selector,” http://www.atmel.com/products/selector_overview.aspx, 2017, [Online; accessed 26-January-2017].
- [51] bzip2, “bzip2 and libbzip2,” <http://www.bzip.org>, 2010, [Online; accessed 20-September-2016].
- [52] 7zip, “LZMA SDK (Software Development Kit),” <http://www.7-zip.org/sdk.html>, 2015, [Online; accessed 20-September-2016].
- [53] Oberhumer, “LZO Version 2.09,” <http://www.oberhumer.com/opensource/lzo/>, 2015, [Online; accessed 20-September-2016].

- [54] C. Sadler, “LZW,” <https://sites.google.com/site/cmsadler/>, 2015, [Online; accessed 20-September-2016].
- [55] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [56] S. Sathe, T. G. Papaioannou, H. Jeung, and K. Aberer, “A survey of model-based sensor data acquisition and management,” in *Managing and Mining Sensor Data*. Springer, 2013, pp. 9–50.
- [57] S. Mallat, “A wavelet tour of signal processing: The sparse way.” Academic press, 2009.
- [58] I. Daubechies, “Ten lectures on wavelets.” SIAM, 1992.
- [59] S. K. Sharma, E. Lagunas, S. Chatzinotas, and B. Ottersten, “Application of compressive sensing in cognitive radio communications: A survey,” 2016.
- [60] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.
- [61] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [62] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [63] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [64] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [65] M. Elad, M. A. Figueiredo, and Y. Ma, “On the role of sparse and redundant representations in image processing,” *IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

- [66] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [67] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, “Compressed sensing mri,” *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, 2008.
- [68] U. Gamper, P. Boesiger, and S. Kozerke, “Compressed sensing in dynamic mri,” *Magnetic resonance in medicine*, vol. 59, no. 2, pp. 365–373, 2008.
- [69] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse mri: The application of compressed sensing for rapid mr imaging,” *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [70] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, “Compressed channel sensing: A new approach to estimating sparse multipath channels,” *IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.
- [71] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, “Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1708–1721, 2010.
- [72] G. Taubock, F. Hlawatsch, D. Eiwen, and H. Rauhut, “Compressive estimation of doubly selective channels in multicarrier systems: Leakage effects and sparsity-enhancing processing,” *IEEE Journal of Selected Topics in Signal Process.*, vol. 4, no. 2, pp. 255–271, 2010.
- [73] P. Zhang, Z. Hu, R. C. Qiu, and B. M. Sadler, “A compressed sensing based ultra-wideband communication system,” in *IEEE ICC*, 2009, pp. 1–5.
- [74] M. Mohtashemi, H. Smith, D. Walburger, F. Sutton, and J. Diggans, “Sparse sensing dna microarray-based biosensor: Is it feasible?” in *Proc. IEEE SAS*, 2010, pp. 127–130.
- [75] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, “Distributed compressed sensing,” 2005.

- [76] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [77] Z. Ben-Haim and Y. C. Eldar, "The cramér-rao bound for sparse estimation," *arXiv preprint arXiv:0905.4378*, 2009.
- [78] W. Chen and I. J. Wassell, "Cost-aware activity scheduling for compressive sleeping wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 9, pp. 2314–2323, 2016.
- [79] L. Huang and M. J. Neely, "Utility optimal scheduling in energy-harvesting networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1117–1130, 2013.
- [80] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.
- [81] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, 2008.
- [82] S. Yang, Y. Tahir, P. Chen, A. Marshall, and J. McCann, "Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing," *IEEE/ACM Trans. Netw.*, 2016.
- [83] M. J. Neely, "Tutorial Slides," <http://www-bcf.usc.edu/~mjneely/>, 2015, [Online; accessed 8-November-2016].
- [84] G. Franklin, "Rational rate [ask the experts]," *IEEE Control Systems Magazine*, vol. 4, no. 27, p. 19, 2007.
- [85] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control system design*. Prentice Hall New Jersey, 2001, vol. 240.
- [86] B. C. Kuo, *Digital control systems*. Oxford University Press, Inc., 1995.
- [87] K.-E. Årzén, "A simple event-based pid controller," in *14th IFAC world congress*, 1999.
- [88] K. J. Åström and B. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems," in *Proc. IEEE CDC*, vol. 2, 2002, pp. 2011–2016.

- [89] X. Wang and M. D. Lemmon, “Event-triggering in distributed networked control systems,” *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 586–601, 2011.
- [90] L. Li, X. Wang, and M. Lemmon, “Stabilizing bit-rate of disturbed event triggered control systems,” *IFAC Proceedings Volumes*, vol. 45, no. 9, pp. 70–75, 2012.
- [91] L. Li, B. Hu, and M. Lemmon, “Resilient event triggered systems with limited communication,” in *Proc. IEEE CDC*, 2012, pp. 6577–6582.
- [92] M. Mazo and P. Tabuada, “On event-triggered and self-triggered control over sensor/actuator networks,” in *IEEE Conf. CDC 2008*, 2008, pp. 435–440.
- [93] J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson, “Self-triggered control over wireless sensor and actuator networks,” in *Proc. IEEE DCOSS*, 2011, pp. 1–9.
- [94] M. Mazo and P. Tabuada, “Decentralized event-triggered control over wireless sensor/actuator networks,” *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2456–2461, 2011.
- [95] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Trans. Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
- [96] D. Michael, X. Wang *et al.*, “Self-triggered feedback control systems with finite-gain l_2 stability,” 2009.
- [97] B. Hu and M. D. Lemmon, “Using channel state feedback to achieve resilience to deep fades in wireless networked control systems,” in *Proc. ACM HiCoNS*, 2013, pp. 41–48.
- [98] L. A. Rossman, “Epanet 2: users manual,” 2000.
- [99] J. Eriksson, F. Österlind, N. Finne, N. Tsiftes, A. Dunkels, T. Voigt, R. Sauter, and P. J. Marrón, “Cooja/mspsim: interoperability testing for wireless sensor networks,” in *Proc. SIMUTools’09*, p. 27.

- [100] J. Lin, S. Sedigh, and A. Miller, “Towards integrated simulation of cyber-physical systems: a case study on intelligent water distribution,” in *IEEE Int. Conf. DASC’09*, 2009, pp. 690–695.
- [101] Adafruit, “Liquid Flow Meter,” <https://www.adafruit.com/products/828>, 2017, online; accessed 25-January-2017.
- [102] Honeywell, “PX2 Series pressure transducers,” <http://sensing.honeywell.com/PX2AN1XX250PSACX-Heavy-Duty-Pressure-Transducers>, 2015, online; accessed 14-September-2016.
- [103] Sparkfun, “Ultrasonic Sensor - HC-SR04,” <https://www.sparkfun.com/products/13959>, 2017, online; accessed 25-January-2017.
- [104] Adafruit, “Waterproof DS18B20 Digital temperature sensor ,” <https://www.adafruit.com/products/381>, 2017, online; accessed 25-January-2017.
- [105] TeraPeak, “Yosoo Dc Water Turbine Generator,” <http://www.terapeak.com/worth/yosoo-dc-water-turbine-generator-water-12v-dc-10w-micro-hydro-water-charging/282112437770/>, 2017, online; accessed 25-January-2017.
- [106] T. Instruments, “INA Family Current Sensor,” <http://www.ti.com/lit/ds/symlink/ina169.pdf>, 2017, online; accessed 25-January-2017.
- [107] A. Muller, “Motorized stepper motor controlled valve,” http://www.akmuller.co.uk/Motorized/10_010_126_SDS.pdf, 2017, online; accessed 25-January-2017.
- [108] M. S. Store, “Jabsco Pump Parmax 4,” <http://www.marinesuperstore.com/marine-pumps/fresh-water-pumps/jabsco-parmax-4-pump-12v>, 2017, online; accessed 25-January-2017.
- [109] Seaflo, “750GPH Bilge Pump,” <http://www.seaflo.com/en-us/product/detail/618.html>, 2017, online; accessed 25-January-2017.
- [110] Arduino, “Arduino shields,” <http://arduino.cc/en/Main/ArduinoShields>, 2015, online; accessed 10-January-2015.

- [111] NXP, “NXP JN516X-EK001 evaluation kit,” <http://www.nxp.com/demoboard/JN516X-EK001.html>, 2015, online; accessed 10-January-2015.
- [112] RaspberryPi, “Raspberry Pi development board,” <http://www.raspberrypi.org>, 2015, online; accessed 10-January-2015.
- [113] Intel, “Intel Edison development board,” <http://www.intel.co.uk/content/www/uk/en/do-it-yourself/edison.html>, 2015, online; accessed 10-January-2015.
- [114] Cla-Val, “Cla-Val Power Generators,” http://www.cla-val.com/electronic-power-generators-c-108_110-l-en.html, 2015.
- [115] S. Kartakis and J. A. McCann, “Real-time edge analytics for cyber physical systems using compression rates,” in *ICAC’14, USENIX*, 2014, pp. 153–159.
- [116] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, “A survey on facilities for experimental IoT research,” *In Proc. IEEE Comm. Mag.*, vol. 49, pp. 58–67, 2011.
- [117] J.-S. Lee, Y.-W. Su, and C.-C. Shen, “A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi,” in *Proc. IEEE IECON*, 2007, pp. 46–51.
- [118] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, “Device-to-device communications with wi-fi direct: overview and experimentation,” *In Proc. IEEE WC*, vol. 20, no. 3, pp. 96–104, 2013.
- [119] M. Onsy, R. Salah, M. Makar, G. Badawi, A. Kenawy, H. H. Halawa, T. K. Refaat, R. M. Daoud, H. H. Amer, H. M. ElSayed *et al.*, “Performance of wsns under the effect of collisions and interference,” *Wireless Sensor Network*, vol. 2014.
- [120] B. E. Bilgin and V. Gungor, “Performance evaluations of zigbee in different smart grid environments,” *In Proc. Computer Networks*, vol. 56, pp. 2196–2205, 2012.
- [121] E. Casilari, J. M. Cano-García, and G. Campos-Garrido, “Modeling of current consumption in 802.15. 4/zigbee sensor motes,” *In Proc. Sensors*, vol. 10, no. 6, pp. 5443–5468, 2010.

- [122] A. El Kouche, A. M. Rashwan, and H. Hassanein, "Energy consumption measurements and reduction of zigbee based wireless sensor networks," in *In Proc. IEEE Globecom*, 2013, pp. 557–562.
- [123] C.-S. Sum, F. Kojima, and H. Harada, "Energy consumption evaluation for power saving mechanisms in recent ieee 802.15. 4 low-rate wireless personal area networks," in *In Proc. IEEE ICC*, 2013.
- [124] M. Zennaro, A. Bagula, D. Gascon, and A. B. Noveleta, "Planning and deploying long distance wireless sensor networks: The integration of simulation and experimentation," in *Ad-Hoc, Mobile and Wireless Networks*, 2010.
- [125] N. Rathod, P. Jain, R. Subramanian, S. Yawalkar, M. Sunkenapally, B. Amrutur, and R. Sundaresan, "Performance analysis of wireless devices for a campus-wide iot network," in *In Proc. IEEE WiOpt*, 2015, pp. 84–89.
- [126] C. Hacks, "Lora experiments, 2016," www.cooking-hacks.com/documentation/tutorials/extreme-range-lora-sx1272-/module-shield-arduino-raspberry-pi-intel-galileo.
- [127] Atmel, "Atmel AVR 8-bit and 32-bit Microcontrollers," <http://www.atmel.com/products/microcontrollers/avr/default.aspx>, 2015, [Online; accessed 9-February-2016].
- [128] J. Kraus and V. Bubla, "Optimal methods for data storage in performance measuring and monitoring devices," in *Proceedings of Electronic Power Engineering Conference*, 2008.
- [129] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proc. ACM SenSys*, 2006, pp. 265–278.
- [130] I. M. Lab, "Filtering Sensor Data with a Kalman Filter," <http://interactive-matter.eu/blog/2009/12/18/filtering-sensor-data-with-a-kalman-filter/>, 2009, [Online; accessed 20-March-2014].
- [131] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proc. IEEE CDC*, 2005, pp. 8179–8184.

- [132] B. Settles, “Active learning literature survey,” *University of Wisconsin, Madison*, vol. 52, pp. 55–66, 2010.
- [133] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, “Wireless sensor networks for environmental monitoring: the sensorscope experience,” in *Proc. IEEE IZS*, 2008, pp. 98–101.
- [134] S. Srirangarajan, M. Allen, A. Preis, M. Iqbal, H. Lim, and A. J. Whittle, “Wavelet-based burst event detection and localization in water distribution systems,” in *Journal of Signal Processing Systems*, 2013, pp. 1–16.
- [135] A. Radder, “On the parabolic equation method for water-wave propagation,” *Journal of fluid mechanics*, vol. 95, no. 01, pp. 159–176, 1979.
- [136] J. P. Shaffer, “Multiple hypothesis testing,” *Annual review of psychology*, vol. 46, no. 1, pp. 561–584, 1995.
- [137] G. Geiger, “State-of-the-art in leak detection and localization,” *Oil Gas European Magazine*, vol. 32, no. 4, p. 193, 2006.
- [138] D. Misiunas, M. Lambert, A. Simpson, and G. Olsson, “Burst detection and location in water distribution networks,” *Water Science and Technology: Water Supply*, vol. 5, no. 3-4, pp. 71–78, 2005.
- [139] W. Yu and J. A. McCann, “Efficient partial-pairs SimRank search for large networks,” *PVLDB*, vol. 8, no. 5, pp. 569–580, 2015. [Online]. Available: <http://www.vldb.org/pvldb/vol8/p569-yu.pdf>
- [140] —, “High quality graph-based similarity search,” in *SIGIR*, 2015, pp. 83–92.
- [141] —, “Co-Simmate: Quick retrieving all pairwise Co-Simrank scores,” in *ACL*, 2015, pp. 327–333.
- [142] NEC/TOKIN, “NEC TOKIN sensors,” http://www.nec-tokin.com/english/product/piezodevaice1/piezo_tech.html, 2016, online; accessed 10-January-2016.

- [143] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [144] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [145] S. Becker, J. Bobin, and E. J. Candès, “Nesta: a fast and accurate first-order method for sparse recovery,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.
- [146] S. Kartakis, G. Tzagkarakis, and J. A. McCann, “Adaptive compressive sensing in smart water networks,” in *2nd International Electronic Conference on Sensors and Applications*, 2015.
- [147] K. Deb, “Multi-objective optimization,” in *Search methodologies*, 2014, pp. 403–449.
- [148] Intel, “Intel Edison Development Board,” <http://www.intel.com/content/www/us/en/do-it-yourself/edison.html>, 2016.
- [149] A. Candelieri, D. Soldi, D. Conti, and F. Archetti, “Analytical leakages localization in water distribution networks through spectral clustering and support vector machines. the icewater approach,” in *Procedia Engineering*, 2014, pp. 1080–1088.
- [150] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proc. ACM ICML*, 2009, pp. 689–696.
- [151] S. Kartakis, W. Yu, R. Akhavan, and J. A. McCann, “Adaptive edge analytics for distributed networked control of water systems,” in *Proc. IEEE IoTDI*, 2016, pp. 72–82.
- [152] W. Lijuan, Z. Hongwei, and J. Hui, “A leak detection method based on epanet and genetic algorithm in water distribution systems,” in *Software Engineering and Knowledge Engineering: Theory and Practice*. Springer, 2012, pp. 459–465.
- [153] S. Kartakis, B. D. Choudhary, A. D. Gluhak, L. Lambrinos, and J. A. McCann, “Demystifying low-power wide-area communications for city iot applications,” in *Proc. ACM WiNTECH*, 2016, pp. 2–8.

- [154] C.-V. Co., “Cla-Val valve controller,” <http://www.cla-val.co.uk/digital-valve-controller-612vdc.html>, 2015.
- [155] S. Kartakis, E. Abraham, and J. A. McCann, “Waterbox: A testbed for monitoring and controlling smart water networks,” in *Proc. ACM CysWater, CPSWeek*, 2015, p. 8.
- [156] M. López-Ibáñez, T. D. Prasad, and B. Paechter, “Ant colony optimization for optimal control of pumps in water distribution networks,” *J. Water Resour. Plann. Manage.*, vol. 70, pp. 337–346, 2008.
- [157] I. Narayanan, A. Vasan, V. Sarangan, and A. Sivasubramaniam, “One meter to find them all: water network leak localization using a single flow meter,” in *Proc. IPSN*, 2014, pp. 47–58.
- [158] S. Pobering and N. Schwesinger, “Power supply for wireless sensor systems,” in *IEEE Sensors*, 2008, pp. 685–688.
- [159] D. Hoffmann, A. Willmann, R. Göpfert, P. Becker, B. Folkmer, and Y. Manoli, “Energy harvesting from fluid flow in water pipelines for smart metering applications,” in *Journal of Physics: Conference Series*, vol. 476, no. 1, 2013, pp. 1–6.
- [160] R. Morais, S. G. Matos, M. A. Fernandes, A. L. Valente, S. F. Soares, P. Ferreira, and M. Reis, “Sun, wind and water flow as energy supply for small stationary data acquisition platforms,” *Computers and electronics in agriculture*, vol. 64, no. 2, pp. 120–132, 2008.
- [161] Y. Gu and T. He, “Bounding communication delay in energy harvesting sensor networks,” in *Proc. IEEE ICDCS*, 2010, pp. 837–847.
- [162] S. Yang, X. Yang, J. A. McCann, T. Zhang, G. Liu, and Z. Liu, “Distributed networking in autonomic solar powered wireless sensor networks,” *IEEE J. Sel. Areas Commun*, vol. 31, no. 12, pp. 750–761, 2013.
- [163] R.-S. Liu, K.-W. Fan, Z. Zheng, and P. Sinha, “Perpetual and fair data collection for environmental energy harvesting sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 947–960, 2011.

- [164] R.-S. Liu, P. Sinha, and C. E. Koksal, "Joint energy management and resource allocation in rechargeable sensor networks," in *Proc. IEEE Infocom*, 2010, pp. 1–9.
- [165] D. J. Wood, S. Lingireddy, P. F. Boulos, B. W. Karney, and D. L. McPherson, "Numerical methods for modeling transient flow in distribution systems," *Journal (American Water Works Association)*, pp. 104–115, 2005.
- [166] D. J. Wood, P. F. Boulos, and S. Lingireddy, *Pressure wave analysis of transient flow in pipe distribution systems*. MWH SOFT, 2005.
- [167] B. Moyer, "Low power, wide area: A survey of longer-range iot wireless protocols," *Electronic Engineering Journal*, 2015.
- [168] Digi, "XBee-PRO 868," <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-pro-868>, 2015, [Online; accessed 5-August-2015].
- [169] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journ. of the Operational Res. society*, vol. 49, no. 3, pp. 237–252, 1998.
- [170] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *et al.*, *Introduction to algorithms*. MIT press Cambridge, 2001, vol. 2.
- [171] W. Heemels, M. Donkers, and A. R. Teel, "Periodic event-triggered control for linear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 847–861, 2013.
- [172] M. Mazo and M. Cao, "Asynchronous decentralized event-triggered control," *Automatica*, vol. 50, no. 12, pp. 3197–3203, 2014.
- [173] B. A. Khashooei, D. J. Antunes, and W. Heemels, "An event-triggered policy for remote sensing and control with performance guarantees," in *Proc. IEEE CDC*, 2015, pp. 4830–4835.
- [174] J. Sandee, P. Visser, and W. Heemels, "Analysis and experimental validation of processor load for event-driven controllers," in *Proc. IEEE CACSD*, 2006, pp. 1879–1884.

- [175] W. Heemels, D. Borgers, V. Dolk, R. Geiselhart, and S. Heijmans, “Constructions of lyapunov functions for large-scale networked control systems with packet-based communication,” in *Proc. ECC*, 2016, pp. 936–938.
- [176] B. V. Eekelen, N. Rao, B. A. Khashooei, D. Antunes, and W. Heemels, “Experimental validation of an event-triggered policy for remote sensing and control with performance guarantees,” in *Proc. EBCCSP*, 2016.
- [177] D. Lehmann and J. Lunze, “Extension and experimental evaluation of an event-based state-feedback approach,” *Control Engineering Practice*, vol. 19, no. 2, pp. 101–112, 2011.
- [178] M. Sigurani, C. Stöcker, L. Grüne, and J. Lunze, “Experimental evaluation of two complementary decentralized event-based control methods,” *Control Engineering Practice*, vol. 35, pp. 22–34, 2015.
- [179] F. Altaf, J. Araújo, A. Hernandez, H. Sandberg, and K. H. Johansson, “Wireless event-triggered controller for a 3d tower crane lab process,” in *Proc. IEEE MED*, 2011, pp. 994–1001.
- [180] S. Trimpe and R. D’Andrea, “An experimental demonstration of a distributed and event-based state estimation algorithm,” *IFAC*, vol. 44, no. 1, pp. 8811–8818, 2011.
- [181] L. Orihuela, P. Millán, C. Vivas, and F. R. Rubio, “Suboptimal distributed control and estimation: application to a four coupled tanks system,” *International Journal of Systems Science*, vol. 47, no. 8, pp. 1755–1771, 2016.
- [182] T. Blevins, M. Nixon, and W. Wojsznis, “Event based control applied to wireless throttling valves,” in *Proc. IEEE EBCCSP*, 2015, pp. 1–6.
- [183] B. Boisseau, S. Durand, J. Martinez-Molina, T. Raharijaona, and N. Marchand, “Attitude control of a gyroscope actuator using event-based discrete-time approach,” in *Proc. IEEE EBCCSP*, 2015, pp. 1–6.

- [184] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, “System architectures, protocols and algorithms for aperiodic wireless control systems,” *IEEE Trans. Ind. Inf.*, vol. 10, no. 1, pp. 175–184, 2014.
- [185] M. Miskowicz and J. Lunze, “Event-based control: Introduction and survey,” in *Event-Based Control and Signal Processing*. CRC Press, 2015, pp. 3–20.
- [186] R. Blind and F. Allgöwer, “On time-triggered and event-based control of integrator systems over a shared communication system,” *Mathematics of Control, Signals, and Systems*, vol. 25, no. 4, pp. 517–557, 2013.
- [187] A. Cervin and T. Henningsson, “Scheduling of event-triggered controllers on a shared network,” in *Proc. IEEE CDC*, 2008, pp. 3601–3606.
- [188] C. Ramesh, H. Sandberg, and K. H. Johansson, “Stability analysis and design of a network of event-based systems,” *arXiv preprint arXiv:1401.5004*, 2014.
- [189] B. Demirel, V. Gupta, D. E. Quevedo, and M. Johansson, “On the trade-off between control performance and communication cost in event-triggered control,” *arXiv preprint arXiv:1501.00892*, 2015.
- [190] C. Ramesh, H. Sandberg, and K. H. Johansson, “Steady state performance analysis of multiple state-based schedulers with csma,” in *Proc. IEEE CDC-ECC*, 2011, pp. 4729–4734.
- [191] R. Blind and F. Allgöwer, “Analysis of networked event-based control with a shared communication medium: Part i—pure aloha,” *Proc. IFAC*, vol. 44, no. 1, pp. 10 092–10 097, 2011.
- [192] M. Vilgelm, M. H. Mamduhi, W. Kellerer, and S. Hirche, “Adaptive decentralized mac for event-triggered networked control systems,” in *Proc. ACM HSCC*, 2016, pp. 165–174.
- [193] R. Obermaisser and W. Lang, “Event-triggered versus time-triggered real-time systems: A comparative study,” *Event-Based Control and Signal Processing*, p. 59, 2015.

- [194] M. Mazo Jr and A. Fu, “Decentralized event-triggered controller implementations,” *Event-Based Control and Signal Processing*, p. 121, 2015.
- [195] F. Calì, M. Conti, and E. Gregori, “Dynamic ieee 802.11: design, modeling and performance evaluation,” in *International Conference on Research in Networking*. Springer, 2000, pp. 786–798.
- [196] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016.
- [197] N. Abramson, “The aloha system: another alternative for computer communications,” in *Proc. ACM Joint Computer Conference*, 1970, pp. 281–285.
- [198] IEEE, “IEEE 802.15.4 Standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs),” <http://www.ieee802.org/15/pub/TG4.html>, 2006, [Online; accessed 12-September-2016].
- [199] H. C. Foundation, “WirelessHART Data Sheet,” <http://www.hartcomm2.org>, 2007, [Online; accessed 12-September-2016].
- [200] A. Fu and M. Mazo, “Periodic asynchronous event-triggered control,” in *Received by CDC*, 2016.
- [201] J. L. Willems, *Stability theory of dynamical systems*. Nelson, 1970.
- [202] E. D. Sontag, “Input to state stability: Basic concepts and results,” in *Nonlinear and optimal control theory*. Springer, 2008, pp. 163–220.
- [203] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.
- [204] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.
- [205] Spectracom, “GPS Clock Synchronization,” <http://spectracom.com/resources/essential-education/gps-clock-synchronization>, 2008, [Online; accessed 12-September-2016].

- [206] TI, “CC430F5137 Communication Module,” <http://www.ti.com/product/CC430F5137>, 2016, [Online; accessed 12-September-2016].
- [207] Intel, “Intel Edison board,” <http://www.intel.com/content/www/us/en/do-it-yourself/edison.html>, 2015, online; accessed 14-September-2016.
- [208] L. Ljung, *System Identification: Theory for the User*, ser. Prentice Hall information and system sciences series, 1999.
- [209] T. M. Walski, D. V. Chase, D. A. Savic, W. Grayman, S. Beckwith, and E. Koelle, “Advanced water distribution modeling and management,” 2003.
- [210] A. S. Morse, “Supervisory control of families of linear set-point controllers-part i. exact matching,” *IEEE Trans. Autom. Control*, vol. 41, no. 10, pp. 1413–1431, 1996.
- [211] Keysight Technologies, “N6705B DC Power Analyzer,” <http://www.keysight.com/en/pd-1842303-pn-N6705B/dc-power-analyzer-modular-600-w-4-slots?nid=-35714.937221.00&cc=GB&lc=eng>, 2016, [Online; accessed 12-October-2016].
- [212] Delta-T Devices, “WET-2 Sensor,” <http://www.delta-t.co.uk/product-display.asp?id=940>, 2016, [Online; accessed 12-October-2016].