

**ENERGY-EFFICIENT MONITORING AND LEAK
LOCALIZATION IN WATER PIPELINES USING WIRELESS
SENSOR NETWORK**

BY

NAJAM US SAQIB

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS AND CONTROL ENGINEERING

May 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **NAJAM US SAQIB** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN SYSTEMS AND CONTROL ENGINEERING**.



Dr. Hesham K. Al-Fares
Department Chairman



Dr. Salam A. Zummo
Dean of Graduate Studies



Dr. Muhammad F. Mysorewala
(Advisor)



Dr. Lahouari Cheded
(Member)

For :



Dr. Moustafa El Shafei
(Member)

13/6/17

Date

©Najam us Saqib

2017

DEDICATION

To my parents and brother.

ACKNOWLEDGEMENTS

First of all, I am grateful to Almighty Allah, the beneficial and merciful, who gave me best of health, knowledge and resilience to accomplish my thesis work. May the salutations and blessings be upon Prophet Muhammad (SAWS).

I would like to seek this opportunity to express my heartfelt gratitude to my thesis advisor Dr. Muhammad Faizan Mysorewala for his continuous support and encouragement. He trained and motivated me and extended his utmost support during my time in the university and especially in my thesis. Furthermore, I would like to thank my thesis committee members, Dr. Lahouari Cheded and Dr. Moustafa El Shafei for their time and effort in evaluation regarding my research work. Also, I am grateful to the Chairman of Systems Engineering department Dr. Hesham K. Al-Fares and other faculty and department members especially Dr. Tariq Nasir, Mr. Shaukat Ali and Mr. Widodo for providing their support and guidance during my time here in graduate studies.

I am also obliged to King Fahd University of Petroleum and Minerals for awarding me full time scholarship to pursue graduate studies as well as the Deanship of Graduate Studies which funded my research.

Moreover, I would like to thank my parents and family for supporting me and providing encouragement with decisions that I make in my life. Finally, I want to extend my sincere thanks to my friends, Saad Waseem, Hassan Sheikh, Aneeq Nasir, Zeeshan Tariq, Taha Nasir, Ahmed Sadeed, Mutsaied Shirazi, Adil Ahmed, Abdul Asad, Owais Ahmed, Abdur Rehman and Abdul Majeed for their generous support and making my tenure at KFUPM a memorable one.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xvii
ABSTRACT (ENGLISH).....	xviii
ABSTRACT (ARABIC)	xx
CHAPTER 1 INTRODUCTION.....	1
1.1. Background	1
1.2. Problem Statement	3
1.3. Thesis Contributions.....	4
1.4. Methodology	6
1.4.1. Assumptions used in Thesis	7
1.4.2. Research Methodology.....	8
1.5. Thesis Organization.....	11
CHAPTER 2 LITERATURE REVIEW.....	13
2.1. Leak Detection and Location in Pipelines	13
2.1.1. Previous Work Done	13
2.1.2. Pressure and Flow Monitoring	14
2.1.3. Acoustic and Vibration Monitoring	17
2.2. Pipeline Monitoring using WSNs.....	19
2.2.1. Previous Work Done	20
2.2.2. Energy-Efficient Techniques	27
2.3. Approaches to Energy Conservation in WSNs	31

2.3.1. Duty Cycling	31
2.3.2. Data Driven Approaches	41
CHAPTER 3 SYSTEM DESCRIPTION.....	51
3.1. Introduction	51
3.1.1. Effect of Leak on Pressure	52
3.1.2. Effect of leak on vibrations	53
3.2. Vibration Modeling	55
3.2.1. Vibrations due to leak	55
3.2.2. Vibrations due to flow	60
3.3. Pipeline Network simulation based on Vibration Model	63
3.3.1. Simulation of Pressure Data	66
3.3.2. Simulation of Vibration Data.....	69
3.4. WSN Node Model.....	77
3.4.1. Energy Equations.....	79
CHAPTER 4 ENERGY EFFICIENCY BASED ALGORITHM FOR PIPELINE	
MONITORING	86
4.1. Duty Cycling.....	87
4.2. Adaptive Sampling Algorithm.....	90
4.3. Adaptive Threshold Algorithm	97
4.4. Data Compression Scheme	101
4.4.1. Algorithm Details	105
4.5. Data Transmission Scheme.....	115
4.6. Solution for Optimal Cost.....	118
CHAPTER 5 RESULTS.....	128
5.1. Adaptive Sampling Algorithm.....	129
5.2. Adaptive Threshold Algorithm	140
5.3. Signal Compression	148
5.4. Effect of Algorithms on Energy Consumption	155
5.5. Effect of Compression on leak localization.....	163

5.6. Cost Analysis of the Scheme.....	169
CHAPTER 6 CONCLUSION AND RECOMMENDATIONS	177
References:	179
Vitae	189

LIST OF TABLES

Table 3-1: Simulation Parameters.....	65
Table 3-2: CC2420 State Transition Times	82
Table 3-3: CC2420 State Currents	83
Table 3-4: Node Components power consumption for different states	83
Table 3-5: Hardware constants for microcontroller	84
Table 5-1: Solutions given by optimization algorithm for different target energies	175

LIST OF FIGURES

Figure 3.1: Pressure Profile in Pipeline for fixed pressure boundary conditions, XL is the leak location	52
Figure 3.2: Pressure profile in the Pipeline for fixed inlet flow and exit pressure, XL is the leak location	52
Figure 3.3: Effect of distance on leak spectrum	54
Figure 3.4: (a) No leak and Leak vibrations for DB3 and (b) No leak and Leak vibrations for position DB11.	55
Figure 3.5: Major Components of the leak vibration model [22].....	56
Figure 3.6: Vibration generation scheme to be used.....	62
Figure 3.7: MATLAB-EPANET Co-simulation scheme.....	63
Figure 3.8: EPANET Model of pipeline, the rectangular position is the leak location and the triangle node is the network exit node with demand of 40 liters per minute.	66
Figure 3.9: Node pressure profile before and after leak for 1mm leak.....	67
Figure 3.10: Node pressure profile before and after leak for 3mm leak.....	68
Figure 3.11: Node pressure profile before and after leak for 5mm leak.....	69
Figure 3.12: Vibrations at 4 different nodes before and after the leak for the 1mm leak case. Node 10 is 23m from the leak, Node 14 is 27m from the leak, Node 7 is 73m from the leak and Node 18 is 77m from the leak.....	70
Figure 3.13: Vibrations at selected nodes for a 3mm leak, it is noticed that at nodes 10 and 14 the leak signature is more prominent now.	71

Figure 3.14: Vibration signature at selected nodes for 5mm leak. The leak is more noticeable in Nodes 10 and 14 but at Nodes 7 and 18 there is no effect of leak.	72
Figure 3.15: Spectrogram of the different leaks at Node 10, 23m from the leak.	73
Figure 3.16: Spectrogram of the different leaks at Node 14, 27m from the leak.	74
Figure 3.17: Spectrogram of the different leaks at Node 7, 73m from the leak.	75
Figure 3.18: Spectrogram of the different leaks at Node 18, 77m from the leak.	76
Figure 3.19: WSN Node main components	77
Figure 4.1: Duty-Cycling Scheme used in the algorithm, dark grey bar indicates sampling with high-energy sensor at high-frequency sampling rate, and light grey bars indicate sampling with low-frequency sampling rate.	88
Figure 4.2: Sampling rate comparison between (a) a conventional sampling scheme and (b) adaptive sampling scheme.	90
Figure 4.3: Adaptive sampling scheme visualization on a frequency variable signal.	91
Figure 4.4: Scheme of calculating the adjusted sampling frequency.....	94
Figure 4.5: Example 3 level Wavelet Packet Decomposition and band selection according to the scheme presented in figure 4.4. The shaded boxes represent sub-bands which are higher than the calculated threshold and are selected for further decomposition.....	95
Figure 4.6: Complete Scheme of the simulation for leak detection using adaptive threshold.....	99
Figure 4.7: (a) Compression and Encoding Scheme for HSVS and (b) for LSVS at WSN Node.	106

Figure 4.8: Decoding procedures for (a) HSVS sampled data and (b) LSVS sampled data.....	106
Figure 4.9: 3 level Wavelet Transform of signal $x[n]$	107
Figure 4.10: Radio Transmission Scheme for the overall simulation.....	116
Figure 4.11: Paths the backtracking algorithm will take to find the most optimal solution. The solid arrows represent forward paths and the dotted arrows shows the backtracking when the cost function is not satisfied for the set parameters.....	126
Figure 5.1: (a) The frequency spectrum and (b) the corresponding sampling frequencies 23 m from 1mm leak.	130
Figure 5.2: (a) The frequency spectrum and (b) the corresponding sampling frequencies 27 m from 1mm leak.	131
Figure 5.3: (a) The frequency spectrum and (b) the corresponding sampling frequencies 73 m from 1mm leak.	132
Figure 5.4: (a) The frequency spectrum and (b) the corresponding sampling frequencies 77 m from 1mm leak.	133
Figure 5.5: (a) The frequency spectrum and (b) the corresponding sampling frequencies 23m from 3mm leak.	134
Figure 5.6: (a) The frequency spectrum and (b) the corresponding sampling frequencies 27m from 3mm leak.	135
Figure 5.7: (a) The frequency spectrum and (b) the corresponding sampling frequencies 73m from 3mm leak.	136

Figure 5.8: (a) The frequency spectrum and (b) the corresponding sampling frequencies 77m from 3mm leak.	137
Figure 5.9: (a) The frequency spectrum and (b) the corresponding sampling frequencies 23m from 5mm leak.	138
Figure 5.10: (a) The frequency spectrum and (b) the corresponding sampling frequencies 27m from 3mm leak.	138
Figure 5.11: (a) The frequency spectrum and (b) the corresponding sampling frequencies 73m from 3mm leak.	139
Figure 5.12: (a) The frequency spectrum and (b) the corresponding sampling frequencies 77m from 3mm leak.	139
Figure 5.13: Median magnitudes of vibration readings different distances from 1mm leak for before and after leak.	141
Figure 5.14: Median magnitudes of vibration readings different distances from 3mm leak for before and after leak.	142
Figure 5.15: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS)and high (HSVS) frequency sensor for Node 10 in 3mm leak.	143
Figure 5.16: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 14 in 3mm leak.	145
Figure 5.17: Leak detection and data request flags from the 4 Nodes in the system.....	146

Figure 5.18: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 10 in 5mm leak.	147
Figure 5.19: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 14 in 5mm leak.	148
Figure 5.20: Compression and distortion for 3mm leak signal at Node 10.	149
Figure 5.21: Compression and distortion for 3mm leak signal at Node 10, 80% threshold, 3 bits SAQ.	150
Figure 5.22: Compression and distortion for 3mm leak signal at Node 10, 65% threshold, 4 bits SAQ.	151
Figure 5.23: (a) Compression at Node 10 with different compression thresholds, (b) similar compression at Node 14 for 3mm leak.	152
Figure 5.24: Effect of varying SAQ from 6 bits to 2 bits.	153
Figure 5.25: (a) Compression at Node 10 with different compression thresholds, (b) similar compression at Node 14 for 5mm leak.	154
Figure 5.26: Energy consumption over time for 4 Nodes under study for 10s duty cycling.	155
Figure 5.27: Energy consumption over time for 4 Nodes under study for 60s sensor duty cycling and 3600s radio duty cycling.	156
Figure 5.28: Energy consumption over time for 4 Nodes under study for a 9 cycle adaptive sampling scheme for 1mm leak.	157

Figure 5.29: Energy consumption over time for 4 Nodes under study for a 9 cycle adaptive sampling scheme for 3mm leak.....	158
Figure 5.30: Energy consumption over time for 4 Nodes under study for a 9 cycle adaptive sampling scheme for 5mm leak.....	159
Figure 5.31: Energy consumption over time for 4 Nodes under study for compression applied using 80% threshold and 4-bit SAQ for 1mm leak.....	160
Figure 5.32: Energy consumption over time for 4 Nodes under study for compression applied using 80% threshold and 4-bit SAQ for 3mm leak.....	161
Figure 5.33: Energy consumption over time for 4 Nodes under study for compression applied using 80% threshold and 4-bit SAQ for 5mm leak.....	162
Figure 5.34: Percentage reduction in energy consumption across all energy conservation schemes for different leak sizes.....	163
Figure 5.35: Cross-correlation result of 1mm leak for Nodes 10 and 14.....	164
Figure 5.36: Cross-correlation result of 3mm leak.....	165
Figure 5.37: Effect of compression on the cross-correlation peak for 3mm leak.....	166
Figure 5.38: Effect of compression on the cross-correlation peak for 5mm leak.....	167
Figure 5.39: Reduction in normalized cross-correlation peak with different packet losses.....	168
Figure 5.40: Cumulative energy comparison between the three adaptive sampling schemes with varying n	170
Figure 5.41: Effect of increasing duty cycle on energy consumption for fixed TL	171
Figure 5.42: Effect of compression on cumulative energy consumption for the nodes under study.....	172

Figure 5.43: Effect of leak detection with different duty cycle times. 173

Figure 5.44: Worst case leak detection of 120s scheme. 174

LIST OF ABBREVIATIONS

WSN	:	Wireless Sensor Network
WPD	:	Wavelet Packet Decomposition
DCT	:	Discrete Cosine Transform
CR	:	Compression Ratio
PRD	:	Percentage Root Mean Difference
SAQ	:	Sub-band Adaptive Quantization
ANN	:	Artificial Neural Network
SVM	:	Support Vector Machine
PSD	:	Power Spectral Density
MEMS	:	Micro Electro Mechanical Systems
IQR	:	Inter Quartile Range
QMF	:	Quadrature Mirror Filter
HSVS	:	High Sensitivity Vibration Sensor
LSVS	:	Low Sensitivity Vibration Sensor

ABSTRACT (ENGLISH)

Full Name : Najam us Saqib

Thesis Title : Energy-efficient Monitoring and Leak Localization in water pipelines
using Wireless Sensor Network

Major Field : Systems Engineering

Date of Degree : May 2017

Leak detection and localization in water pipeline networks is of paramount importance to industry, especially in regions where water is scarce. In this work we present a novel multi-scale approach for leak detection and localization in water pipeline networks utilizing Wireless Sensor Networks (WSNs) by using a multipronged approach to reduce the WSN's energy consumption. The algorithm presented is based on the use of vibration data because this way the leak can be localized without extensive knowledge of the network, and it is easier to acquire at arbitrary points as the vibration sensors are non-invasive. In order to validate the effectiveness of the approach for particular conditions in the pipeline, we simulate the pipeline model using EPANET. Then the vibration model is integrated with it in MATLAB, since EPANET do not include models for vibration measurements. In our multimodal sensing scheme, duty cycling is used to reduce the number of sampling instances. Additionally, our novel scheme hinges on the use of (a) an adaptive sampling scheme wherein the sampling frequency is computed using the wavelet transform, (b) adaptive thresholding to detect the leak and (c) signal compression to reduce the transmission energy. A test-bed of a water pipeline network is considered here, which uses, at various locations in the network, pressure and vibration sensors with different energy ratings and precision levels. The low duty-cycling vibration sensor (HSVS) adjusts the

sampling frequency of the high duty-cycling vibration sensor (LSVS) in the adaptive sampling scheme. Aided by the proposed energy efficiency scheme, the sensor node is able to detect and locate the leak while achieving a significant reduction in energy consumption. Furthermore a cost function is developed and for a given energy consumption requirement a backtracking based solving method is used which gives the most optimum parameters for the given constraints.

ABSTRACT (ARABIC)

ملخص الرسالة

الاسم الكامل: نجم يوس سقيب

عنوان الرسالة: كفاءة الطاقة الرصد و التسريب وموقعه في شبكات أنابيب المياه باستخدام شبكات الاستشعار اللاسلكية

التخصص: هندسة النظم

تاريخ الدرجة العلمية: مايو ٢٠١٧

يعتبر الكشف عن التسرب وتحديد موقعه في شبكات أنابيب المياه أمرا بالغ الأهمية في الصناعة، ولا سيما في المناطق التي تكون فيها المياه شحيحة. في هذا العمل نقدم نهجا جديدا متعدد المقاييس للكشف عن التسرب وموقعه في شبكات أنابيب المياه باستخدام شبكات الاستشعار اللاسلكية باستخدام نهج متعدد الجوانب للحد من استهلاك الطاقة. وتعتمد الخوارزمية المعروضة على استخدام بيانات الاهتزاز لأنه بهذه الطريقة يمكن معرفة مكان التسرب دون الحاجة إلى المام كبير بالشبكة، وانه من الأسهل للحصول على النقاط التعسفية وأجهزه استشعار الاهتزاز غير الغازية. من أجل التحقق من فعاليه النهج لظروف معينه في خط الأنابيب ، ونحن محاكاة نموذج خط الأنابيب باستخدام اباتيت. ثم يتم دمج نموذج الاهتزاز معها في ماتلاب ، لان الملاحم لا تشمل نماذج لقياسات الاهتزاز. وفي نظامنا الخاص بالاستشعار المتعدد الوسائط ، تستخدم الدرجات الواجبة لخفض عدد حالات أخذ العينات. الاضافه إلى ذلك ، فان مخططنا الجديد يعتمد علي استخدام (ا) مخطط لأخذ العينات المتكيفة حيث يحسب تردد أخذ العينات باستخدام تحويل المويجات ، (ب) العتبة التكيفيه للكشف عن التسرب و (ج) ضغط الإشارات للحد من طاقة الإرسال. ويعتبر هنا اختبار سرير لشبكه أنابيب المياه ، والتي تستخدم في مواقع مختلفه في الشبكة ، وأجهزه استشعار الضغط والاهتزاز مع مختلف تصنيفات الطاقة ومستويات الدقة. ويضبط جهاز استشعار الاهتزاز المنخفض الرسوم علي الدرجات الهوائية تواتر أخذ العينات لجهاز استشعار الاهتزاز العالي الرسوم علي الدرجات في نظام أخذ العينات التكيفيه. وبمساعده من المخطط المقترح لكفاءة الطاقة ، فان عقده الاستشعار قادره علي اكتشاف وتحديد مكان التسرب مع تحقيق انخفاض كبير في استهلاك الطاقة. وعلاوة علي ذلك ، يتم تطوير وظيفة التكلفة النسبة لاحتياجات معينه من استهلاك الطاقة وتستخدم طريقه الحل القائم علي التراجع الذي يعطي المعايير المثلي للقيود معينه.

CHAPTER 1

INTRODUCTION

1.1. Background

A wireless sensor network consists of numerous sensor nodes deployed over a geographical area for monitoring and/or recording of a wide variety of physical phenomena including but not limited to temperature, location, vibrations, sound and magnetic field [1]. Typically a sensor node is powered by a battery and has a finite energy source that will run out over time, recharging schemes are applied but power harvesting from such schemes cannot be completely relied upon due to changes in the operating environment of the node. Energy efficiency techniques have to be applied so that energy usage is minimized and node lifetime increases. To this end researchers have done work on multiple ways to reduce energy consumption in WSN nodes and these methods are discussed in detail in the literature review. For example it was found that energy consumption for processing and sensing is much less than energy consumption due to communication [2], but in some cases sensing energy is very high so data acquisition has to be optimized. Similarly in cases where communication energy is high, communication has to be optimized. Using this information designing a node for a specific application becomes a complicated task as the design has to cater for functionality as well as energy efficiency, for this to happen multiple techniques for energy efficiency may be applied [3].

As WSNs can cover a large area for monitoring they can be applied to pipeline monitoring in the oil and gas industry as well as to water pipeline monitoring. Since pipelines can

sometimes go through inhospitable or harsh terrain it is difficult or sometimes impossible to monitor such areas physically. Furthermore since the pipeline structure is scalable and it undergoes changes during its lifetime deploying a wired sensor network on it would be cost prohibitive. Much research has been done on deploying WSNs to pipelines [4], and the advantages that WSN offer are tested extensively in the field experimentally [5] and compared to traditional techniques they are generally more favorable. Some of the more notable works in which WSNs are used for pipeline network monitoring are projects such as PIPENET [6], in which WSN was applied to monitor a pipe network installed in a city for leaks and SWATS [7], which was used for steam flood monitoring and was shown to be more responsive than a normal SCADA based system towards detecting anomalies in pipelines. NAWMS [8] is also notable in which a scheme was developed to solely monitor the flow through vibration readings only using WSNs. Another mention is PipeTECT [9] in which the authors developed an energy conservative scheme for burst monitoring in pipelines using vibration sensors only.

A sensor network is made up of multiple nodes which have processing, sensing and communication capabilities [10]. A sensor node is a module that includes four basic subsystems: a sensing subsystem for data acquisition from the environment, a processing subsystem for local data processing and storage, and a wireless communication subsystem for data transmission. In addition, a power source supplies the energy to the node. This often consists of a battery whose energy is limited. It is usually impossible to charge the battery manually because of harsh terrain. On the other hand, the sensor network should have a lifetime long enough to fulfill the application requirements. A recharging scheme using solar or wind power can also be applied to prolong the lifetime but the battery is

usually included such that the calculated network monitoring lifetime is fulfilled which may be in the order of months.

As such monitoring of pipelines using WSNs is a vast area with improvements that can be applied in many sections of this field which can be improving existing methods or developing new methods for monitoring using the existing devices available. The next subsection highlights the problem statement for the thesis.

1.2. Problem Statement

Conventional pipeline monitoring techniques need to be augmented with sensor-based methods capable of sensing more phenomena related to fluid flow in pipe. These methods need to be energy-efficient and have the sensitivity to report effectively any anomalies such as leaks in the pipeline. This will be used to improve the pipeline monitoring by using less energy than the conventional techniques while maintaining effective leak detection and reporting. The conventional techniques related to flow and pressure monitoring may be used to monitor the operational status of the pipeline and the sensors used to augment the monitoring capabilities may be implemented using a wireless sensor network. For the implementation of sensors using a wireless sensor network, energy-efficient techniques will be needed to make sure that the network lifetime is maximized. To this end an overview of possible methods that may be used to improve network lifetime will be discussed in the literature review. Some of these methods can be used in pipeline monitoring and the methods that will eventually be used in this thesis to reduce energy consumption will be defined in the methodology section.

1.3. Thesis Contributions

The main research objective was to design an algorithm for a system which is based on sensing nodes for pipeline leak detection. The nodes will be connected using a wireless sensor network that implements energy efficiency for both data acquisition and data transmission at the node level.

This main algorithm uses multiple algorithms that have been developed so as to be implemented with the energy efficiency techniques in mind. We have implemented a scheme which reduces the energy consumed by having data driven schemes as well as a duty cycling scheme. These schemes are explained in detail in the methodology section.

The reporting environment gives us a multimodal sensing scheme which enables us to rely on multiple data types from which to choose so as to achieve an energy conservative leak detection solution.

The main contributions of the thesis are highlighted below:

- **Applying a duty cycling scheme for data acquisition.** Data acquisition is usually periodic and when WSNs are deployed we do not need to stream data continuously from the sensor nodes even though it is desirable as it would drain the battery very quickly due to transmissions, WSN nodes are constrained with respect to battery capacity. Duty cycling reduced the amount of time the nodes and radios need to stay awake. This in turn saves energy and increases the energy efficiency.
- **Applying an adaptive sampling scheme.** Usually sensors use less energy when data is being sampled at low rates. And methods are available in literature in which the adaptive schemes scale down the sampling frequency if they sense that the

variation in the sensed phenomena is low and they scale it up if they sense that the temporal variation in the data is high. The main premise of using adaptive sampling is to maximize the amount of information gained using the least amount of samples. Using adaptive sampling would help conserve energy by changing the sampling rate according to the sensed phenomena and reduce energy for sampling instances where variation in data is low for normal monitoring conditions.

- **Implementation of an adaptive threshold based scheme.** As the pipeline is a dynamic environment and vibrations are random and chaotic by nature, it is not necessary that vibrations at all locations are at the same threshold for background noise due to flow and the sensor, an adaptive threshold based scheme will need to be implemented that would work at the node level and set a threshold for normal readings from the vibration sensor, it would then be used to indicate the leak when it happens due to anomalous readings.
- **Implementation of Data Compression algorithm** to reduce the amount of data transmitted. The amount of data can also be reduced by data prediction schemes which model the incoming data to a stochastic model, this reduces the amount of data transmitted by predicting the queried value rather than sampling it. In our case noise is being used and it is random, so data compression will be used to reduce transmitted data instead of prediction. Data compression is desirable because it is used to represent a larger amount of data using fewer number of values and if the data is compressed using a lower amount of energy than is used to transmit the uncompressed data then it results in a net gain in terms of power efficiency.

- **Localizing leaks using a Hierarchical Scheme.** Leaks can be detected and localized using only the High Sensitivity Vibration Sensor (HSVS), but due to energy constraints a hierarchical scheme is developed in which a Low Sensitivity Vibration Sensor (LSVS) is used to monitor for long time periods as it is of lower energy consumption and is used for leak detection. The HSVS is used to further confirm the leak and then localize it as it more sensitive than the LSVS.
- **Optimize the energy consumption using a cost function.** A cost function is developed which is used to optimize the most appropriate monitoring parameters given constraints. This would give us the best possible solution or multiple solutions that can be implemented with respect to the energy efficiency techniques that have been discussed above.

In summary the main contributions of the thesis are application of data driven energy conservation schemes namely the application of an adaptive threshold based scheme, application of an adaptive sampling scheme, implementation of a data compression algorithm and to apply a duty cycling scheme using hierarchical scheme for localizing leaks which saves energy and then using a cross correlation scheme to localize leaks, finally designing of a cost function for optimum calculation of the most energy efficient solution using system parameters. All of these techniques will be applied to a simulation of a pipeline network.

1.4. Methodology

The thesis aims to address the issue of sensing and detecting leaks in a pipeline network using a wireless sensor network with energy constrained sensing nodes, the focus of the thesis will be on applying energy efficiency techniques to the problem. For this to be

successfully implemented, a simulation of a pipeline network has to be performed with a leak present at a known location. Some assumptions need to be defined regarding the pipeline and the sensing nodes that are present. These are highlighted in the section below.

1.4.1. Assumptions used in Thesis

For acquiring data regarding the pipeline conditions at normal operating conditions and under leaks EPANET will be used. Libraries are available which allow the pipeline simulator to integrate with MATLAB to allow for data acquisition. A certain pipeline network with equidistant node placement will be implemented in the simulation. Since EPANET does not have a module or extension in which vibration data can be generated for certain pipeline conditions we will aim to use the data from works presented by different authors to generate vibration data along with the pressure and flow data for a simulated pipeline that we have from EPANET. Vibrations due to flow in a pipe will need to be generated according to models present in literature, since flow in pipelines will be assumed to be laminar these vibrations will not be of high magnitude in nature, we are assuming laminar flow because higher magnitude flow based vibrations negatively affect the cross-correlation result. And leak based vibrations will need to be incorporated into the simulation, this will consist of the leak noise model and the noise propagation model that will be taken from the relevant literature. Using these we can approximate vibrations in a pipeline due to leaks. Using both of these types of vibrations we can approximate vibrations at different points in a pipeline. For these simulations the pipeline material and properties need to be defined and will be assumed for a PVC pipe of a fixed diameter. With the incorporation of vibration data we will have pressure, flow and vibration data for the pipeline which will be used in the algorithms for energy efficiency.

A power consumption model also needs to be implemented for a WSN node and examples are present in literature for consumption models. These models have power consumption profiles for different WSN components like the processor, sensor and the radio. These models can be applied according to the hardware that is usually available for WSN nodes. For sensing purposes we would need vibration and pressure sensors. A power consumption profile will be built according to the components that will be used. It will be determined which components will be used in the nodes.

It is assumed that the nodes do not have any issues in communication and that data transfer is lossless. The multiple sink scheme will be assumed in which the individual sensor nodes will transmit their data to the sink in a single hop and the sink will forward the data to the server. A realistic power consumption profile will need to be assumed for the single hop scheme.

The next section deals with the research methodology that will be used in the thesis.

1.4.2. Research Methodology

This section deals with the research methodology used in the main thesis for energy conservation which will be based on duty-cycling, adaptive threshold, adaptive sampling, data compression and a hierarchical sampling strategy that will be used to localize the leak.

A sensing scheme will need to be implemented which applies duty cycling to save energy by periodically waking the nodes to sample the data. For simplicity a Fully Synchronized Scheme will be used which wakes up all the nodes at the same time because to calculate cross correlation the data from the sensors will need to be time synchronized. The duty

cycling period can be changed according to the required network lifetime while maintaining effective pipeline monitoring status.

After the pipeline has been simulated along with the power consumption model an adaptive threshold based algorithm has to be applied for the detection of leak signatures in the pipeline. This will also improve the reporting model by only reporting events if the vibration thresholds cross a certain specified value. This would help reduce the data transmitted in the network and reduce the transmission energy consumption mostly for events related to leaks. An adaptive threshold based scheme has to be used because the background noise level in a pipeline will vary according to the flow in the pipe and will need to be readjusted according to the conditions in the pipeline. A fixed preset threshold value for vibrations cannot be used in the dynamic environment of a pipeline.

The adaptive threshold scheme will be further augmented with an adaptive sampling algorithm that can change the sampling frequency based on the variation of the sensed phenomena. This would allow us to reduce the sampling frequency during periods of low vibrations and increase it when the vibrations increase or when the algorithm detects a leak has occurred. This would have a twofold effect on decreasing energy consumption as lower sampling frequency uses less energy and signals with less samples will be transmitted in a shorter duration thus reducing communication energy. For the adaptive sampling strategy it would be needed to see which algorithm fits best the conditions that are simulated for the pipeline.

Finally for reporting purposes, the vibration signal will be compressed using a signal compression algorithm. There are many types of signal compression algorithms that are available for example the Discrete Cosine Transform, Wavelet Transform or Discrete

Walsh Hadamard Transforms as well as algorithmic procedures that are able to effectively filter out background noise and give a signal which is both smaller in size than the original signal but has all the relevant information encoded in it. The signal compression stage will be used in the leak localization stage as high frequency sampling will most probably occur at this stage of the process. All of these methods take up a great deal of the processor's time and will therefore be energy-intensive but it will tradeoff between transmitting all of the data which takes up less energy, and instead increasing the processing time to accommodate the compression task which takes less energy. Between the data compression algorithms we will see which ones work best for the system, usually for vibration signals either Discrete Wavelet Transform or Discrete Cosine Transform are used in literature and are effective in compressing the data.

Using any of these methods for data reduction, it is believed that an acceptable decrease in energy costs will occur and the transmitted signal will have enough useful information for an accurate cross-correlation function estimation.

The final step in the complete algorithm for leak localization is the cross correlation algorithm if only vibration signals are to be considered.

In vibration data, the leak from two sensors can be used to localize the leak using cross correlation and it is aimed to find the leak location using the least possible amount of energy. The cross correlation scheme has been previously shown to work effectively to localize leak in simulations and in experiments but it needs to be improved for energy efficiency. Vibration sensors are units with the lowest power consumption amongst pressure, flow and vibration sensors, and it is desirable to use these to calculate the leak location from an energy minimization point of view. In cross-correlation the signal size

can also be reduced by applying a sign function to the vibration signals, this would reduce the number of bits that are required to represent the signal.

Using all of these methods for energy conservation the cost function will need to be computed. This will be based on the amount of energy that the energy conservation algorithms save as well as the accuracy of the leak location. For a given configuration we can then find out the most optimal energy conservation scheme that can be implemented for maximum energy conservation as well as acceptable leak location. This will include terms for adaptive sampling and signal compression as well the leak localization accuracy and promptness in localizing the leak. It will need to be determined which form the cost function will take and how it will be solved.

1.5. Thesis Organization

The report is organized as follows, Section 2 deals with the literature review and in the first part of the literature review examples of leak detection in literature are given, two main methods were found and the methods that the researchers applied are discussed. After this the second part is related to pipeline monitoring using WSNs and some types of sensors used in pipeline monitoring are given as well as some examples in literature where monitoring is done using WSNs. The third part is related to energy efficiency in WSN is given in which different ways to apply energy efficiency techniques is discussed, three main different methods were found in literature that were applicable to pipeline monitoring and they are discussed with examples from literature. In Section 3 the pipeline model is introduced along with the Energy Consumption of the WSN node components. In Section 4 the adaptive sampling scheme that will be used in the simulations is introduced and

developed and the signal compression scheme is developed. It is also related to the cost function with is developed to solve for optimal algorithm parameters. In Section 5 the results are presented and Section 6 concludes the report and future work and recommendations are given.

CHAPTER 2

LITERATURE REVIEW

The literature review is divided into three sections

1. Leak Detection and Location in Pipelines
2. Pipeline Monitoring using WSNs
3. Energy conservation in WSNs

The first section is related to instances in literature which are related to leak detection and location in pipeline, the second part mentions literature in which WSNs have been applied to pipeline monitoring and some papers which have applied energy efficiency techniques. The third section is related to energy efficiency techniques which have been applied to WSNs with respect to duty cycling and data driven approaches.

2.1. Leak Detection and Location in Pipelines

2.1.1. Previous Work Done

Extensive work has been done by researchers with regard to detection of leaks in pipelines, most conventional methods are related to pressure or flow monitoring which can be compared against expected values to detect leaks. Other methods which have been developed relate to the leak noise generated in pipelines, these can be monitored using either acoustic sensors such as hydrophones or using vibration sensors to measure the vibrations induced in the pipes due to the leak noise.

Monitoring pipelines based on models of flow and pressure has been discussed in [11] where accurate leak magnitude is measured.

Work has also been done by researchers to try to find a correlation between flow rate in pipe and the vibrations that are induced in the pipe and this has been both experimentally verified, along with verification from simulations [12], [13]. This can be used to build a model of pipeline vibration due to flows. This concept is further verified in [14] where researchers were able to verify that flow in pipelines create vibrations which are difficult to distinguish from leak vibrations if looked at the raw data, but comparison of the PSD of both were different because leaks have some frequency components of higher energy in some bands. The next subsection discusses works in which pressure and flow based monitoring of pipeline networks.

2.1.2. Pressure and Flow Monitoring

In this subsection some notable methods which were developed to detect leaks in pipelines are reviewed.

In [15] the authors have developed a new mathematical model for pipelines which takes into consideration two phase flow. Pressure and flow rate is measured in the pipes for the experimental data collected. Leaks at various positions was introduced and the pressure and flow rates at the input and output was measured for two scenarios. The scenarios were then simulated using the model and it was observed that the experimental data and the simulation were very close in readings. The error for the flow rate was less than 6% and for the pressure was within 1% of the experimental value. In conclusion the authors have mentioned that the model can be used in leak detection.

In [16] authors have first done a comparison of acoustic detection methods and transient detection methods that are available in literature. It was found that the transient analysis methods discussed were mostly applicable to single pipes and had potential problems when multiple failures were encountered. The authors then further reviewed literature regarding Neural Networks and their application to pipeline monitoring in which quasi static pressure and flow readings were taken to detect and locate leaks in the pipe network. The networks were simulated and the ANNs trained on the leaking and non-leaking network parameters to detect leaks in the testing phase. It was found that ANNs worked very well for large leaks or in low noise environments. A SVM or Support Vector Machine was then trained for use in leak detection as SVMs are better than ANNs and are more general than them. A network was modeled in EPANET for leaking conditions for various emitter coefficients and leak locations. The SVM was then trained on 67% of the data set and tested on the remaining 33%. Differential Pressure and flow rates were used in the SVM to detect and locate leaks. The authors noted that the simulations were not exact replicas of the real network and that training the SVM using variations would entail having a very large data set and that noise has a negative effect on the accuracy so using SVMs are not feasible.

In [17] the authors have presented an adaptive Kalman Filtering approach on the flow and pressure data from a water supply system. They attempt to detect pipeline bursts using residuals. A week of data is used to make a model. Data is taken for 15 minute intervals and it is assumed that consumption is the same every day for the whole week for a given time. Using this information a model is made for each time and the Kalman filter is used to compute the covariances. These will be used later in the sampled data. Tests were done by adding leaks to the system at various points and measuring the flow rates and computing

the residuals. A positive residual indicated a leak and it corresponded well with the simulated leak period. The system was then implemented on some areas and it was found that the system detected leaks well and it matched the customer complaints and the repair work carried out on the leaking lines. The Kalman filter was also applied to pressure data but it did not detect leaks as well as the flow data.

In [18] the authors have proposed a new leak detection method using only pressure data. An algorithm based on the Head loss Ratio of different pressure sensors mounted in the network is proposed. Where data is acquired at a certain time instant from multiple pressure sensors and the head loss ratio is computed by taking the difference between some sensors and then taking the ratio between them, this can be done for three or four sensors. Some assumptions are taken that within the monitored network there are no prior leaks present and there is no pump and the water is from one source only, also the node consumption is a function of some kind that is it is model able. The network is then modeled in EPANET with leak introduced for two scenarios at different time periods. The HLR is then computed for three sensors and it shows that after leaks have been introduced in the network the system is able to properly detect leaks. It was noted that further research can be done as to which nodes to select for leak detection and localization and which Head loss ratio to use.

In [11] a hydraulic model of a pipeline is derived in state space form and a simulation is run in MATLAB for initial steady state conditions. Pressure and Flow are modeled in to the system. The pipeline was divided into two branches from a main pipe. The pipes were segmented into 10m long pieces for the simulation. The state space model was built for all the segments present. A leak is then introduced and the leak magnitude is estimated for the simulation. The pipe segments all have flows estimated for them for known and unknown

leak positions both in the main pipe and in the branches. Simulations were run for small and large leaks and a Kalman Filter was used for state estimation. It was found that the Kalman Filter was able to successfully detect the leak but a constant deviation occurred in the estimated leak value, it took a very long time for the filter to converge too. The system was also able to detect in which location the leak occurred. The next subsection is related to methods which use acoustic principles to detect and localize leaks.

2.1.3. Acoustic and Vibration Monitoring

In [14] vibration is monitored to detect leaks in pipelines. Experimental work is done by the authors and data is collected for multiple scenarios. An algorithm is applied to 13 cases of leaks out of which 10 are successfully detected without filtering the data. The vibration is first monitored in leaking pipes at night then the pipe is repaired and the vibrations monitored for some nights. The motivation by the authors for such a method is that at night time water usage is at the lowest so only vibrations due to leaks will be monitored. 60 samples are taken for each acquisition cycle. Standard Deviation is calculated and the 10 lowest values are chosen and averaged, this is known as the MI or monitoring index. MIE (Monitoring Index Efficiency) is then calculated which is the ratio of the MIs for the leaking and non-leaking nights. The higher the MIE the better the leak detection. In data sets where leak detection was unsuccessful band pass filters were employed after observing the PSD of the signals and leaks were successfully detected for all 13 scenarios.

In [19] experimental acoustic monitoring was done simultaneously using three types of sensors, hydrophones, accelerometers and geophones. Accelerometers were found to be the most appropriate for the testing rig used. Theory [20] is described in which cross correlation between two sensors is used to find the leak location. This can be done by the

time difference which the leak noise takes to reach both sensors, for this method to work the leak has to be located between the two sensors. Filtering effect of pipe and sensors is also discussed as the pipe acts as a low pass filter and attenuates high frequency signals over long distances. Experiment was then done for a strong leak and a weak leak. It was found that the leak localization worked fairly well in the strong leak case for all three sensors but for the weak leak case only the accelerometer managed to get a good localization. Previous work was done on this in [21] and it gives a similar result, although in [20] the background noise prevented the hydrophone from obtaining a solution.

In [22] a virtual pipe test rig using information from previous works [19]–[21] was constructed in which leak noise was generated from a computer using the leak noise model presented in [23] and the pipe propagation model. Using this they were able to generate a leak noise signal using a MATLAB routine and then send it to speakers. The speakers had accelerometers mounted on them and the readings from the accelerometer were used to test different cross correlation algorithms that were similar to the one mentioned in [20].

In [24] hydrophones are used to experimentally detect leaks in pipelines using acoustic data. The method used is invasive and the hydrophone is mounted inside the fluid flow. Since the sensor is inside the fluid there is considerable noise from sources other than leaks for example irregularities in pipe surface and bends. This needs to be addressed for better leak detection. Some theory is discussed regarding noise generated due to leaks and the low frequency components present along with the attenuation and filtering properties of the pipe. The experiment is set up and a large and a small leak are introduced at different pressure settings. The PSD of the signal is then analyzed and it was found that the PSD of leaks are higher in some frequency bands than of normal flow conditions. Different

positions of the sensor were tried and data was collected. It was found that leaks are better heard downstream and at the leak location and that bigger leaks were easier to detect.

In [25] the work in [14] was extended and a similar scheme was carried out for leak detection. It was found that if non-leaking measurements were taken with some flow in the pipeline it became difficult to detect the leak, at higher flow rates it becomes impossible as the PSD of the pipe with high flow and PSD of pipe with leaks become similar. Data Acquisition was done at a 4 KHz rate as frequencies above 1.5 KHz were not observed, it was also observed that as the sensor is located further away from the leak location the pipe begins acting like a low pass filter and the higher frequency components are damped, at more than 10m only frequencies below 500 Hz are visible. Experimental work was carried out with a setup like a mains water distribution in which smaller diameter pipes come off a large diameter pipe, the accelerometers were mounted on the smaller pipes adhesively. A 16 bit ADC was used to measure the readings, Accelerometer of sensitivity of 1V/g was used. It was found that the statistical algorithm used in [16] was sufficient to detect leaks in pipelines if external noise was not present. The authors noted that more work needs to be done to verify a broader set of data.

The section discussed above was related to pipeline monitoring using different types of methods

2.2. Pipeline Monitoring using WSNs

Pipeline Monitoring is problematic using regular sensors because there is associated problem with using wired sensors which are described below.

- (i) The wires in the network may get damaged thus compromising the monitoring capability of the network.
- (ii) Wires can be cut by unauthorized personnel
- (iii) Faulty wires are difficult to locate in long networks, even more so where access is difficult.

WSNs get around most of the problems wired networks have, the major problem that WSNs have is with battery power because since these networks are sometimes deployed in remote areas it may be difficult to replace them, these issues were discussed in the previous section and energy efficiency has come a long way in WSNs. Also if some nodes are non-operational then redundant nodes can be deployed in the network.

There are many instances in literature where pipeline monitoring using WSNs is discussed and some researchers have deployed their networks for leak and pipe monitoring applications [6].

There are some issues regarding pipeline monitoring mostly related to what sensors to use, what protocols to use for routing and how to sample the data, this is discussed in the following sections.

2.2.1. Previous Work Done

This section gives an overview of the more notable works which were done using WSNs to monitor pipeline networks, mostly experimental work was done by the researchers with some work related to simulation.

In [6] PIPENET is introduced which does leak detection in pipelines using a variety of sensors which include pressure, ultrasonic, pH, acoustic and vibration sensors. PIPENET

is a complete water monitoring system with battery powered nodes and nodes which are connected to a power supply. The deployed network consisted only of nodes which were capable of using pressure and ultrasonic sensors. A data streaming algorithm was implemented because nodes did not have enough storage capacity to store and perform operations on data. Data was streamed from the sensors to the server. The sensors themselves had many configurable parameters such as sleep time, acquisition rate, and triggered data acquisition from lower energy sensors, in this case lower energy pressure sensors were used to trigger ultrasonic sensors.

There are several paths of data presented in the paper, the first path is composed of powered sensors because these are continuously streaming data to the server and are monitoring pressure at all times. Since these are used for burst detection of large leaks whose pressure variations travel over large distances it is not required for them to be closely placed so they are installed at pumping stations or reservoirs. The second path consists of battery powered nodes which sample pressure data at intervals and these are sent to the server. Another path exists in which hydrophones are mounted densely and the data is sampled at a high rate periodically and this is also sent to the server for processing.

In the analysis stage pressure data is continuously streamed and at the server a Haar wavelet transform is applied to find leaks, if the coefficients exceed a certain threshold a leak is signaled. For vibration data laboratory testing was done for leak detection using correlation method to find the leak location. The PSD of the sampled data was taken to first classify leaking and non-leaking signatures and then if a leak was present then cross-correlation was applied to the signals to find the leak location. It was found that sensor placement 3m apart can give an average error of 0.2m.

In the deployed network the battery life averaged 7 - 8 weeks for a 6V 12Ah battery, although the authors noted that it could be increased up to a year by reducing the communication and by applying adaptive sampling techniques. It was also noted that time synchronization between nodes was loosely applied and needs to be improved, the authors are working on an improved platform with better processing capabilities and that the current system has drastically improved monitoring of pipelines.

In [8] the authors have proposed NAWMS, a system which uses accelerometers to measure flow rates in pipes which are installed in homes. They use a home water meter and accelerometers mounted on different pipes to measure the flow rates accurately. The accelerations in the pipes are modeled according to the flow rates and then they are calibrated according to an optimization algorithm.

With regard to the flow rate [12] is used as a reference in which the authors through derivations show that the vibrations in the pipe is a direct result of the flow in the pipe. Using this theory the authors of [8] developed a correlation between flow and vibration and using vibration and flow data were able to fit them to an equation which gave flow based on vibration, this was not exactly like the equations described in the derivation due to the difference like sensor properties and pipe structure and un-modeled turbulences. Another issue that arose was that vibration propagated in pipes due to flow in other pipes, this was also modeled after taking data samples and fitting them to an equation. Using these two properties the authors were able to use Geometric Programming and Linear Programming to estimate the coefficients for the equations used to measure vibration coupling between pipes and the coefficients for the equations for the pipe flow and vibrations.

The system on which this method was tested on was for three pipes with two material and two different diameters. The models were tested for flow rates in individual pipes and two pipes, it was discovered at low flow rates the accelerometer did not have enough sensitivity (2g at 10bit ADC) to measure the flow and it improved at higher flow rates. The recalibration algorithm corrected the wrong coefficient values if they were fed into the system. Errors of less than 10% were observed when measuring flow through vibrations. The vibrations due to leaks was not modeled in the system and that could be done for the future so that leakage detection could be incorporated, till now the authors noted that the work only envisioned flow rate monitoring. A wireless network was implemented to read data from the sensors mounted on the pipes which periodically read the sensors and sent them to the central server for computation.

In [9] authors have introduced PipeTECT which is a pipeline leak monitoring system based on accelerometers. The theory used is that there is a temporal change in Maximum Water Head Gradient when there is a burst event in a pipe. The authors propose another method based on measuring acceleration and monitoring the Maximum Pipe Acceleration gradient. The authors noted that a sharp change in pressure in pipe resulted in a sharp change in acceleration at that location. The researchers have proposed a system in which MEMS accelerometers are installed at all the joints in a pipe network so that if a leak occurs in a pipe it can be located by measuring acceleration at the pipe joints. The data is collected in real time using a WSN.

The system uses a WSN with wireless bridging unit named roocas and sensor nodes named gopher. The communication interface between the sensors is a CAN bus and the nodes can be daisy chained together for up to 100 sensors per bus. These nodes are made to be dust

and water proof. The roocas nodes have three communication options using Xbee and Wifi communicators with three different data rates and three different ranges. Time is synchronized in the communication nodes with GPS and WWVB radios. The sensing nodes employ accelerometers in the x, y and z axis and are capable of measuring 2g at 16 bit ADC resolution with filtering. Data is sampled at 150 Hz and is buffered for 1 second before being sent by Wi-fi to server. Nodes wake up for 100ms after every 900ms to send data and perform other tasks. The accelerometer data is time stamped and the acquisition process is initiated by the sink.

The wireless sensor network was then implemented on an experimental pipe network with running water and simulated leaks at different locations. There was water flowing in the network and the accelerometers were set to log data and send to the sink, ambient noise due to water flow was observed. This correlates well to the research done in [9] as along the paths in which water flow was the highest the highest variation in acceleration was observed. After a certain time two valves were opened to simulate leaks and there was a corresponding acceleration spike in the readings after it happened.

The results showed that the closer the nodes are to the leak the higher the acceleration spike was observed, farther away it was weaker. Also nodes which were nearer had encountered the spike first. Using this data the rupture location in the pipeline can be predicted. Also the PSD of the signals before, during and after the rupture was plotted and it was found that the PSD of the data before and during the leak is markedly different. The authors noted that leak detection and location can be predicted by using the acceleration gradient data.

In [26] the authors have proposed a system of pipeline monitoring which uses a WSN for data transmission and ultrasonic sensors for data acquisition. The motivation behind using

such an approach is that it is non-invasive and easier to install as opposed to conventional systems which require working on pipes and delays to operation while installation is carried out. The authors then present a review of current pipeline monitoring schemes most of which have already been discussed.

The authors propose a system using two types of ultrasonic sensors, DUF and TTUF, both have different schemes of measuring flow. DUF depends on the air bubbles present in the pipe flow to measure the velocity and TTUF measure using difference in speed of transmitted signals, TTUF accuracy decreases with increasing number of bubbles in the flow. Both are active sensors and use energy to acquire flow data. Since both these sensors work in different regimes a method to combine the readings of both sensors is developed which would adaptively determine which sensor is giving a more accurate and reliable reading based on the statistics of the acquired signal. Both types of sensors simultaneously acquire data and the best one is chosen.

The sensing system is then simulated on a pipeline which is implemented in Matlab, the sensor is also modeled in Matlab and in the fluid flow different air bubble ratios are simulated. The simulations show that as air bubbles increase the accuracy of TTUF sensor is negatively affected while DUF accuracy increases and vice versa. Also the authors were able to verify their method of heuristically determining which sensor reading to be chosen. The individual sensors were simulated on the pipe and the combined sensors were simulated on the pipe flow and it was found that the trend line matched the combined flow readings with the real flow in the pipeline. The authors also discussed WSNs but did not show results pertaining to data transmission and only showed that their method of ultrasonic sensing works in accurately predicting the flow velocity in the pipeline.

In [7] the authors have proposed SWATS which is a steam flood and water flood pipeline monitoring system. The authors have developed a system which provides real time data acquisition and processing in-network to detect leaks and anomalies like blockage. They say that their system is better than the current SCADA systems implemented because it has got better coverage of the area and is more flexible and has better reporting capability. WSNs have not been used to monitor oilfields and the authors have proposed an algorithm which would consist of communication between nodes locally to solve problems related to anomaly detection. A decision tree algorithm is used which would detect what type of anomaly is occurring and where.

The authors propose SWATS because the current SCADA system is inefficient in terms that anomaly detection takes a long time and the measurements are not fine grained. Multiple sensor nodes and in-network processing can overcome these challenges. The authors note that reliable detection of the problem along with correct identification is important. It also has to be done in a timely manner. For these situations an efficient communication protocol is needed that would give good reporting rates and would conserve energy. Also the data has to be delivered in a timely and reliable manner that would not be affected by interference from other nodes.

The authors use a decision tree based approach to detect problems in the system, for example if it detects an anomaly in the readings at a particular node to correctly identify the issue it would query sensors both upstream and downstream of where it is residing. Spatial and Temporal patterns are made and in the decision tree to classify which problems are happening based on data from neighboring nodes and the time durations. The Steam flood monitoring algorithm is presented in which important steps are highlighted. First

processing is done at a single node which looks at the temporal data to check if an anomaly is occurring, then it uses multi-modal sensing to certify the readings and filters the signal. In the event detection section the node then does a temporal trend analysis in which changes in the value are monitored and their magnitude, then it collaborates with neighboring nodes using a decision tree algorithm to classify the problem. After it has been classified the algorithm then uses a voting scheme to find if neighboring nodes have reached the same result and then the system finds the best upstream node with the same result to localize the problem.

The above section relates to the application of WSN in the monitoring of pipelines, the next section will discuss energy efficient techniques in WSN monitoring in pipelines.

2.2.2. Energy-Efficient Techniques

In this section the literature related to work in which the authors have focused on energy efficiency techniques are mentioned.

In [27] the authors have proposed a mobile sensor node system in which mobile sensor nodes are deployed in the pipe and they are carried by the flow of the transported fluid through the pipe. Multiple sensors may be deployed in the pipeline and they localize themselves by RFID tags that are placed on the pipe in certain locations.

The RFID tags are battery powered and they are spaced equidistant from one another. Multiple node model is employed and the active time of a node is the total transit time through the pipe divided by the number of nodes employed. Handover from one node to another node is based on either of the three methods, location based, time based and interrupt driven. In location based mode the nodes continuously locate themselves using the RFID nodes and wake up at the designated node. In the time based scheme the nodes

wake up after a set time, in this scenario it may be possible that one section of the pipe is not monitored due to varying speeds of individual nodes so time overlap can be employed so that monitoring can be complete. In interrupt driven mode the nodes are connected by wire and when one node is going to sleep it would wake up the neighboring node.

All the three wakeup methods have their associated energy costs and it is discussed in the paper, then a simulation is run using all the three wakeup schemes and energy consumption for the nodes plotted.

It was found that for localization and time based methods if the number of nodes was increasing the energy consumption of the last node was the highest, this was because the node has to be awake to monitor these activities and it slept only after sensing was activated and the sleep location or time passed. The first node consequently had the least energy consumption because it slept immediately after it encountered sleep time and it was activated only from the start. On the other hand in interrupt driven method the node energy consumption was significantly reduced and it was almost same for all the nodes, this was because a node is in deep sleep before interrupt signal was given for wakeup and consequently consumed no energy. The distance between RFID tags also had an effect of reducing the energy consumption when distance between them was increased. Also less RFID tags decreased memory consumption as nodes had to sample less data.

In [28] the authors have introduced CON_NET, an algorithm which uses node power usage profile and an energy harvesting technique using vibration from pipeline flow. This then attempts to find the optimum number of sinks in a network. Or if the number of sinks is defined, then it attempts to find the optimum number of nodes in the network based on the maximum tolerable transmission distance and sampling frequency, on the other hand if the

number of sinks is defined than the maximum possible sampling rate can be calculated. The transmission power in a node is usually described in discrete steps and the authors have used that to determine connectivity and optimized the power usage in the network such that the nodes do not run out of power. The sampling rate can also be modified according to the number of sensor nodes available. The researchers have then run several simulations on a linear pipeline based on different energy harvesting rates and different packet sizes in the transmission and determined the maximum sampling rate that can be tolerated for a different number of sensors. It should also be noted that the focus of the paper was to optimize the power usage so that the sensor node does not run out of energy and is theoretically self-sustaining.

In [29] the researchers have optimized the lifetime of nodes in the pipeline using transmission power. Usually when data is being transmitted in a pipeline the node just before the sink gets the most data passed through it and becomes the critical node in the network that is if the network will fail it will most probably be due to this node failing because of energy issues. It is also noted that most RF modules come with discrete power levels which can transmit up to certain discrete ranges and then simulated a network lifetime for certain power ranges and sensor numbers for a pipeline which showed that at the lowest power setting the maximum lifetime can be obtained with the maximum possible number of sensors. But this did not take into account the increase in traffic due to more nodes, factoring this into the simulation it was found that increasing the number of nodes actually decreased the network lifetime. This is for equal distance placement technique on the pipeline. The authors then did a simulation for equal power placement technique and ran two algorithms, one which increases the transmission power in discrete steps and

another which decreases it. They can compared the performance of both and found that both of them increased the network lifetime over conventional equal distant placement techniques but the High to Low power algorithm gave the best results with a network lifetime of 1.3 times the conventional placement technique for an optimum number of nodes.

In [30] the authors have done leak localization in pipelines using noisy pressure data. A straight pipe section of 11km is taken with pressure sensors mounted at distance of 1km with a total of 11 sensors. A duty cycling scheme to conserve energy is applied when sensing in routing cases and sampling is done once every minute. A model of the node is defined with components that would give us an estimate of energy consumption for various tasks. Leak is detected by finding above normal pressure at a node under steady state conditions. In this scenario the first sensor downstream would detect a leak. After the leak is detected the nodes do into high rate sensing mode to sample pressure data and to locate the leak. Leak is located by applying least squares to sensors before the leak and after the leak to find the point of intersection and thus the leak. At this point leak localization can be done by two methods, either use all the sensors of the network or use the minimum amount of sensors i.e 4 to locate the leak, 2 on each side of the leak. It was found that using only 4 sensors gave an appreciable amount of reduction in energy costs while still giving relatively accurate leak localization along with the leak magnitude.

In [31] the authors have introduced an adaptive thresholding algorithm that improves upon the work done in Pipetect [9] to detect leaks in pipelines. Various methods that may be used to improve the energy consumption profile of the existing system are discussed like adaptive sampling or computing the frequency spectrum of the leak signature before

reporting the event or algorithmic approaches that are used by other authors. Since the pipeline is a dynamic environment with several external factors to induce vibrations apart from the leak an adaptive thresholding algorithm with statistical analysis is used along with a cascaded wakeup hierarchy when the node detects an event to transmit the event parameters. The algorithm was then applied to an experimental system and was found to be working correctly with up to 80% reduction in energy costs compared to the previous system [9].

This section was related to the literature review of pipeline monitoring using conventional techniques as well as WSN schemes, also examples from literature were discussed in which energy efficiency of WSN based systems was the main focus of the work, the next section is related to the energy efficiency techniques that are normally used in WSNs.

2.3. Approaches to Energy Conservation in WSNs

The main focus of this section is discussing energy efficiency methods that are used in WSNs. These are general methods and can be applied to most systems, although not all methods are applicable everywhere, the next subsection is related to duty cycling which is the most used method in WSNs.

2.3.1. Duty Cycling

When we are talking about duty cycling in literature [10] we find that there are mainly two subclasses. Duty Cycling based on Topology Control and Duty Cycling through Power Management.

In Topology Control the main idea is to exploit the network redundancy to prolong the network lifetime. This increase in the lifetime is based on how many redundant nodes are

there with respect to transmitting and data acquisition. In literature a network lifetime increase of 2 to 3 times is observed as opposed to a network with all nodes on at all times.

Power Management based Duty Cycling refers to the switching of radio between sleep and wakeup times according to the network requirements and activities. Both of these Duty Cycling methods complement each other when used in tandem.

Within Power Management the Sleep/Wakeup cycling can be implemented in two layers. It can be embedded in the MAC layer itself or they can be run on top of the MAC layer in the application itself. Running the Sleep/Wakeup Protocol in the application layer has the advantage that it can be run on any MAC layer and is very flexible. On the other hand MAC layer Protocols allow us to optimize the Medium Access at the physical level itself.

2.3.1.1. Topology Control Protocols

Topology control methods are used when dense sensor fields are employed to ensure good coverage of an area. The network is usually deployed at random. The high density ensures that redundancy is ensured in case of multiple node failures. The network lifetime can then be increased based on the minimum number of sensor nodes the protocol decides to fulfill the task at hand. This can be decided based on protocols which cover two broad categories.

(i) Location Driven Protocols: These decide which nodes to use based on their location.

(ii) Connectivity Driven Protocols: These decide which nodes to use to complete the connection path or to completely cover the area under surveillance. A brief summary of both these methods are given below.

2.3.1.1.1. Location Driven Protocols

Geographic Random Forwarding or GeRaF [32] is a protocol in which nodes switch within a given duty cycle to listen to data transmissions and participate in routing. Forwarding is done in priority regions which is based on the nearness of the active nodes to the target. As soon as a node receives a packet it forwards it in a packet which contains the target location as well as the location of the forwarding node. Since the sleep wake up cycles of the nodes are random the chance that multiple nodes are awake when forwarding is taking place is present. In this scenario if multiple nodes within the same priority region forward a packet collision occurs. This collision is then dealt with a back-off scheme which retries transmissions for a certain amount of time. There can also a scenario in which all nodes in a certain priority region are sleeping. This can be handled by trying forwarding through nodes in a lower priority zone. This routing method only needs the target location to completely route the data.

Another protocol is the Geographic Adaptive Fidelity [33] which breaks down the sensor field into grids. All nodes in one grid will be able to communicate with nodes in the adjacent grid. Sleep times are decided for the nodes in one grid after they have finished the discovery procedure. Periodic re-election of a leader is also done based on the residual energy of the node. This method ensures that network lifetime will increase based on node density. Only one node in a square is available at any one time to take care of routing, nodes will periodically wake up to check if any other node is available for routing, if other nodes are available it would go back to sleep. The main problem with such a method is that underutilization of the network resources take place with respect to communication

distance, less than half of the optimum distance for transmission is covered due to the grid structure of the method.

2.3.1.1.2. Connectivity Driven Approaches

SPAN [34] is a protocol which adaptively elects coordinators in a network environment. Coordinators are nodes which stay awake to ensure that the routing path is complete while other nodes sleep and function as normal, the other nodes also wake up to check if they need to become a coordinator. If two nodes of a node that is not a coordinator cannot reach each other than that node becomes a coordinator. In the process of selecting a coordinator it may happen that multiple nodes decide to become coordinator, because of this possibility the protocol provides for a random back-off delay before announcement. This random time is a function of residual energy present and the number of neighbors it can provide connectivity to. A coordinator withdraws if all its neighboring nodes can reach each other directly or through another coordinator, this way energy is saved as redundant paths are reduced. The Span algorithm needs information regarding neighbor and connectivity in selection of a coordinator node.

ASCENT [35] is another protocol based on connectivity. It is based on the concept of active and passive nodes in the network. Passive nodes listen to transmissions and active nodes take part in routing of the transmissions till their energy is depleted. If packet loss becomes high enough at active nodes then they send out a message to passive nodes to take part in the transmission of the data packets. When active nodes join the network they broadcast their presence in a neighbor announcement message. The number of active nodes in the network are increased till the packet loss rate goes below a certain threshold. This method of maintaining connectivity is independent of the routing protocols.

Another connectivity based protocol is NAPS [36]. In this the node wakes up after a time period for a certain amount of time and then broadcasts a HELLO message. It then waits for a certain number of replies. If it receives the replies within its wake up period it goes back to sleep otherwise it remains awake till the threshold of number of replies has been reached. During this period it takes part in all the routing of the data.

2.3.1.2. Sleep/Wakeup Protocols

There are three main types of Sleep/Wakeup Protocols, these are independent of the MAC layer. These are mostly related to the radio power cycling and are listed below:

(i) On-Demand Schemes

(ii) Scheduled Rendezvous Schemes

(iii) Asynchronous Schemes

2.3.1.2.1. On-Demand Schemes

In this scheme the node wakes up when it is about to receive a packet from another node. The main idea is that nodes need only communicate when it is needed such as when an event occurs otherwise nodes are in their monitoring state. Usually two radios are used in this scheme, one is low power and is on all the time and the other is high power and only turns on when is required. The drawbacks in such a configuration is that the low power radio has got significantly lesser range than the high power radio and consequently that radio is underutilized range wise, also the system node has two radios increasing its cost.

STEM [37] uses two radios of same power for wake up signal and data transmission. Since wake up radio does not have low power it employs an asynchronous wakeup scheme in which it periodically wakes up and checks for beacon. The beacon is sent periodically by

a node which wishes to transmit data. Beacon is transmitted for a maximum time until target node responds. If collision occurs on the wakeup channel no acknowledgement is sent on that channel.

Another variant of STEM known as STEM-T [38] the authors use a wakeup tone to wake all the neighbors of the initiator node. STEM has a problem in which path setup latency is very high. The active time for a node to detect a beacon is more than double the beacon transmission time plus the acknowledgment time. Combine this single hop delay over long networks and the transmission latency increases. Also the time to remain active would take up a lot of energy.

PTW [39] Pipelined Tone Wakeup is a protocol which aims to achieve a tradeoff between energy and latency. This uses two radios like STEM but the sender now sends a wakeup tone that is long enough for the periodically waking receivers to be detected.

Another method uses low power radio receiver in continuous on state [40] and when it receives the signal it turns on the data radio. The main problem with this approach is that since the wakeup radio is on at all times and power consumption has to be minimized the low power radio has a lesser range than the high power radio. It should be noted that the high power radio is underutilized as its transmission range is greater.

Another method investigated is the radio triggered wakeup scheme [41]. In this the radio energy from the signal is used to turn on the data radio. This has the plus point of not using any standby radio which consumes energy to listen for the signal. The main issue in such a scheme is the lack of range.

2.3.1.2.2. Scheduled Rendezvous Schemes

In this type of scheme all the neighboring nodes wake up at the same time. Typically nodes wakeup after a certain sleep time but the main advantage is that it is certain that neighboring nodes will be awake for transmission purposes. The main issue in this case is clock synchronization in that all nodes have a clock which is the same and does not drift over time.

In Fully Synchronized Pattern [42] all nodes of a network wakeup at the same time periodically. Many systems use this, TinyDB [43], TASK [44], S-MAC [45], T-MAC [46]. Time to wake up and active time are fixed for a certain network. This scheme can be improved by turning the radio after a certain time if it does not detect any network activity. The drawbacks to this approach are that when nodes try to transmit simultaneously, collisions will occur and the network cannot scale up from the size of the original network.

Staggered Wakeup Pattern [42] nodes at different levels in the data gathering formation wake up at different times but in a manner that they are staggered and at least one neighboring node which will carry data to the sink is awake. Advantageous because the number of collisions is lower as only a few nodes are awake a certain point in time. Cons are the same as Fully Sync Pattern in as all the nodes waking up at the same level will have high probability of collisions, also the wakeup and active times are fixed for a network.

Flexible Power Scheduling [47] has a slotted approach for the time in which the node has to transmit data. The slot time is fairly large as to stop issues with synchronization and nodes can reserve slots in advance to overcome the inflexibility of fixed slot positions.

Other variations of the staggered scheme are also in literature like forward staggered [42] and two staggered pattern [42] and even and odd pattern [48].

2.3.1.2.3. Asynchronous Schemes

Asynchronous schemes allow nodes to wake up independently of each other while making sure that neighbors always have overlapped active periods for communication.

Asynchronous schemes were first introduced in [49] for IEEE 802.11 standard.

In [50] Asynchronous Wakeup Protocol is implemented in a (7,3,1) configuration. 7 Time slots are allocated to a node out of which it has to be active for any 3 of them. The active slots are set such that only two schedules are overlapping for one slot. This ensures that whenever a node is active it has at least two neighbors to which it can communicate data.

Random Asynchronous Wakeup (RAW) [51] takes advantage of dense networks and that several paths may exist between sources and sink. Random Wakeup of nodes occurs as they forward the data to the set of active neighbors. If the forwarding candidate set is large enough there is high probability that the packet will be forwarded otherwise if network is sparse then this gets difficult as very less nodes might have probability to wake up. The discovery of nodes for receiving data is done much like either STEM-B or PTW which were discussed previously.

2.3.1.3. MAC Protocols based on Low Duty Cycle

The following section deals with MAC protocols in literature [52] that are based on low duty cycle. Usually MAC protocols deal with channel access methods but for power management they implement low duty cycle.

There are three main types of MAC Protocols available in literature:

(i) Time Division Multiple Access

(ii) Contention Based

(iii) Hybrid Protocols which employ a mix of the two

2.3.1.3.1. TDMA Based MAC Protocols

In these type of protocols [53] time is divided into frames which are further divided into slots. These slots are assigned to nodes according to a scheduling algorithm. Usually there are cluster heads in the network which are assigned this job. One common example of this protocol is Bluetooth [54].

TRAMA [55] divides time into two periods, a contention based random access period and a scheduled access period. In the contention based period nodes find two hop neighbor information to establish collision free schedules. Then they start an election procedure based on which slots which nodes will occupy based on this information. After this has been done a schedule is created on which nodes can communicate in a collision free manner.

FLAMA [56] is derived from TRAMA and is optimized for periodic reporting applications. As the data transmission path in such applications is not random it sets up flows for the data and uses a method in which data is reported only when it is requested by the sink.

LMAC [57] aims to reduce protocol overhead for energy efficiency. The slot assignment is based on the current occupied slot and random assignment among the free slots. The frame length is fixed and has to be specified before deployment.

2.3.1.3.2. Contention Based MAC Protocols

In B-MAC [58] nodes wakeup after a duration known as the check interval and remain awake for a wakeup time. This is asynchronous. Nodes then listen for any ongoing transmission. If it detects transmission then it simply receives the packet. The packet is made up of a preamble and the payload, the preamble is long enough to be more than the check interval of the node so that if transmission is going on than any node waking up would be able to detect channel activity.

In S-MAC [45] nodes exchange sync packets to coordinate their duty cycling patterns. Nodes can then make their own schedule or follow of their neighbor, nodes following the same schedule make a cluster. A node can follow two schedules if they do not overlap and can thus form a bridge between two clusters. Two time periods for channel access, listen period to exchange sync and control packets and remaining period to exchange data. Nodes not concerned with data transfer can sleep but if overhearing a transmission wakeup for short time at end of it so latency of receiving next packet if it is intended for the sleeping node is very low.

D-MAC [59] is implemented in networks where a tree organization is done for the nodes. The nodes schedules are staggered according to their position in the tree. Nodes having more packets request for additional slots from the nodes above them. This way the active period is adaptively adjusted.

IEEE 802.15.4 PAN [60] has got coordinators which manage nodes and if need be more coordinators which manage subsets of the nodes present in the network. In beacon enabled mode a special frame is periodically generated by the coordinators. These consist of an active period which has a contention access period controlled by a CSMA/CA algorithm

and a collision free period in which nodes are given time slots to communicate. The inactive period is for the nodes to sleep. In non-beacon enabled mode the nodes communicate using a CSMA/CA algorithm. Energy conservation is then applied in the application layer.

2.3.1.3.3. Hybrid MAC Protocols

PTDMA [61] is a scheme in which time is slotted and nodes are either owners or non-owners of the time slot and the network switches between TDMA and CSMA based on the level of contention in the network. This was formulated for a one-hop wireless network so does not take into account topology changes and other issues which affect wireless networks.

In Z-MAC [62] nodes populate a list of its two hop neighbors then apply a slot assignment algorithm so that any two nodes in neighborhood don't have the same slot. This guarantees that neighbors in the two hop neighborhood would be collision free. Z-MAC allows the nodes to maintain their own time frame based on the transmissions of its neighbors. Nodes are either in LCL or HCL. LCL is low contention level in which any node can compete for the slot but in HCL the owner of the slot and only its one-hop neighbor is allowed to compete for it. HCL mode is activated by an ECN or Explicit Contention Notification, this is sent out by nodes when they experience high contention. This section was related to the duty cycling schemes in WSNs, the next section is related to data driven approaches for energy conservation in WSNs.

2.3.2. Data Driven Approaches

Data driven approaches can be broken down into two main categories.

(i) Data reduction which aims to reduce the data transmitted in a network through various methods

(ii) Energy Efficient Data Acquisition which aims to conserve energy by using various methods to reduce the energy during the data sampling phase.

2.3.2.1. Data Reduction

Data Reduction aims at reducing the amount of data that can be transmitted. There are basically three main methods through which this can be achieved.

(i) In network processing

(ii) Data Compression

(iii) Data Prediction

2.3.2.1.1. In network Processing

In network processing refers to the various techniques like network coding and data aggregation which is applied to the data in the wireless sensor network. Network coding is a method in which instead of processing packets and sending them they apply a function on them beforehand. This way the nodes in the network will have to apply a function on them to decode them instead of receiving multiple processed packets from different nodes. The number of transmissions is reduced as the same data is being sent to all nodes and it only needs to be sent once. Adapcode [63] is a data dissemination protocol in which it forwards one packet per N packets received. This way flooding the network with packets is mitigated. This may not work for sparse networks as the number of nodes will already be low. So nodes can recover data by sending negative acknowledgement to recover data. Data aggregation is usually application specific but in wireless sensor networks it may be

used to reduce the amount of traffic. This way latency in a network can be improved. Data aggregation can be done by averaging at one node or by combining packets coming from higher levels in the network at one node. The disadvantage of such a technique is that the original data may be lost and may not be recoverable depending on the network. In [64], [65] data aggregation techniques are surveyed in detail.

2.3.2.1.2. Data Compression

Data Compression is the technique in which the number of bits used to represent a data is reduced. This increases energy efficiency as the time taken to transmit a packet is reduced. This ultimately reduces latency also. Data can be encoded at the source node and then decoded at the sink node for example, these techniques are not application specific to wireless sensor networks and can be generally applied. Also normal data compression techniques are resource and energy intensive and controllers at the node level are usually not equipped with that kind of hardware and specific techniques have been surveyed in [66], [67] for application in wireless sensor networks.

There are many algorithms which can be applied to compressing data in wireless sensor networks which have been applied to vibration signals in particular, [68] uses the discrete cosine transform to compress the incoming data for transmission through the wireless sensor network, it is compared against LEC [69] which is also an algorithm for compression of data in the WSN. There are other types of compression methods which are also available for example the lapped orthogonal transform [70] and the discrete Walsh transform [71] which can also be applied to compress signals in WSNs. The discrete wavelet transform [72] is also extensively used in signal compression and [73] has applied to compressing speech signals using a DSP based microcontroller platform.

2.3.2.1.3. Data Prediction

Data Prediction consists of building a model of the sensed value within certain error bounds. If the accuracy bound is not exceeded queries can be answered at the sink without any communication. On the other hand if the model has changed then communication has to take place to update the model. Normally prediction results in lower communication overhead.

There are 3 basic data prediction schemes that are discussed in literature:

(i) Stochastic Schemes

(ii) Time Series Forecasting

(iii) Algorithmic Schemes

Stochastic Schemes:

Stochastic methods are used for a probabilistic model of a random process.

In [74] the authors have defined a procedure in which there are a number of models of the process at each node and at the sink. The model is based on a trained probability density function and when it is not considered valid anymore it is recomputed as the source node updates the sink node. In [75] a similar method has been used but it uses a Kalman filter for prediction.

The work in [74] is extended in [76] where a Dynamic Probabilistic Model is used to show the user the hidden states of the system, this is possible through using a model of the system and using the sensed state to compute the other states which are hidden or not measurable. Particle filtering approach is taken for estimation of the sampled data.

Time Series Forecasting:

Time Series forecasting of any process can be done using the AR, MA or ARMA models. These simple models can be used with acceptable accuracy rates in wireless sensor networks.

PAQ [77] uses a low order AR model. It first computes the model at the nodes using the sampled data and then sends only the coefficients of the model to the sink. If the values of the model and readings are within the user specified error bound the model is considered valid, otherwise if many consecutive readings start falling outside the range then the model is relearned and the values sent to the sink again. Clustering also takes place when multiple nodes have similar models, this reduces communication as nodes within the cluster have the same model and communication to sink is handled by the cluster head.

SAF [78] improves upon the previous algorithm by including a trend element in the forecasting. Also outliers are accommodated by filtering and enlarging the model if possible. Even if then a stationary model is not possible then the model is rebuilt at the nodes at resent to the sink.

In [79] multiple models are computed at the nodes. All are running in parallel. Models with the least predicted error are chosen over time. Also considered is the cost associated in updating a model. The more complex a model the higher the update cost but prediction may be better. Every time a new model is chosen at the nodes the model is updated at the sink. Over time a racing mechanism is employed to discard the poorly performing models. This results in selected models which best portray the process.

Algorithmic approaches:

Algorithmic approaches are more application dependent and vary according to the needs of the network and the approach used by the authors of the paper, one approach in one application may not be applicable in another scenario.

PREMON [80] uses the idea of nodes as pixels and attempts to use spatial correlation. As the sensed data varies over time the authors present it like a movie where the sensors are the pixels in the image. Sensors send their initial readings to the sink where it computes the model and sends it back to the nodes. The nodes then use the model to predict the data. If it is within the error bound the data is not sent to the sink. Periodically the model is recomputed again at the sink to replace the older model.

In [81] a buddy protocol is presented in which nodes form clusters with their neighbors. In each cluster there is a cluster head responsible for query processing and monitoring. This is periodically changed to conserve node energy. Communication between nodes and cluster head can be default mode in which it sends the raw data or like PREMON where model is computed. In systems where the sensed variable is changing rapidly default method is more suitable and in systems where data is smooth and stable PREMON method is more suitable.

In [82] authors propose a system in which the sensor node sends the upper and lower bound of its readings to the sink. This is constantly updated if values fall outside these bounds. When sink receives query from user it checks the bounds, if it is within the acceptable range then the cached range is given, otherwise the sink requests the real value from the nodes and updates its bounds. Changing these ranges affects power consumption as it limits

communication between nodes. An optimal setting can be given for a scenario to save power.

2.3.2.2. Energy Efficient Data Acquisition

These techniques are born out of the belief that sensing is relevant from the energy consumption standpoint and that it may be greater than the energy for computation or for that matter communication. This can be due to some factors that are listed below:

(i) Long Acquisition Time: The data acquisition time of sensors may be in the order of seconds, this will accumulate over time and even if the sensor uses low energy the total energy for one cycle of data acquisition may be high

(ii) Power Intensive Sensors: Some sensors for some applications require high power, an example is the CCD array in a digital camera.

(iii) Power Intensive ADCs: Some sensors [83], [84] require high precision, for this level of precision the ADCs are usually of high resolution and can provide data at a high rate, these requirements call for more power than is usually required.

(iv) Active sensors: Some sensors require energy to be transmitted to sense a phenomena, examples are ultrasonic rangefinders and sonar. These require energy even if the sensing part uses low energy the active signal requires a high amount of power.

In these cases, reducing the communications or even reducing data may not be feasible as sensing is more expensive. Methods have been researched in literature [85] to reduce sensing and there are three basic fields in which they can be broken down:

(i) Adaptive Sampling

(ii) Hierarchical Sampling

(iii) Model based Active Sampling

2.3.2.2.1. Adaptive Sampling

Adaptive sampling can reduce the number of unnecessary samples for a given time. For example in [86] temporal analysis of data is done dynamically to choose the correct sampling rate that can reconstruct the original signal according to the Nyquist criteria. The algorithm is executed at the sink and the updated sampling rate is sent to the nodes. In [87] a Kalman filter is used for similar purposes.

In [88] the concept of a dense sensor field and spatial correlation is used. Sensor field is divided into blocks where variations of the sensed phenomena is low. Then in preview phase only a few sensors in a block are activated to sense the data. Each block is managed by a cluster head. If the variations in data is high and spatial correlation is low then refinement step is taken in which more sensors in the clusters are activated. If the variation in data is low only the preview phase is enough to get sufficient information regarding the data.

Spatial Correlation is also used in CC-MAC [89] in which an Iterative Node Selection algorithm based on the calculation of the correlation radius based on maximum distortion that can be tolerated residing at the sink . The radius is transmitted to other nodes in the network. Two other MAC protocols E-MAC and N-MAC are used in the transmission. E-MAC reduces the amount of redundant data in the system and N-MAC manages transmission of older packets in the network giving them higher priority against newer

packets. Also initially all nodes contend for access of network, the winning node becomes the representative of nodes in the radius.

Floodnet [90] uses an application specific adaptive sampling system in which data from critical nodes with higher sampling rates are given higher importance and nodes with lower sampling rates are used to route the data. The idea being that nodes less loaded with sampling tasks would have more energy to spare for routing purposes, also if nodes with higher sampling rates are not used to route data then they would last longer.

2.3.2.2.2. Hierarchical Sampling

Hierarchical Sampling is based on the idea that some sensors providing coarse information use less energy and other sensors providing information with high resolution use a higher amount of energy, using this information we can use the lower power sensor for event detection and then use the higher power sensor for tracking or purposes requiring better information of the sensed phenomena [91], [92].

In [93] for structural health monitoring the sensors are split into two types, u-nodes containing accelerometers and m-nodes containing strain gauges. u-nodes are low power sensors and are used for continuous monitoring, if they find an anomaly and it is verified by cross checking with the neighbors then the m-nodes are activated and finer information is obtained.

In [94] static sensors report to the sink, if there is an anomaly a mobile sensor is dispatched to the area with the anomaly for better information gathering and it reports back to the sink.

2.3.2.2.3. Model Based Active Sampling

This approach is similar to data prediction. Data prediction methods periodically tune the model, this reduces communications as it does not have to be at all instants but it does not reduce sampling. This approach reduces the number of samples itself.

In BBQ [95] a probabilistic model and planner are at the sink. The model has PDFs which are flexible enough to cater for the spatial and temporal correlations. The model is updated by combining the PDFs with the incoming samples. The model is constructed at the sink by the initial values sent by the sensors. The model is updated with answers received to queries. The planner decides the when and what of queries, what values it requires and when does it need to query, also what sensors it needs to use.

In ASAP [96] clusters and sub-clusters are formed. The cluster heads are responsible for the cluster and within a sub-cluster only one node is the sampler. The cluster head forms the sub-clusters based on initial data received and selects samplers from them. Clusters and sub-clusters are periodically recomputed. The probabilistic models are computed for each sub-cluster and sent to the sink. The model is made at the sub-cluster so data is not needed to the sink all the time. Communication energy for the whole network reduced.

In USAC [97] the sampling frequency is calculated at each node not at the sink. A linear regression model is used for forecasting. If the prediction is not in the confidence interval the sampling frequency is increased to f_{max} . This increases accuracy when updating the model. On the other hand if values lie in the confidence interval then sampling frequency is decreased till it reaches f_{min} . Also the authors propose a routing protocol in which paths are taken along which data is sampled at a lower frequency.

CHAPTER 3

SYSTEM DESCRIPTION

To implement the proposed WSN based energy conservative scheme for pipeline monitoring a method for simulating a pipeline and generation of data has to be devised. Also the node model will be introduced on which the scheme will be applied. This would allow us to best estimate the energy efficiency that we are achieving because of the energy conservation methods applied. First we will introduce how the leak effects the pressure and vibration in the network.

3.1. Introduction

Leak localization is a major problem in the water industry and, as such, remains an active area of research, with various methods [98] having been developed recently to localize leaks in water pipeline networks using pressure and flow data. The success of these methods is due to their use of artificial intelligence techniques and the recently available extensive computing power to localize leaks in pipe networks using pipe network models. In addition to pressure-based leak localization, the past years have also witnessed the use of acoustic-based leak localization methods [99], [100] to locate underground leaks which were difficult or impossible to locate using conventional methods. For acoustic localization the leak noise transmitted acoustically through the pipeline is picked off by sensors located at different distances. The next section discusses the effect of leak on the pressure and vibration in the pipe to understand how a leak detection and localization algorithm can be applied to the pipeline network.

3.1.1. Effect of Leak on Pressure

When a leak happens in a pipeline or a pipe network the pressure and flow profile of the pipeline changes, this is mainly based on the way it is being simulated because of what the boundary conditions are being set in the simulator.

For example there are two ways the boundary conditions can be set as shown figure 3.1 and 3.2 below [15] for liquid based pipelines:

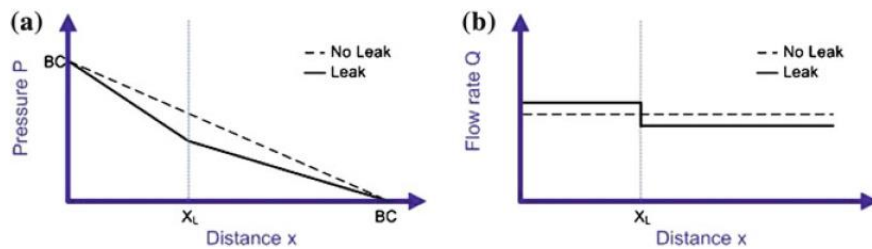


Figure 3.1: Pressure Profile in Pipeline for fixed pressure boundary conditions, X_L is the leak location

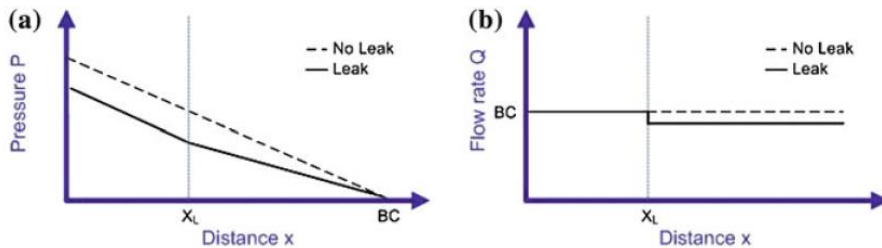


Figure 3.2: Pressure profile in the Pipeline for fixed inlet flow and exit pressure, X_L is the leak location

In figure 3.1 (a) and (b) the boundary conditions for the simulator are chosen as the pressure at the entrance and exit node of a straight pipeline under study, using these conditions for before and after the leak the pressure and flow are shown by dotted lines and solid lines respectively. We can see that in the pressure readings there is a sharp dip in the pressure gradient for a straight section of a pipeline when a leak is occurring, also the flow after X_L which is the leak location has been reduced due to the leak in the system. Similarly in the

second model we can see that the pressure gradient before and after the leak changes due to leak in the straight section of the pipeline. The boundary conditions for the second model are a bit different in that it assumes that the inlet flow and the exit pressure are taken instead of inlet and exit pressure like the first model. But we get similar response of the pipeline like that the pressure gradient changes and that flow after the leak is reduced. EPANET [101] also uses the same methods to generate a quasi-static pressure and flow profile for a pipeline network for simulation i.e. the pressure and flow data are generated assuming that they are static for the time duration that they are being acquired and that the results are generated assuming that the conditions given in the input have allowed the pipe network to settle in a steady state condition. In addition to this the boundary conditions that the simulator uses are the exit flow demand and the inlet pressure for the simulations, this way the simulator is able to run simulations for scenarios in which the inlet is able to fulfill demand of the whole network. In the next section the effect of leak on the vibrations in the pipeline are given.

3.1.2. Effect of leak on vibrations

Vibrations in pipelines are usually due to flow of the fluid occurring in it. Vibrations will also be present because of leaks in the pipeline. There has been much experimental work regarding leaks in pipelines done for detection of leaks using vibration measurements. For example in [14] the authors ran several experiments for leakage detections in pipelines for several different conditions. The major findings were that the leak spectrum is different from the flow spectrum within a certain range of values and that the standard deviation of the signals when the leak is present is different from the standard deviation when there is

no leak occurring in the system. In figure 3.3 below the effect distance on the leak spectrum is shown [14].

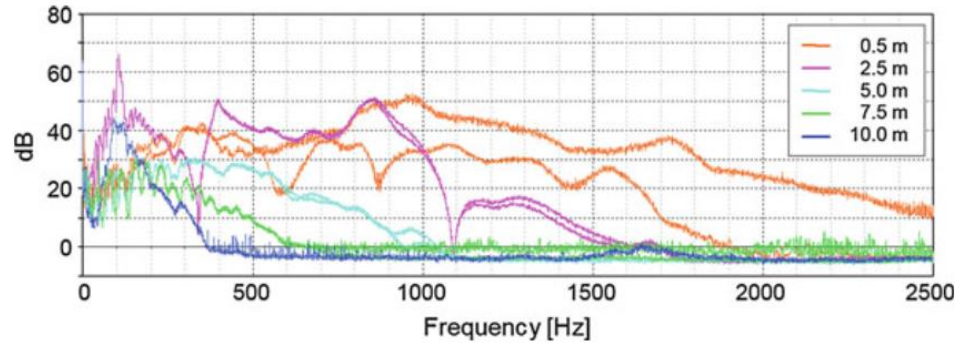


Figure 3.3: Effect of distance on leak spectrum

From the above figure it can be seen that closer the vibration sensor is to the leak the higher the spectrum of the leak it is able to capture. In effect the pipe acts as a low pass filter as in it filters out the higher frequency components further away from the leak location. Furthermore in [25] the authors did some experiments to acquire vibration data under different conditions. It was found that if the leak signature was low enough and if there was enough flow in the pipeline the leak detection would become difficult because the flow vibrations were high enough to mask out the leak vibrations. The effect can be seen in Figure 3.4 [25]. In figure 3.4 (a) shown is the effect of the leak on the vibrations when there is no flow in the system, conversely in figure 3.4 (b) when there is maximum flow occurring in the pipe the leak vibrations cannot be detected easily. Leak detection was then done by filtering the appropriate bands of the signal as the energy spectrum of the leaking vibration signals is different. Mostly the flow vibrations are concentrated in the low frequency bands and the leak frequency bands are higher in the energy spectrum. Modeling of the vibrations due to flow and leaks is done in the next section.

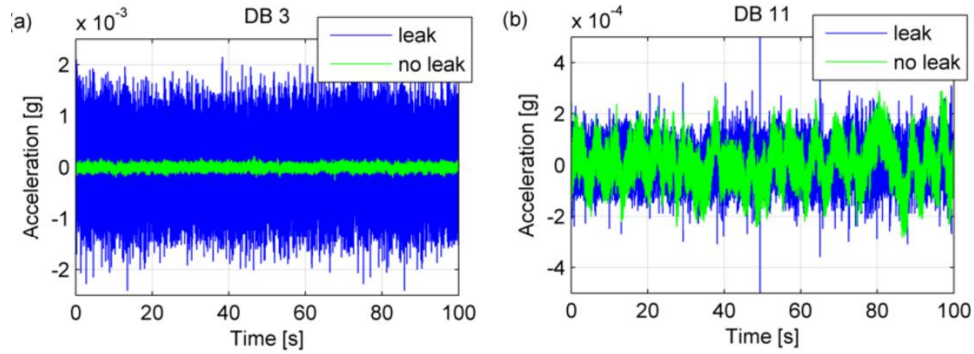


Figure 3.4: (a) No leak and Leak vibrations for DB3 and (b) No leak and Leak vibrations for position DB11.

3.2. Vibration Modeling

Vibration modeling from relevant literature is done based on two things, firstly there is vibration due to flow in pipelines, this vibration varies due to the flow and there has been a fair amount of work done to model it experimentally. And considered are vibrations due to leaks in the pipelines, this was briefly discussed in the previous section. The vibrations that happen due to leaks are more important in our work as leak detection is based on detecting these vibrations. Vibrations due to flow are important as they make up the dynamic noise floor that is present in the system, it is dynamic in a sense that it would vary based on the current condition of the pipe section under monitoring. The next subsection is related to the vibrations due to leak in the system.

3.2.1. Vibrations due to leak

Extensive work has been done by researchers to model the low pass filtering effect of the pipeline that was shown in figure 3.3. Researchers have studied the plastic pipe response to sound propagation and vibration and modeled it [102], [103] and found that the pipe acts as a low pass filter, the further the measurement point is from the leak location the lower the maximum frequency and higher the attenuation of the signal is going to be. Based on

this information the researchers further attempted to localize leak in pipelines in further work [20] and worked on cross correlation techniques to localize leaks in pipelines based on experimental data and an analytical model. The effect of filtering data was examined and it was found that it improved the result if applied up to a certain extent to the signals. This model was further used in experimental work to validate its practicability in locating leaks in underground pipelines [104]. Furthermore a virtual leak localization setup [22] was designed by the researchers based on all the experimental work done previously. The details of the model are based on the concept of leak sound propagation through the pipeline and it being picked up by the sensors, it most appropriately shown by figure 3.5.

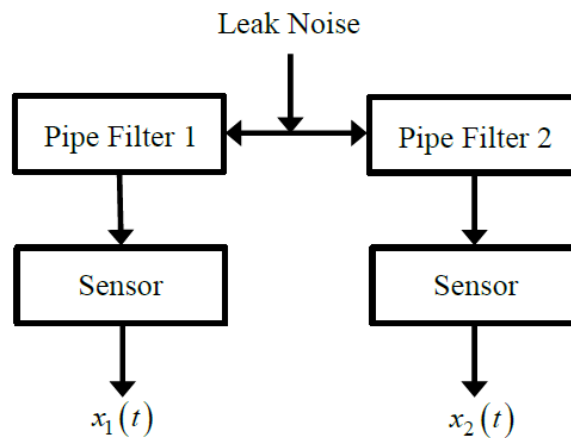


Figure 3.5: Major Components of the leak vibration model [22].

As shown in figure 3.5 there are 3 major components that contribute to the vibration signals that are picked up as signals $x_1(t)$ and $x_2(t)$ as shown in the figure above. The first major component is the leak noise, this is how the leak noise is generated acoustically at the leak source. This is based on several parameters such as the leak flow rate and the size of the hole through which the leak is occurring, based on these factors the leak noise is generated acoustically which then propagates through the pipe, this is represented as *Pipe Filter 1* and *Pipe Filter 2* in the figure above, these two pipe filters correspond to two different

lengths, and the pipe effectively acts like a low pass filter. Finally the sensor comes into play, the sensor model is dependent on what type of signal is being picked up at the measurement points, the researchers in the work discussed have worked on hydrophones, geophones and vibration sensors, these three measure the sound pressure level, the velocity profile at the surface and the acceleration profile at the surface of the pipe, of these three the hydrophones are the invasive method as it requires measuring the signal in the pipe. In our work we will be focusing on the vibration so modeling related to it will be discussed. Moving on to the leak model extensive experimental work was done in [23] regarding leak sizes and the related sound pressure level that was generated. The leak noise signal is a signal that follows the ω^{-1} power law until a specific frequency is reached for a specific leak and flow velocity. Numerous experiments were conducted for discrete diameter values and a relationship was derived for the leak noise spectrum. A signal based on this spectrum can be readily implemented in MATLAB. The empirical mathematical expression for the frequency spectrum that was derived in [23] is shown by equation (1):

$$S_{ll}(\omega) = \begin{cases} \frac{A(V, d)}{\omega} & \omega \leq \omega_c \\ \frac{A(V, d)\omega_c^{n-1}}{\omega^n} & \omega \geq \omega_c \end{cases} \quad (1)$$

In equation (1) $A(V, d)$ is the magnitude of the sound pressure level in dB for a given exit velocity V and hole diameter d in units of meter. ω is the frequency. ω_c is the critical frequency after which the attenuation factor of the sound decreases by more than the factor 1 as given in the first part of the expression.

Since the sampling frequency used is below the critical frequency ω_c , for all cases of leaks, and therefore only the low frequency components of the signal are propagated through the

pipe. As such, only the first part of expression (1) is used. In previous works [105] the range of 0 – 250 Hz is used by the authors for leak noise as the higher bands get highly attenuated. This range is only present in the first part of expression (1). Regarding the first part of equation (1), $A(V, d)$ is dependent on the following equation [23]:

$$A(V, d) = A_o \left(\frac{d}{d_{ref}}\right)^3 \cdot \left(\frac{V}{V_{ref}}\right)^2 \cdot \left(\frac{D_{ref}}{D}\right)^2 \quad (2)$$

In expression (2) d_{ref} and D_{ref} are reference leak hole and reference pipe diameter, these are set at 1 meter and V_{ref} is the reference exit velocity of the fluid through the leak, this is set at 1 m/s and A_o is the reference sound pressure level. This was experimentally found to be 10^4 Pa. Using the known leak diameter d , the pipe diameter D and the leak exit flow velocity V , the leak magnitude can be found. This magnitude can then be used in equation (1) to generate the signal spectrum. To find the leak velocity V the leak diameter will be needed. For the leak to be generated in EPANET we will be using the emitter property of the nodes available. In EPANET emitters are used as leaks which have the equation (3):

$$Q = Cp^\gamma \quad (3)$$

Here C is the emitter coefficient for discharge, γ is the emitter exponent which is usually set to a default value of 0.5, p is the pressure at the node and Q is the leak flow rate.

For modelling purposes, the Bernoulli equation is used for fluid flow through an orifice of a known diameter [106]:

$$Q = C_n A_n \sqrt{\frac{2(p_1 - p_2)}{\rho(1 - \beta^4)}} \quad (4)$$

In the above expression (4), $p_1 - p_2$ is the difference in pressure between the inside and outside of the pipe, β is the ratio of the leak diameter to the pipe diameter, ρ is the density of the fluid, A_n is the area of the orifice, C_n is the flow coefficient which is equal to 1 for ideal cases but it usually has a lower value. In our case, we have taken it to be equal to 0.6, which is a reasonable assumption for a leak from a sharp edged hole [106].

If we take $p_1 - p_2$ equal to p then C in equation (3) can be written in terms of equation (4):

$$C = C_n A_n \sqrt{\frac{2}{\rho(1 - \beta^4)}} \quad (5)$$

Using equation (5), the emitter coefficient in EPANET for a certain diameter can be approximated. This would give us the node demand which would enable us to calculate the exit flow velocity which can be used to calculate the leak noise for propagation in the pipeline model.

Once the leak noise is generated we will then propagate it through the pipe response model.

This is given by equation (6) according to the work done previously in [104]:

$$H(\omega, s) = e^{-\omega\beta s} e^{-j\omega s/c} \quad (6)$$

Expression (6) gives us the frequency response function of the pipeline based on the physical parameters of the pipeline. In expression (6) c is the speed of sound in pipeline, E is the Young Modulus of the pipe, h is the thickness of pipe, ω is the frequency and a is the mean radius of the pipe. s is the distance of the measurement point from the leak location, this is variable based on the distance of the measurement point from the leak. In expression (6) β is the attenuation factor given by equation (7).

$$\beta = \frac{(\eta Ba/Eh)}{(c_f(1 + 2Ba/Eh)^{0.5})} \quad (7)$$

In expression (7) most of the equation components have already been described before, the new things are η which is the loss factor, B which is the bulk modulus of water and c_f which is the free field velocity of sound in water. Furthermore c , which is the speed of sound in the pipeline is given by expression (8).

$$c = \frac{c_f}{(1 + 2Ba/Eh)^{0.5}} \quad (8)$$

Using the equations (6) – (8) we can construct the frequency response function of the pipe for the radial accelerometer response which is shown by expression (9) below.

$$G(\omega) = -\frac{a^2\omega^2}{Eh} \quad (9)$$

Equation (6) and (9) are combined to give the complete frequency response function of the pipe at a measurement point which is given by equation (8):

$$P(\omega, s) = -\frac{a^2\omega^2}{Eh} e^{-\omega\beta s} e^{-\frac{j\omega s}{c}} \quad (10)$$

Thus using equations (1) and (10) we can generate vibration readings at an arbitrary distance s from the leak. As discussed previously apart from vibrations due to leaks there are vibrations due to flow present also, these are discussed in the next subsection.

3.2.2. Vibrations due to flow

Vibrations due to flow in pipelines are essentially a function of the water particles hitting the surface of the pipe wall. The measured vibrations have been previously used to accurately measure the flow rate in a given pipe [8]. The researchers measured the

vibrations in different pipes and attempted to model the standard deviations of the vibrations against the flow rate in the pipe. A correlation found which predicted the flow rates relatively accurately. This was done based on the work done in [12] where the authors through a nontrivial derivation showed that theoretically the standard deviations of the vibrations measured at a certain point in the pipeline is directly proportional to the flow rate at that point. The authors then ran a series of experimental tests in which their hypothesis was proved in which that they were able to show that the experimental results matched the theoretical results. This work was further extended in [13] where the authors also did numerical and experimental analysis of the theory. The experiments were run on a relatively wide variety of materials and pipe diameters and were able to correlate all of them to a single expression that describes the vibration as a function of the different parameters of the pipes. An empirical formulation based on flow was developed as shown in the following equation (11):

$$A' \left(\frac{p_m}{p_w} \right)^{0.5} \left(\frac{t}{D} \right) = 3.5 \times 10^{-5} + (7.36 \times 10^{-18}) Re^{2.55} \quad (11)$$

where t is the thickness of the pipe, D the diameter, p_m the density of pipe material, p_w the density of water, Re is the Reynolds number and A' the standard deviation of the noise signal in units of g.

Using equation (11), the standard deviation of vibrations can be calculated and incorporated in the time series for the leak vibration data. It was noted that the vibrations became very low at low fluid velocities. Equation (11) will be used to calculate the flow based vibration as, experimentally, it shows a good fit for a wide range of pipe diameters, materials and flow rates [13].

Finally the vibrations due to flow and leak are summed up at each sensor location to give the total vibration signal at that point. It can be written down as expression shown below [20]:

$$x_n(t) = s_n(t) + n_n(t) \quad (12)$$

Here $x_n(t)$ is the time series data for the total signal available at node location n . The leak signal time series data is shown by $s_n(t)$ and is generated by passing the leak noise model equation (1) through the pipe response equation (9) for a particular distance. The noise in the signal is shown by $n_n(t)$ and represents the vibrations due to flow that are occurring in the background which are due to flow. The standard deviation for this noise is modelled by equation (11). Using this standard deviation, a random signal would be generated of the same length as the leak signal and the signals (leak and noise) would be summed up to get the complete signal at a particular sampling location. Using this formulation we can generate vibrations at different points in the pipeline as shown in the scheme in Figure 3.6.

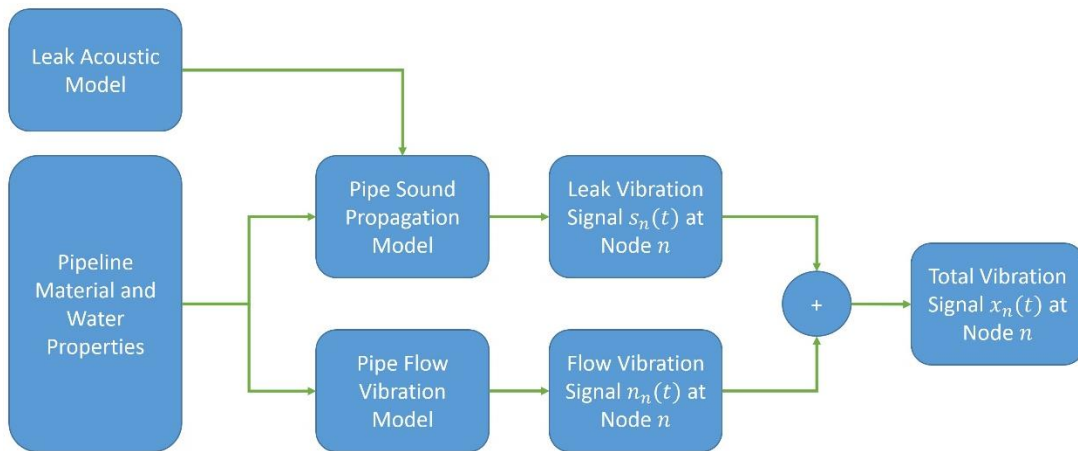


Figure 3.6: Vibration generation scheme to be used.

In Figure 3.6, the Pipe Sound Propagation model is shown by (6) based on work in [19].

This equation is shown for a specific position s_1 but will be similar for different positions

on the network with only the value of the distance s_1 changing. Applying this procedure of generating the complete noisy signal, we can now proceed to the implementation of our simulation model in MATLAB, as discussed next.

3.3. Pipeline Network simulation based on Vibration Model

In this section we will be simulating the model described in the previous section for the pipeline network for no-leak and leak conditions.

To simulate and validate the proposed sensor integration and simulation augmentation method, the scheme of Figure 3.7 is proposed which comprises EPANET where the desired pipeline characteristics and conditions are first set and then fed into MATLAB which simulates the EPANET environment and generates the required vibration data for the set pipeline conditions.

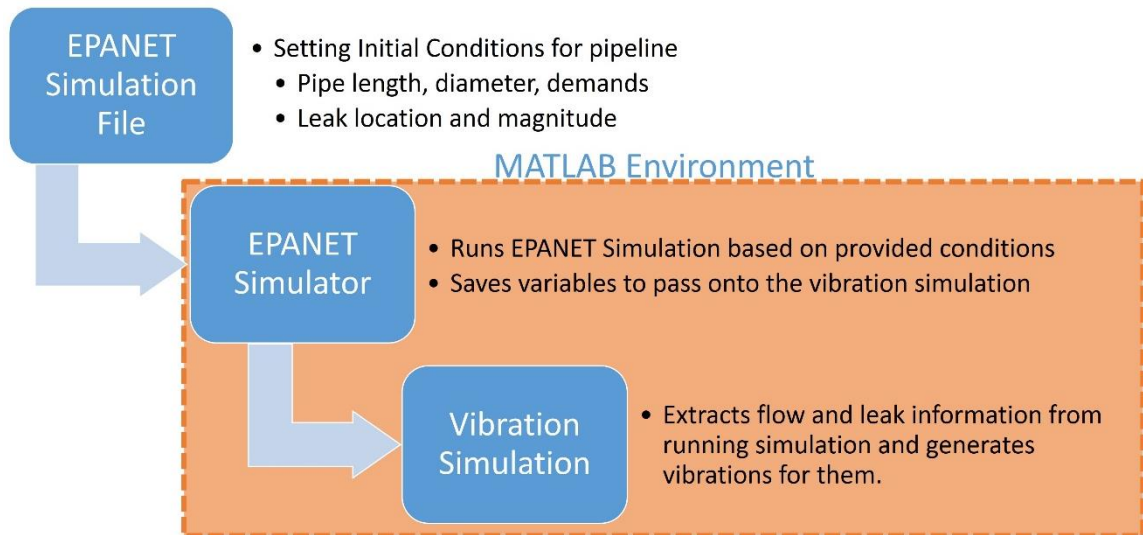


Figure 3.7: MATLAB-EPANET Co-simulation scheme

More specifically, the simulation is initiated by creating a simulation environment in EPANET specifying the pipe length, diameter and flow rate through the pipe. This is then

accessed from EPANET using a toolkit developed in [107] using MATLAB. This allows us to control the simulation and apply the leak localization algorithms to the pipe network simulated in EPANET. Furthermore using the flow information from EPANET, we can simulate the vibration in a pipeline. The vibration simulation block calculates the vibration based on the flows and leaks that are present in the system using the method described by (7) and gives the result for discrete measuring points that are defined in the EPANET simulation. Since the vibration can be measured at any point, an arbitrary point can also be defined in the pipe for vibration measurement purposes. In our simulation, the pressure sensor and vibration sensors are mounted at the same point. For the vibration signal, the length can be set arbitrarily depending on the signal length requirement of the algorithm, by setting the sampling frequency and the sampling time. In our case, a 5-second signal length are used. Longer signal lengths can be generated but they would increase the algorithm computation time.

In the simulation process, the signal spectral range is not limited, a sampling rate of 2KHz is taken, this is because for the distances simulated in the leak detection scheme, it is to be noted that at larger distances, the leak's power spectral density (PSD) drops as shown previously in data from [25], [103] and the leak signal, even if it were a broadband one, will mostly contain only data from the lower frequency bands as the pipeline attenuates the higher frequency bands. In [22], the authors have taken only a signal in the range of 0 – 250 Hz as only the lower frequency bands would propagate through the pipeline, with the higher bands getting highly attenuated. Moving on to the simulation of the pipeline network, we have selected parameters for the pipeline from references [19] based on the assumptions that the environmental conditions are consistent since this is a simulation, for

example, if the temperature changes due season or time of the day, then this will consequently change both the Young's modulus and the speed of sound, and hence the vibration-based localization accuracy. Therefore, we need readings of the environmental parameters that are very accurate if we are to achieve an accurate leak location. For a nominal operating temperature range, Young's Modulus is between 2GPa and 3GPa [108].

The pipeline parameters used in the simulation are shown in Table 3-1.

Table 3-1: Simulation Parameters

Hydraulic Diameter of Pipe	0.096	Meters
Pipe Wall Thickness	0.00856	Meters
Mean Pipe Radius	0.05236	Meters
Speed of Sound in Water	1500	m/s
Bulk Modulus of Water	2.2×10^9	Pascal
Young's Modulus of Pipe	2.4×10^9	Pascal
Pipe Section Length	50	Meters
Distance Between Sensors	50	Meters
Leak Location	21	Node
Exit Node Demand	40	Liter/minute
Leak Diameter	0.001 – 0.005	Meter

An example of a pipeline network is considered, and simulated in EPANET using the conditions given in Table 1, such as pipe length, hydraulic pipe diameter, flow rate, distance between sensors to run the flow- and pressure-based simulation using MATLAB.

3.3.1. Simulation of Pressure Data

For the example considered in Figure 3.8, there are 19 nodes (labeled as nodes 1 to 19) for pressure data acquisition. The small rectangle labeled A is modeled as a 3mm-leak according to equations (3-5). Node 15 represented by the triangle is the output node, the leak is 23 meters from Node 10 and 27 meters from Node 14. The simulation is run in EPANET and the pressure data is acquired for 3 different scenarios for the leak. The leak diameter is varied from 1mm in size to 5mm in size and the simulation run to collect data for the different leak diameters. The pressure profile for the 1mm leak is shown in figure 3.9. We can see that for the leak the pressure is dropping across all nodes in the network. This is mainly because the pump needs to pump more fluid to keep up with the exit node demand of 40 liter per minute as well the demand of the leak at position A.

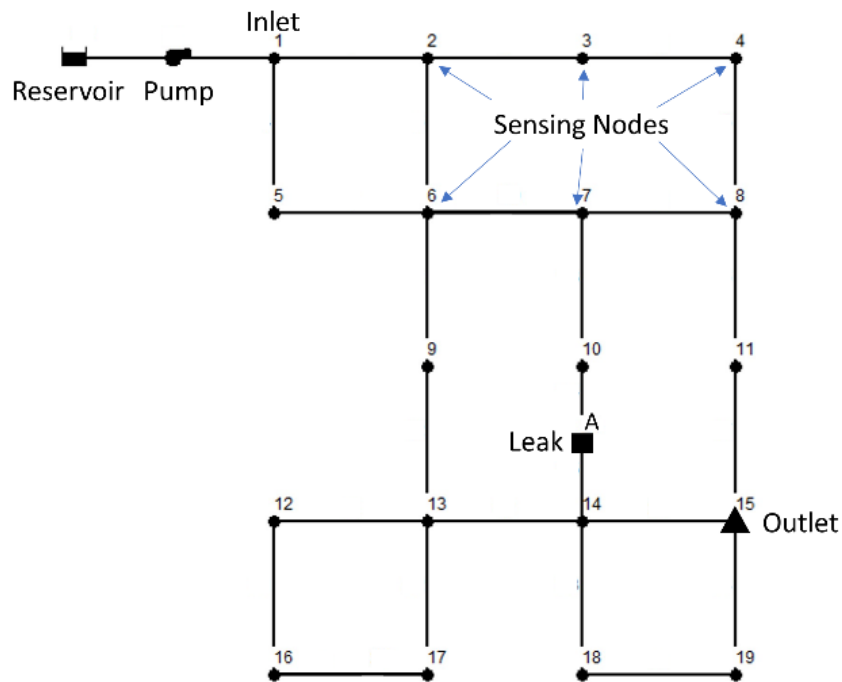


Figure 3.8: EPANET Model of pipeline, the rectangular position is the leak location and the triangle node is the network exit node with demand of 40 liters per minute.

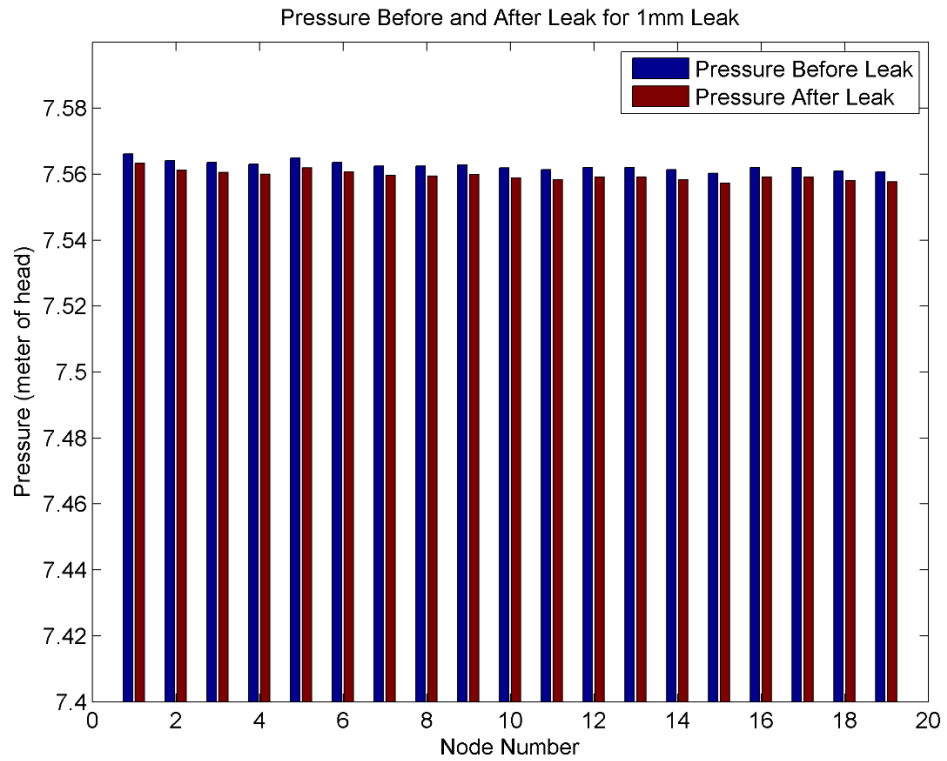


Figure 3.9: Node pressure profile before and after leak for 1mm leak

The plotting for pressure for the 3mm leak case is shown in figure 3.10.

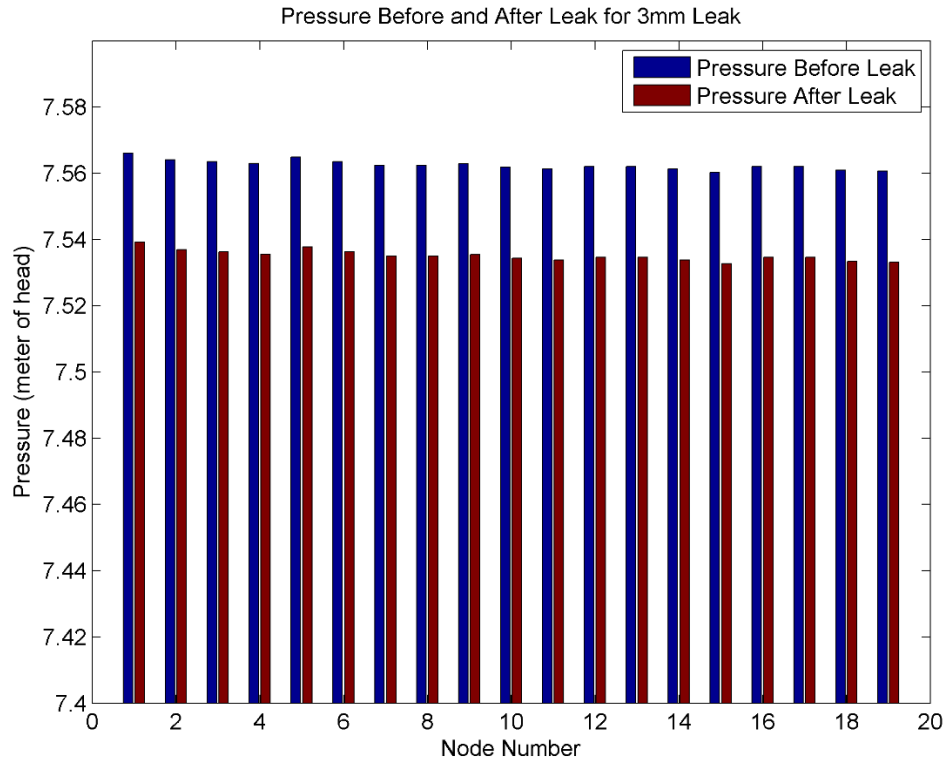


Figure 3.10: Node pressure profile before and after leak for 3mm leak

In figure 3.10 it can be seen that the pressure drops more than the 1mm leak case because the demand of the leak is higher at 3.1 liters per minute. The leak diameter was further increased to 5mm and the simulation was run again to find the pressure drop due to the leak, it is shown in figure 3.11. The pressure drop is higher compared to the other cases as the leak magnitude is higher at 8.56 liters per minute. The leak rate would also change with the pressure but a constant output is assumed and it wouldn't change drastically with output flow rate near to the assumed 40 liters per minute. Moving on the next subsection is related to the vibration readings at different nodes that were generated by the developed simulation scheme described previously.

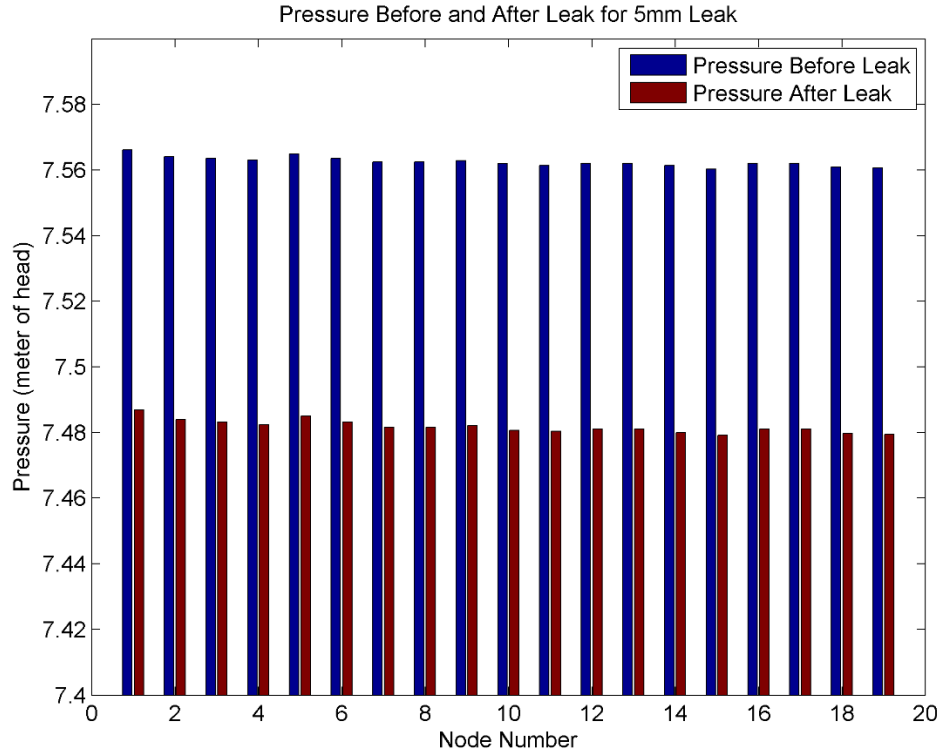


Figure 3.11: Node pressure profile before and after leak for 5mm leak

3.3.2. Simulation of Vibration Data

The vibration data was simulated according to equations (1 – 12) described in the sections previously. The simulation was run on four data acquisition nodes that were nearest to the leak location as the effect of the leak is largest nearest to the leak itself. We will be looking at the effect of the leak on Nodes 10, 14, 7 and 18 which are nearest to the leak location in a straight line, there are other nodes which are the same distance from the leak similar to Nodes 7 and 18, and the effect of the leak will be similar to them because they will be effectively the same distance from them. The noise floor is made up with the vibration due to flow. The vibration magnitude of the low leak is expected to be lower than the vibration magnitude of the high leak. We will see that effect in the results below. The vibration is sampled at a 2 KHz rate for 5 second intervals. Figure 3.12 shows the effect of leak on the vibration at the 4 mentioned measurement nodes for a 1mm leak.

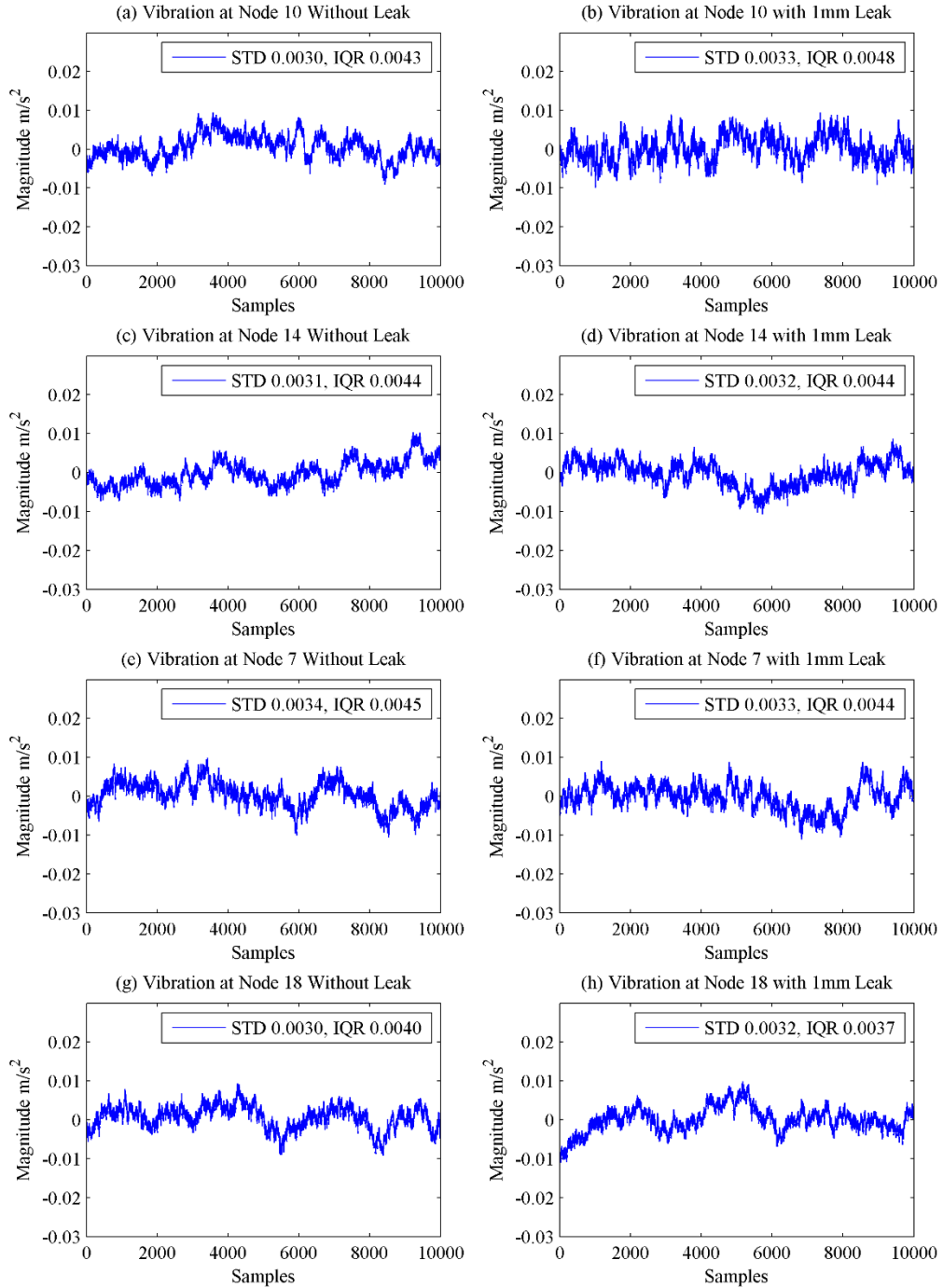


Figure 3.12: Vibrations at 4 different nodes before and after the leak for the 1mm leak case. Node 10 is 23m from the leak, Node 14 is 27m from the leak, Node 7 is 73m from the leak and Node 18 is 77m from the leak.

In figure 3.12 we notice that at the nodes nearest to the leak the magnitude of the leak signature is higher, also the standard deviation and interquartile range (IQR) of the signal

was calculated, it was found that both of these are directly proportional. In a WSN the IQR gives an attractive alternative to the standard deviation for signals as it only requires sorting

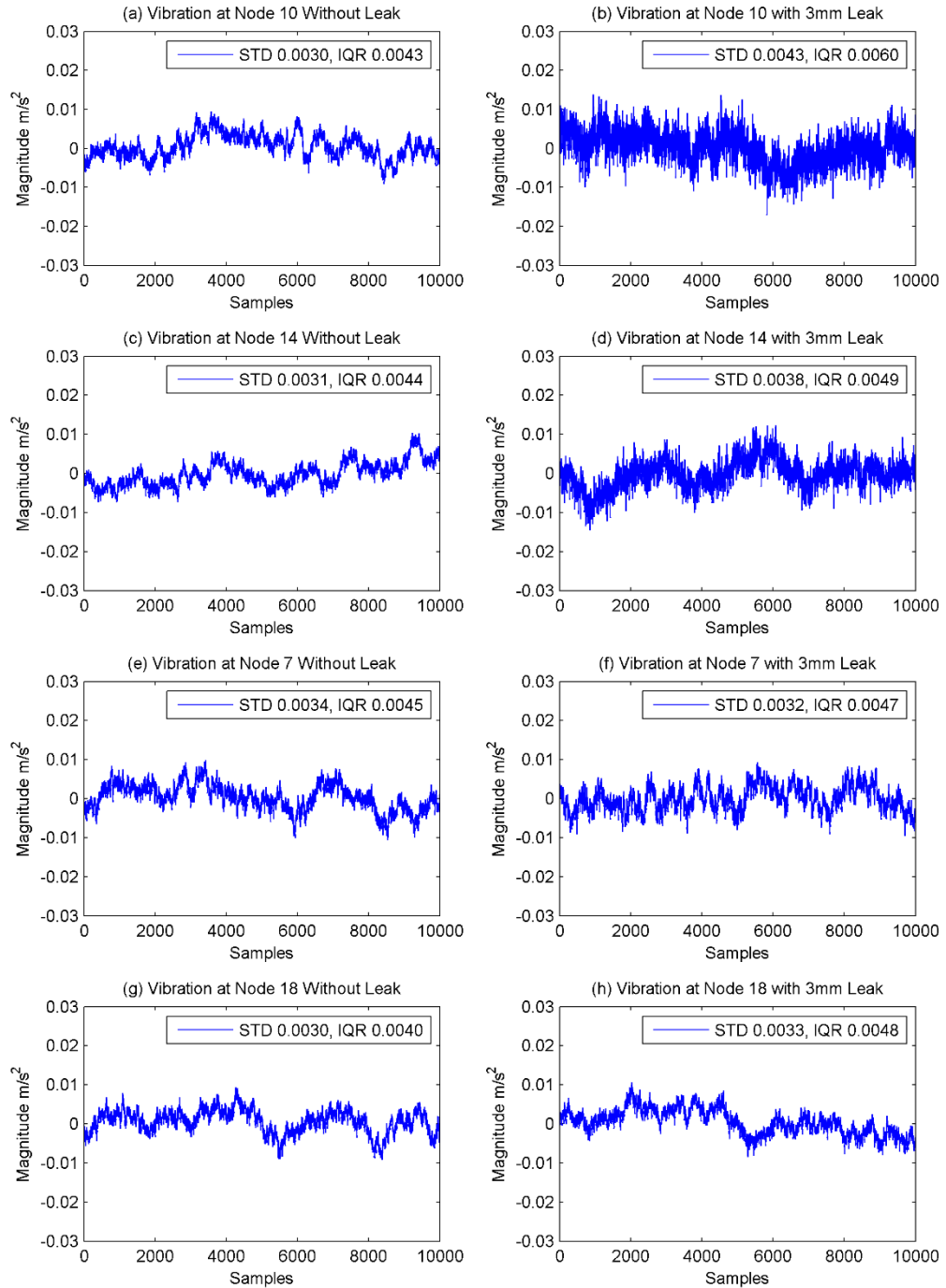


Figure 3.13: Vibrations at selected nodes for a 3mm leak, it is noticed that at nodes 10 and 14 the leak signature is more prominent now.

of the signal. Also for the 3mm leak the vibrations are plotted in figure 3.13. In figure 3.13 we can see that the leak is now more prominent at Nodes 10 and 14, also the IQR computed

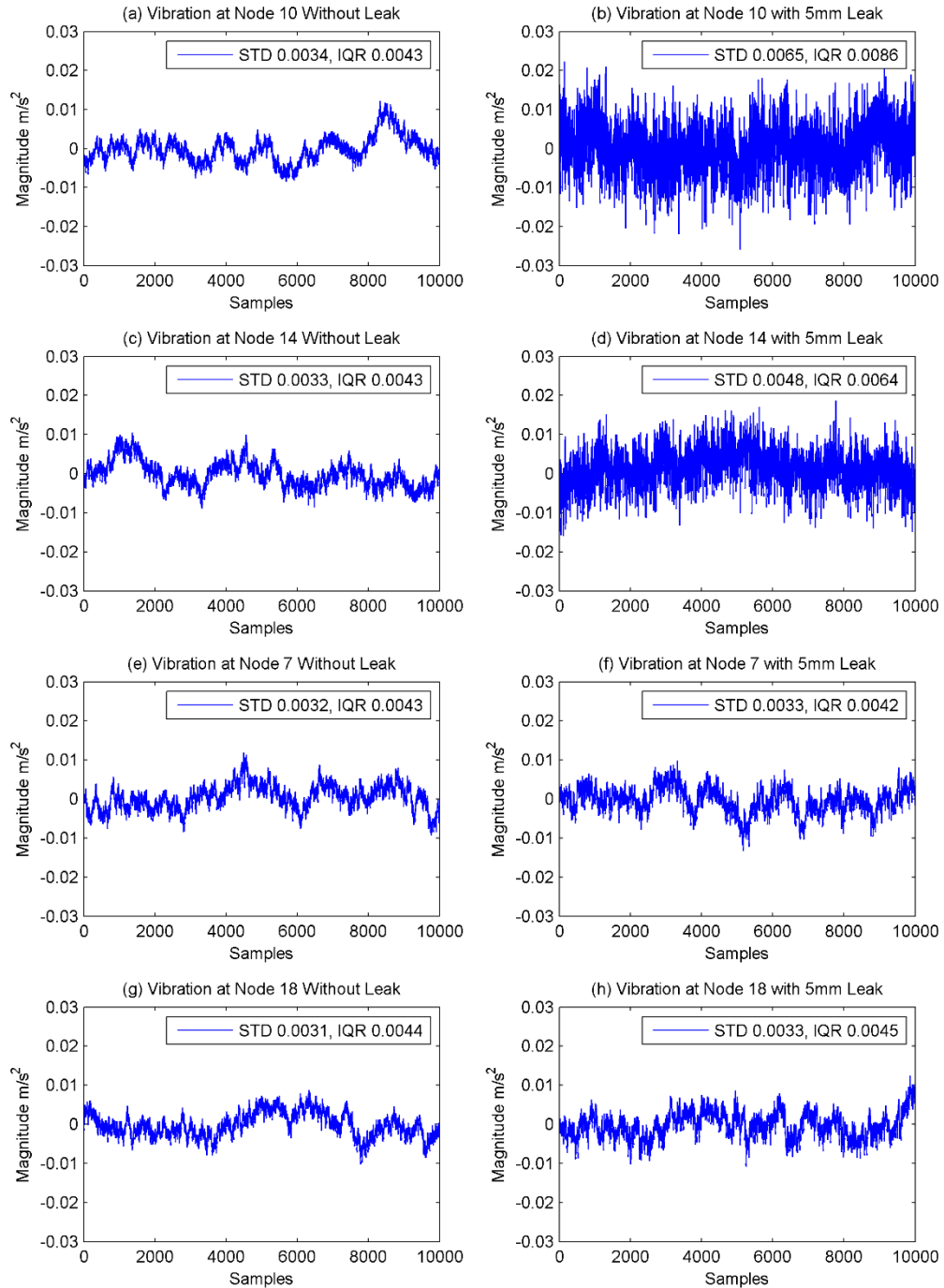


Figure 3.14: Vibration signature at selected nodes for 5mm leak. The leak is more noticeable in Nodes 10 and 14 but at Nodes 7 and 18 there is no effect of leak.

at Nodes 10 and 14 registers an increase over the reading before the leak. Finally in figure 3.14 we can see the vibrations for the 5mm leak case and we can see that at Nodes 10 and 14 the vibration magnitude increases as they are very close to the leak location but at Nodes 7 and 18 the vibration magnitude does not change as they are very far away from the leak location and the leak signature is attenuated by a large magnitude.

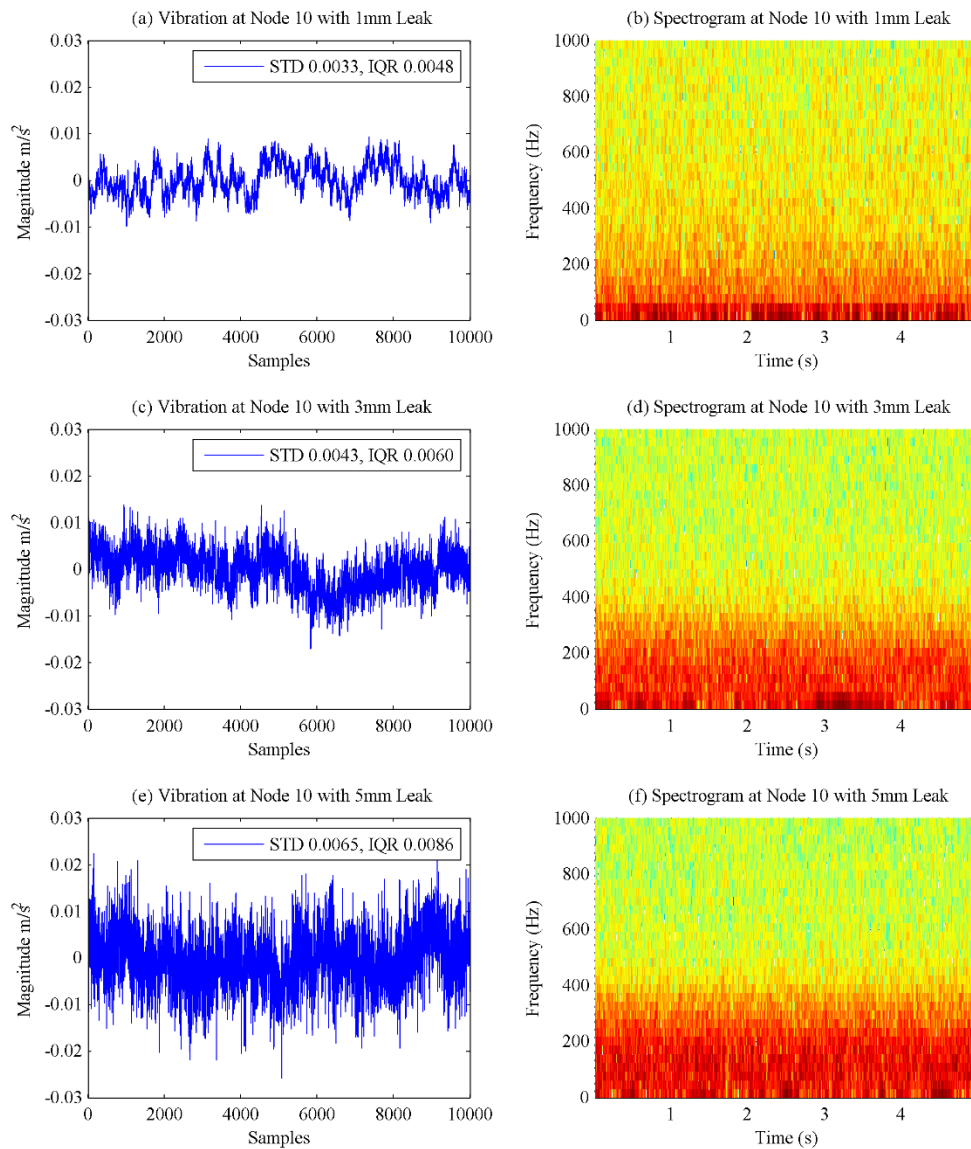


Figure 3.15: Spectrogram of the different leaks at Node 10, 23m from the leak.

To further see the effect of leak on the vibration signals at Nodes 10 and 14 the spectrograms of the 1mm, 3mm and 5mm signals were plotted and compared. Figure 3.15 shows the comparison of the frequencies at Node 10 for the three leak magnitudes.

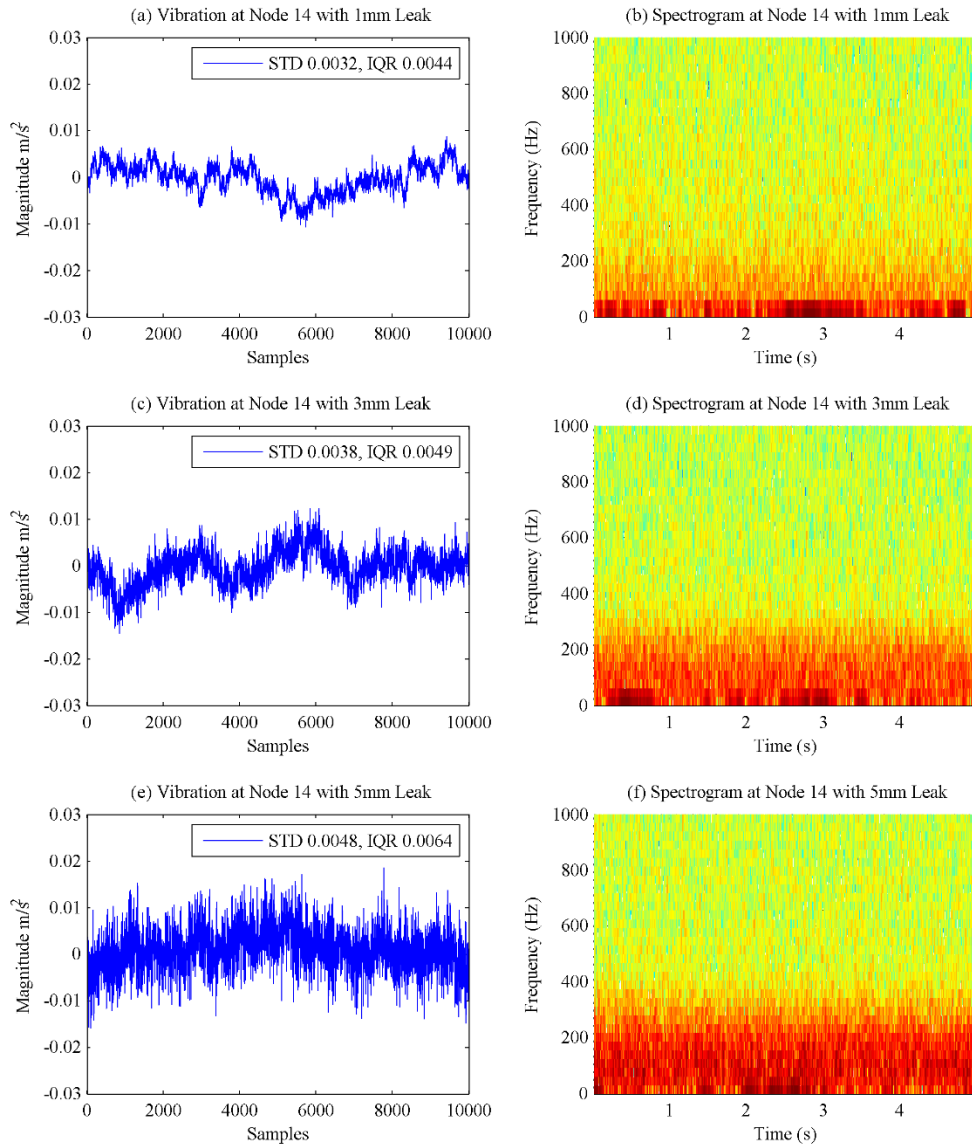


Figure 3.16: Spectrogram of the different leaks at Node 14, 27m from the leak.

Here we can see that the IQR of the vibration signal at Node 10 is increasing along with the leak magnitude and that the bandwidth of the signal at Node 10 is also increasing as the leak magnitude increases, the spectrum is only till approximately 400 Hz as this is the

response of the pipe. With respect to the 3mm and 5mm leak in figure 3.15(c) and figure 3.15(e) these both have the same bandwidth but the magnitude of the 5mm leak is higher as shown by the IQR in figure 3.15(e) as well as the intensity shown in figure 3.15(f) compared to figure 3.15(d). Next the vibrations at Node 14 were plotted and compared for 1mm leak to the 5mm leak.

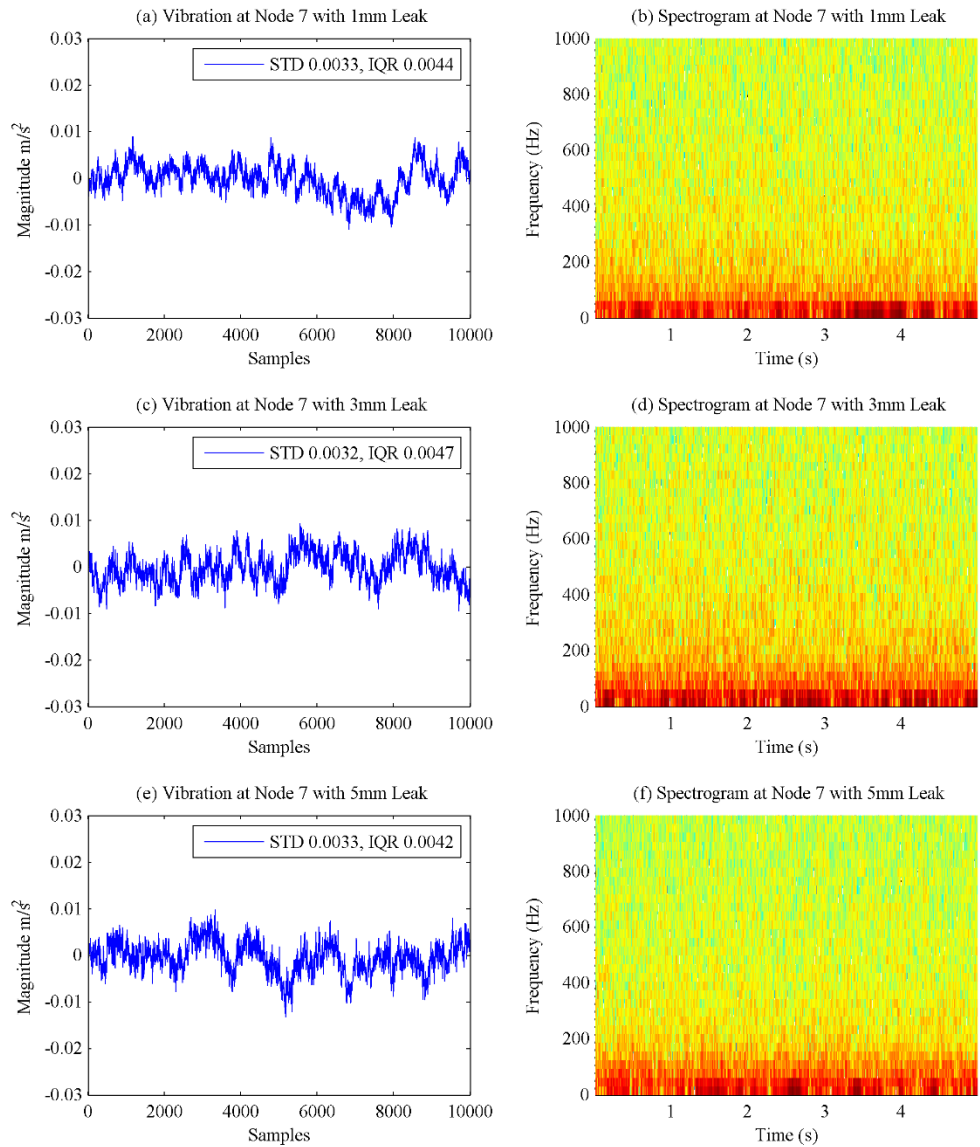


Figure 3.17: Spectrogram of the different leaks at Node 7, 73m from the leak.

In figure 3.16 we can see that since the leak is further away from the measurement node the bandwidth of the signal is lower, for example in figure 3.15 the bandwidth was up to 400Hz, in this case the bandwidth is up to 300Hz.

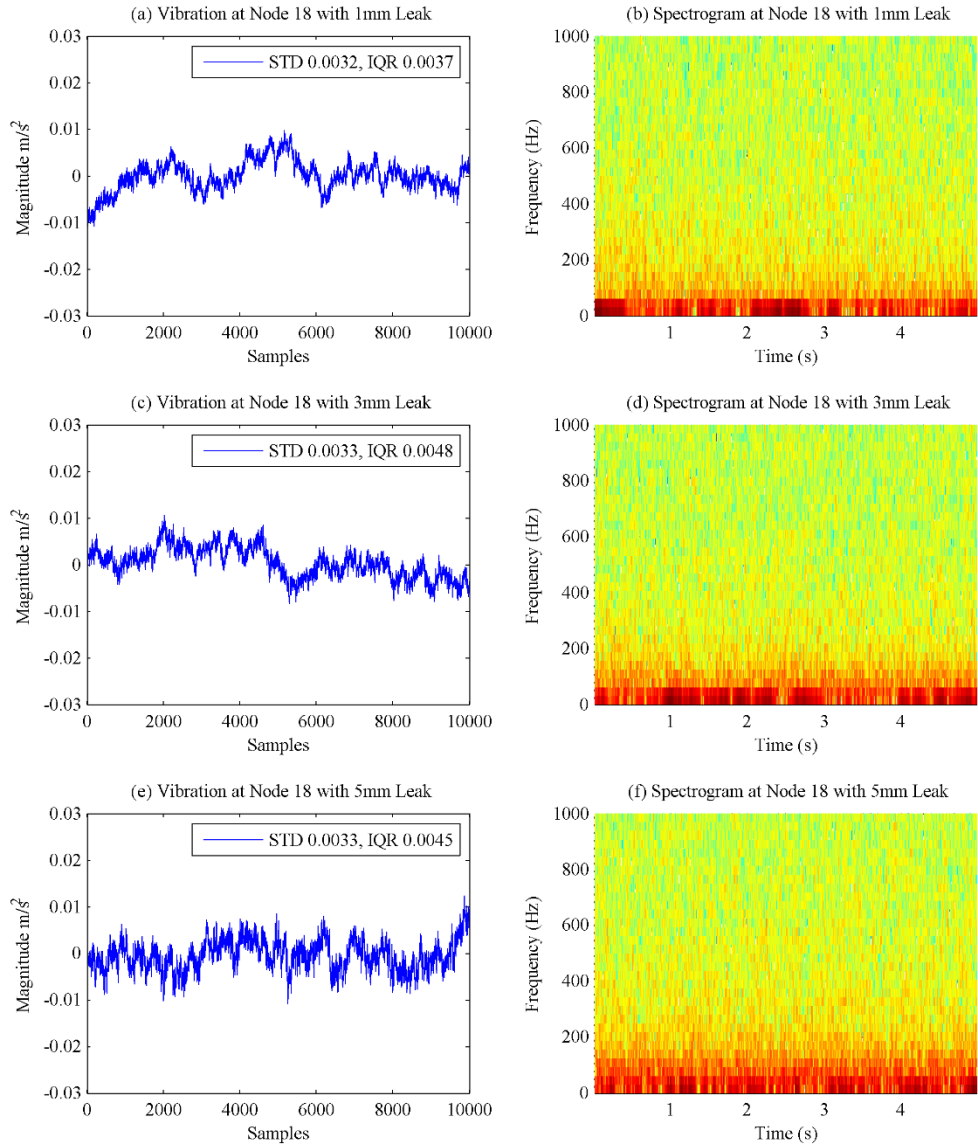


Figure 3.18: Spectrogram of the different leaks at Node 18, 77m from the leak.

For the 1mm leak case the vibration does have a minor component of the leak signal as evident in the spectrogram but it is very low in magnitude, and because of this reason the IQR and standard deviation of the signal are similar to the pre-leak values, on the other

hand for the 3mm leak the IQR and standard deviation is higher and the leak signature is more evident in the spectrogram and it is similar for the 5mm leak case. It should be noted that the bandwidth of the leak signature in 3.16(d) and 3.16(f) are similar and this is because of the pipe frequency response because of the distance involved, this is similar to the results obtained for Node 10 in figure 3.15. Similarly the effect of leak on Nodes 7 and 18 is plotted according to the frequency analysis in figures 3.17 and 3.18 respectively. It can be seen that the IQR and standard deviation of the signal is within the same range for all signals and it is evident from the spectrum that the leak signature is not present in the acquired data. This will be important when we would be applying energy efficiency based leak detection and localization technique to the problem. The next section is related to the model of the WSN node that will be used in the developed monitoring solution.

3.4. WSN Node Model

This section is related to the WSN nodes that will be used in the simulation. There are four main components [10] in the WSN node, the power supply, the sensing subsystem, the processing subsystem and the communication subsystem as shown in Figure 3.19.

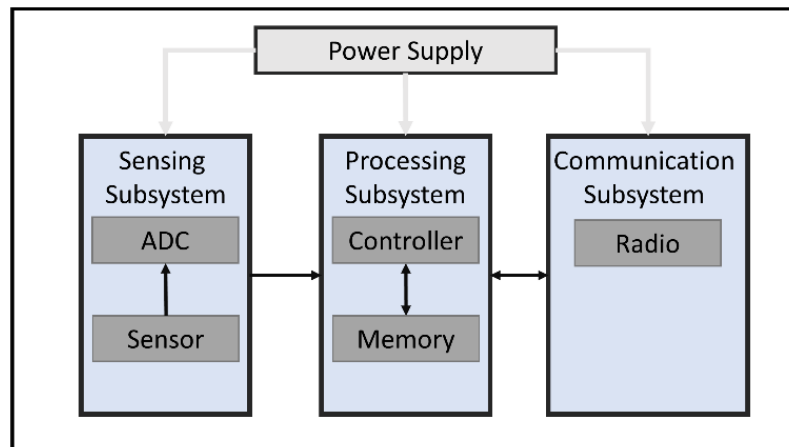


Figure 3.19: WSN Node main components

For pipeline monitoring applications usually pressure, flow [11], [16], [18], [30], [109] and acoustic or vibration [9], [19], [21], [25], [109], [110] sensors are used to detect and localize leaks in the network. As such, in the literature, a vast amount of alternatives are available for monitoring solutions and each one has its own strengths and applications which are described in more detail in [111]. Usually event detection is improved when a multi-modal approach is used in systems, and for such a scenario in pipeline monitoring, there are multiple approaches that can be used and we have used an approach in which pressure and vibration monitoring can be jointly used. In addition to this, most of the vibration sensors being discussed in the literature, are of either MEMs or piezoelectric type, and the main difference between both of them is that piezoelectric vibration sensors are more sensitive and have lower noise, As a result of this, they use higher energy. On the other hand, MEMs sensors have more noise and hence consume less power. Using both of them would give us a hierarchical scheme [93] in which energy consumption would be further reduced when using lower energy sensors in our adaptive sampling scheme, as discussed in the next chapter. The pressure sensor being used in our system is the standard 4-20 mA sensor used in the industry, and the vibration sensors are the SD-1221-002L for the Low Sensitivity Vibration Sensor (LSVS) and PCB Piezotronics 626B03 accelerometer for the High Sensitivity Vibration Sensor (HSVS). The microcontroller being used is from the MSP430 line of microcontrollers as it has been used extensively in WSN applications and the radio is the CC2420, also selected because it is extensively used and sufficient literature is available to model it effectively. The next subsection will deal with the equations used for the energy modelling of these node components.

3.4.1. Energy Equations

The total energy consumption of the node depends on the components used, the operations it performs and the duty cycling scheme used. It is assumed that all communications take the minimum path and are accomplished in a single hop. In [112], [113], the power consumption model of all the node components is given for a single cycle as:

$$E_i = e_{radio} + e_{comp} + e_{sensing} \quad (13)$$

Here e_{radio} is the energy consumed by the radio for transmission, e_{comp} is the energy consumed by the computation unit or processor and $e_{sensing}$ is the energy consumed by the sensing subsystem which includes the sensor and the ADC. This can also be written as:

$$\begin{aligned} E_i^w = & (e_{radio}^s + e_{comp}^w + e_{sensor}^w + e_{adc}^w) \times T_{sample} + (e_{radio}^s + e_{sensor}^s + e_{adc}^s) \\ & \times T_{comp} + E_{comp}^w + (e_{comp}^w + e_{sensor}^s + e_{adc}^s) \times T_{radio} \\ & + E_{radio}^w \end{aligned} \quad (14)$$

Here the superscript E_i^w indicates that this equation is for the wakeup period i and the superscript w and s indicate for each component that it is either awake or asleep. The radio will be sleeping for T_{sample} and T_{comp} which are the sampling and computation durations. The processor will be awake for the whole period and the sensor and ADC will only wake up for T_{sample} which is the sampling period. Further on, the processor will be awake for the time the radio is active T_{radio} so its energy consumption during that time period will also be taken into account. E_{radio}^w is the energy the radio uses the whole wakeup cycle for the different operations, the comprehensive model is given in [114]. For a duty cycling scheme the sleeping energies would also need to be accounted for also for sleep period i and it is given in the equation below:

$$E_i^s = (e_{radio}^s + e_{comp}^s + e_{sensor}^s + e_{adc}^s) \times T_i \quad (15)$$

Thus the total energy consumption of the sensor for one complete sampling and sleep period i of the duty cycle will be:

$$E_i = E_i^w + E_i^s \quad (16)$$

For the processor energy consumption E_{comp}^w , the model given in [112], [113] is adopted as it bases the energy consumption on the computational complexity of the algorithm.

$$E_{comp}^w = O(m) \times \mu \left(\frac{f}{k} + \beta \right) \quad (17)$$

Here μ is the switching capacitance, f the operating frequency and k and β the hardware constants. From (17) we can calculate the energy consumed at a specific processor frequency and find out the time consumed T_{comp} if we know the processor's current and voltage at that frequency. From this time sleeping energy of the remaining inactive node components will be calculated. All the other components have the sleeping and active energies expended per second using the small e notation, as in e_{radio}^s .

For E_{radio}^w , the expression, modified only for the wakeup period, is taken from [114] and the parameters used are also taken from it. The expression for E_{radio}^w is shown below:

$$E_{radio}^w = E_{tx} + E_{rx} + E_{cca} + E_{idle} + \sum_{i=0}^n E_{state-switch} \quad (18)$$

Here E_{tx} is the transmission energy, E_{rx} is the receiving energy, E_{idle} is the energy consumed in the idle state and E_{cca} is the energy consumed while performing CCA. Here $E_{state-switch}$ is the energy required to switch between states when the radio is switching

between idle, sleep, tx, rx and cca states, n is the number of switches required and is usually dependent on the number of operations that the radio will be doing in a single cycle.

For E_{tx} , the computational time is basically the number of bytes divided by the transmission rate. Since the packet is broken down into 102 byte packets [115] and each requires an overhead of 31 bytes, the transmission energy is then shown to be in the equation (19):

$$E_{tx} = V_{radio} \times I_{tx} \times T_{tx}$$

$$T_{tx} = \left(\frac{\text{data bytes}}{Tx \text{ Rate}} \right) + \text{Ceiling} \left(\frac{\text{data bytes}}{102} \right) \times \left(\frac{31}{Tx \text{ Rate}} \right) \quad (19)$$

Here V_{radio} is the voltage provided to the radio and I_{tx} is the transmission current. For E_{rx} , the calculation is similar. In (19), the Tx Rate is assumed at 250Kbps which is the nominal rate at which most 802.15.4 devices work at. E_{cca} is also calculated in a similar manner. CCA usually takes 128ms and in our case, we are taking the most ideal conditions with free channel and no interference so that it will only be performed once and the data transmitted right after it. The equation is shown below:

$$E_{cca} = V_{radio} \times I_{cca} \times 0.000128 \quad (20)$$

E_{idle} will be calculated in a similar way. Idle time is between 2 consecutive tasks of the radio. Or when radio has been woken up by the microprocessor in the node and is awaiting to receive or transmit instructions. Between tasks the radio will go into idle mode, and the energy for such state switching must also be accounted for. Such a formula is already proposed in [114] and will be used accordingly.

For T_{radio} , the active time spent in the whole cycle will be calculated as the sum of the transmission time, receiving time, idle time, CCA time and total time spent between transitions, otherwise, i.e. if the radio is not being used, it will be equal to 0. An expression for typical T_{radio} in a single cycle for λ number of packet transmissions and ρ number of receiving packets is given below:

$$T_{radio} = T_{sleep-idle} + \rho \times (T_{rx} + T_{ifs}) + T_{idle-cca} + T_{cca} + T_{cca-idle} + \lambda \times (T_{tx} + T_{ifs}) + T_{idle-sleep} \quad (21)$$

Here T_{ifs} is equal to 640 μ s and is known as the inter-frame sequence which is the delay between 2 successive transmissions [115]. All the other times are defined in [114] and are given in table 3-2. The currents for the radio states are given in table 3-3.

Table 3-2: CC2420 State Transition Times

State Transition Type	Time
$T_{off-idle}$	1 ms
$T_{cca-idle}$	2 μ s
$T_{idle-cca}$	192 μ s
$T_{sleep-idle}$	0.6 ms
$T_{idle-sleep}$	192 μ s
$T_{tx-idle}$	2 μ s
$T_{idle-tx}$	192 μ s
$T_{rx-idle}$	2 μ s
$T_{idle-rx}$	192 μ s

Table 3-3: CC2420 State Currents

State	Current
I_{off}	0.02 μ A
I_{tx}	17.4 mA
I_{rx}	19.7 mA
I_{idle}	426 μ A
I_{cca}	17.4 mA
I_{sleep}	20 μ A

Using the information given in tables 3-2 and 3-3 the energy of the radio for transmission or receiving any amount of data from the node can be computed. Using the equations in this section, we would be able to model the energy consumption of the node components under different conditions and be able to simulate the energy profile of the nodes under study. The components that will be used for the nodes are already mentioned in the previous section and Table 3-4 lists the energy consumption of each component that will be used to calculate the energy from equations (14) and (15):

Table 3-4: Node Components power consumption for different states

Device	State	Symbol	Voltage	Current
<i>MSP430f series Microcontroller</i>	Active	e_{comp}^w	2.2V	2.93 mA
<i>MSP430f series Microcontroller</i>	Sleep	e_{comp}^s	2.2V	68 μ A
<i>SD-1221-004L Vibration Sensor</i>	Active	e_{sensor}^w	5V	5 mA
<i>SD-1221-004L Vibration Sensor</i>	Sleep	e_{sensor}^s	0V	0

<i>PCB 626B03 Vibration Sensor</i>	Active	e_{sensor}^w	24V	2 mA
<i>PCB 626B03 Vibration Sensor</i>	Sleep	e_{sensor}^s	0V	0
<i>Omega PXM-409 Pressure Sensor</i>	Active	e_{sensor}^w	24V	12 mA
<i>Omega PXM-409 Pressure Sensor</i>	Sleep	e_{sensor}^s	0V	0
<i>ADS1220 ADC</i>	Active 2KHz	e_{adc}^w	5V,3.3V	540 μ A, 95 μ A
<i>ADS1220 ADC</i>	Active 1KHz	e_{adc}^w	5V,3.3V	340 μ A, 75 μ A
<i>ADS1220 ADC</i>	Active 250Hz	e_{adc}^w	5V,3.3V	95 μ A, 55 μ A
<i>ADS1220 ADC</i>	Sleep	e_{adc}^s	5V,3.3V	0.3 μ A, 0.1 μ A

For microcontroller the hardware constants used in equation (17) are shown in Table 3-5:

Table 3-5: Hardware constants for microcontroller

<i>Constant</i>	Value	Unit
μ	17.8	nF
f	8	MHz
k	7.9	MHz/V
β	1.8	V

Using the parameters in these tables and the energy consumption for each state of the radio defined in [114] for equation (21) , we can model the energy consumption of the node for

the whole simulation time. For the simulation, we are additionally assuming that the nodes will be transmitting at maximum power and there is only one hop from the node to the sink, and there are no nodes in between. We will be using the energy consumption model when gauging the performance of the energy efficiency techniques that we will be developing in the next section.

CHAPTER 4

ENERGY EFFICIENCY BASED ALGORITHM FOR PIPELINE MONITORING

This chapter is related to the energy efficiency techniques that will be developed and applied to the WSN nodes that will be monitoring the pipeline under different scenarios. It should be noted that we are aiming that timely leak detection and localization takes place along with energy conservation. The algorithm will provide a general framework along with the pipeline monitoring scheme will work at each WSN node and together as a complete system. Moving on to the algorithms that will be making up our monitoring scheme at node level for energy efficiency, we will be using duty cycling to wake up the sensor at pre-set intervals rather than keep them on the whole time, this would reduce the power consumption significantly by having the nodes sleep for the majority of the time and only wake up to sense for a small time and perform computations on the signal and if necessary perform data receiving and transmission by the radio. The second portion of the scheme will use an adaptive sampling algorithm with hierarchical sampling added, this would provide a two-fold reduction in energy consumption by first varying the sampling rate according to the frequency content of the vibration signal in the pipeline, and then using the lower energy sensor for extended sampling periods in which the vibration will be sampled at a lower frequency. After this step, since the signal will need to be transmitted to the sink a signal compression algorithm will be applied to reduce the amount of the data

that will be sent. For the last two steps the computation cost will be more involved than normal methods in which both of these techniques are not used but the energy savings due to both the HSVS and radio working less will pay off and give a net gain in energy consumption as opposed to a solution where no energy efficiency techniques are applied.

We aim to apply here the developed algorithm in an energy-constrained environment for monitoring a water pipe network using WSN nodes. The nodes are placed at fixed locations in the pipeline network, and each location has a pressure sensor, and two vibration sensors, one of which is high-accuracy (and consumes more energy), and the other is low-accuracy (and consumes less energy). All nodes carry a radio transmitter to transmit all the data to the sink. A low-energy microcontroller will be handling tasks for the system as these are extensively used in WSN applications. The algorithm that is being executed at each node is described in the following subsections. The first portion of the algorithm is the duty cycling described in the next sub-section.

4.1. Duty Cycling

For duty cycling a fixed-sampling scheme is adopted in which the node would wake up every fixed amount of time and sample the vibrations for a fixed duration of time and then go to sleep, this would ensure that all nodes are waking up at the same time and that the data is synchronized with respect to time, this would be helpful if the leak detection and localization algorithm is taking time synchronized data from multiple nodes to monitor the system. The timing equation for one complete cycle of the duty cycling scheme is shown below:

$$T_t = T_{active} + T_{sleep} \quad (22)$$

Here T_t is the total time duration of one cycle of the node which includes the wakeup time T_{active} , and the low power sleeping time T_{sleep} . T_t is constant as the node would wake up at fixed time intervals, and T_{active} and T_{sleep} will depend upon the number of operations that will take place during T_{active} . These are shown by equation (23) shown below:

$$T_{active} = T_{sample} + T_{comp} + T_{radio} \quad (23)$$

In equation (23) T_{sample} is fixed, since we assume that the sampling duration is fixed and energy is saved through adaptive sampling by varying the sampling frequency. T_{comp} depends on the number of computations that are carried out in one cycle of the scheme, T_{radio} is the time taken by the radio to complete its operations if it is active. If the radio is not active, then T_{radio} is equal to 0 for that cycle, and due to these factors T_{active} will vary. However, if T_{active} is varying, then T_{sleep} will be varying accordingly to keep T_t constant. The duty-cycling scheme is illustrated in figure 4.1 assuming no radio usage.

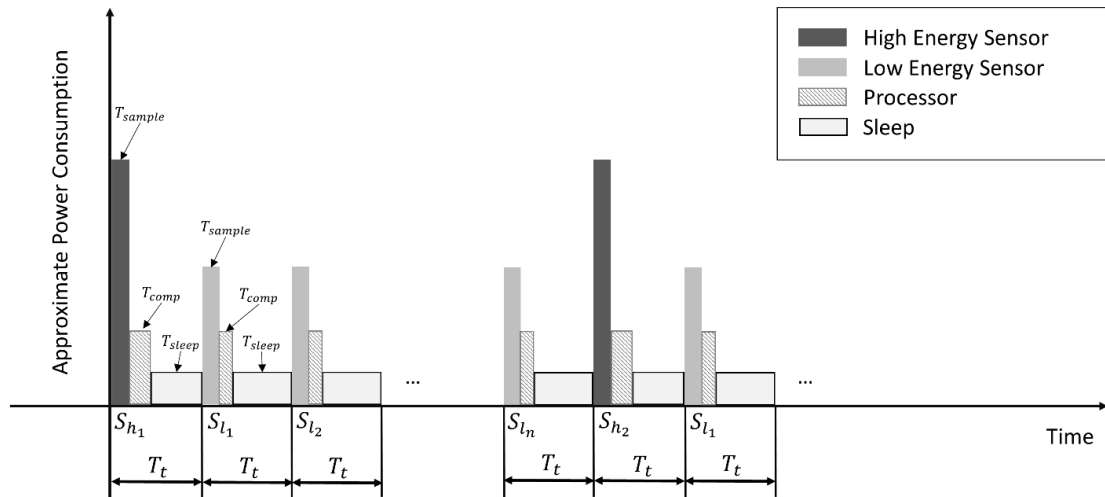


Figure 4.1: Duty-Cycling Scheme used in the algorithm, dark grey bar indicates sampling with high-energy sensor at high-frequency sampling rate, and light grey bars indicate sampling with low-frequency sampling rate.

In figure 4.1, it can be seen there is a dark grey bar labeled S_{h_1} and light grey bars labeled S_{l_1} and S_{l_2} till S_{l_n} . This denotes the sampling by the sensors done at a high frequency and low frequency respectively. When the node wakes up in the first sampling instant S_{h_1} it uses the high frequency vibration sensor to sample the signal at F_{max} bandwidth and set a particular frequency of F_h bandwidth which denotes the highest detectable frequencies it is able to measure in the signal, it is assumed that F_h would be either less than or equal to F_{max} because F_{max} is set sufficiently high that it would be able to measure all of the frequencies in the spectrum. The sampling duration during this step is taken as T_{sample} . After this step, the WSN goes to sleep for period T_{sleep} . In the next cycle S_{l_1} , the WSN would sample the vibration signal at the rate adjusted according to F_h and would perform threshold calculation operations for leak detection based on it and go to sleep. This cycle will continue until either an event has been detected or we reach S_{l_n} , the maximum number of times we can sample at the adjusted rate until F_h will be readjusted. After this step, the high-frequency sensor would be used again to set F_h after sampling at F_{max} in sampling instance S_{h_2} . T_{radio} is not shown in figure 4.1 but will be within T_{active} for cycles where the radio will be used. Furthermore, in each high-frequency sensor sampling instance, the adaptive sampling algorithm is used. Apart from duty cycling for vibration sensors, it would also be applied to pressure sensors but in a separate loop for monitoring. The pressure sensor duty cycle would be of a longer duration because the pressure sensor is a higher-energy sensor and the readings would mostly be used for monitoring purposes and detecting leaks if vibration based methods are insufficient for leak detection. The next section is related to the discussion of the adaptive sampling algorithm which was mentioned above.

4.2. Adaptive Sampling Algorithm

The adaptive sampling algorithm provides a contrast to fixed-rate sampling as an alternative strategy in which the sampling rate of the sensor is varied according to the frequency content of the signal under study. The main aim in such a scheme is to reduce the energy consumption of the sensor, which is sampling the signal, as well as reduce the amount of sampled data to be transmitted. This can be shown in figure 4.2 for a signal whose frequency content is varying and there are either one of two ways to sample it with regards to sampling frequency.

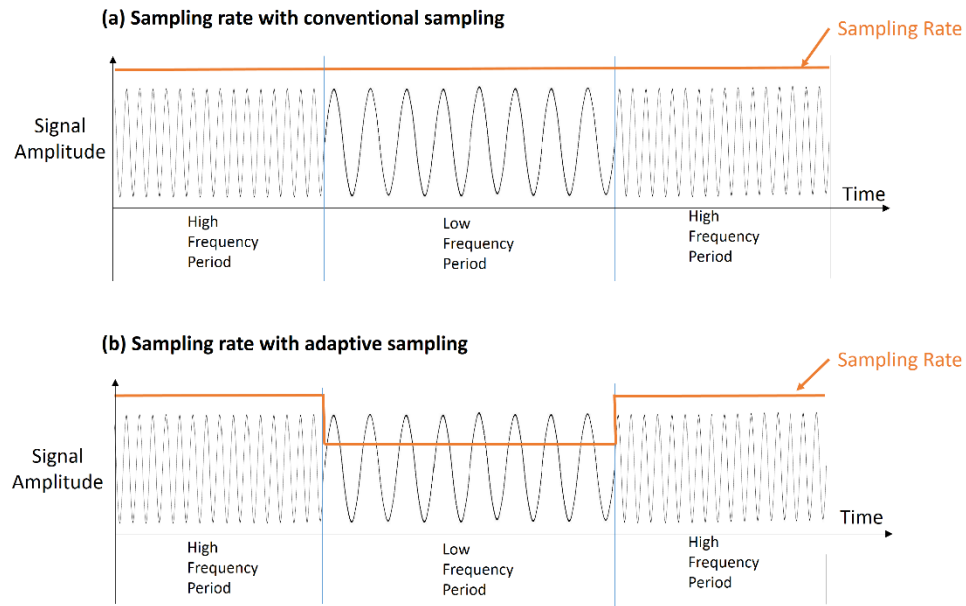


Figure 4.2: Sampling rate comparison between (a) a conventional sampling scheme and (b) adaptive sampling scheme.

Figure 4.2 shows the difference between conventional sampling and adaptive sampling scheme such that in an adaptive sampling scheme the frequency of sampling would adapt to the frequency of the signal. Since the frequency content of the signal is usually not known in cases where this sampling solution is to be applied, the usual method is to sample the signal at maximum frequency F_{max} for short time period and then sampled at the

calculated frequency for longer time, this is selected such that it is at least twice as high as F_h , which is the highest expected frequency in the signal for the system under study. From this information, we would then be able to select the desired sampling frequency-according to Shannon's sampling theorem. The periodic frequency re-calculation and sampling scheme can be best described by figure 4.3.

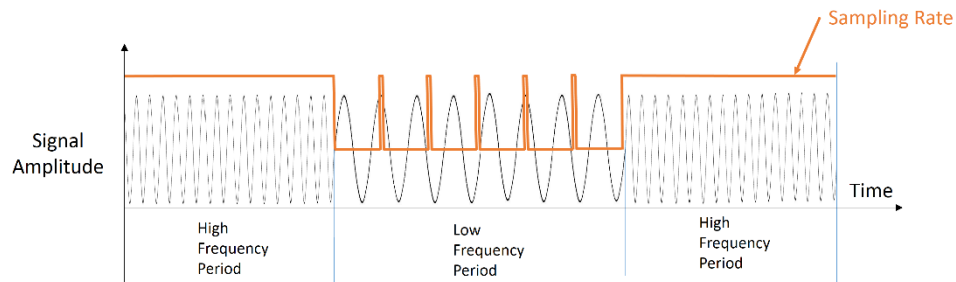


Figure 4.3: Adaptive sampling scheme visualization on a frequency variable signal.

In the literature, there are various methods that are used to calculate the sampling rate, including the wavelet transform which is the best to use since it does not require as much computation as the Fourier transform for finding the frequency content of the signal, and it has been applied extensively in microcontroller-based environments. After finding the frequency content of the signal, the sampling in subsequent sampling instances will be done at the detected frequency until it is deemed necessary to resample the signal. To sample the signals, a duty cycling scheme is implemented in which the signal will be sampled in discrete sampling intervals. This was described in the previous sub-section. For sampling purposes, using the high-energy sensor, the highest frequency F_h , in the signal will be calculated in the adaptive sampling scheme so that the sampling rate for sampling at the adjusted lower frequency can be calculated. Likewise Shannon's sampling theorem is shown (24) for a signal with frequency content at F_h and the confidence factor c , this is at least equal to 2 to satisfy the theorem.

$$SR = c \times F_h \quad (24)$$

Equation (24) would need to be satisfied if we are sampling signals using our algorithm. Since we do not know F_h beforehand, it will be estimated by sampling at F_{max} for a fixed sampling time and then calculating the highest possible contributing frequency in the signal, which is F_h . To decrease the effect of signal noise on the algorithm, the final sampling rate SR_{lf} (lower frequency) will be calculated based on the median of the last three readings, in a similar way used in the method of [112] for calculating the updated sampling rate of the lower frequency sensor. The authors of [112] used the maximum of the last n readings. It was felt that such a method will not be robust to random noise so a median of the last 3 readings will be used to calculate the updated sampling rate. The median is usually used because it is robust to outliers in the readings so for any case if the sampling rate calculated is very high or very low compared to the actual bandwidth of the signal, using the median of the last three readings would help select a nominal sampling frequency. This is shown by equation (25).

$$SR_{lf} = median(SR(k), SR(k - 1), SR(k - 2)) \quad (25)$$

Here k is the index of the high-frequency sampling instances at the node. To extract F_h from the sampled signal and set the sampling rate SR , a wavelet transform scheme is used which decomposes the signal into subsequent high and low pass bands to determine the band with the dominant signal characteristics, so that this band would be used to calculate the sampling frequency.

For the spectral decomposition process involved in the wavelet transform, both low-pass and high-pass filters are used. These are quadrature mirror filters (QMF) banks which are

constructed from orthogonal wavelet scaling filters [116]. Both the low-pass and high-pass filters have frequency responses of $G(\omega)$ and $H(\omega)$, respectively. The discrete-time impulse responses of both these filters are $g[n]$ and $h[n]$, respectively, and are applied in convolution form to the signal for wavelet transform computation as shown below:

$$\begin{aligned}
 x_{\beta+1}^{2\alpha+1}[n] &= \sum_j h[2n-j]x_{\beta}^{\alpha}[j] \\
 x_{\beta+1}^{2\alpha}[n] &= \sum_j g[2n-j]x_{\beta}^{\alpha}[j]
 \end{aligned} \tag{26}$$

In our algorithm, the wavelet transform will be applied to the highest selected sub-band in each level β such that the number of coefficients on which the transform will be applied will be reduced by a factor of 2 at each level, thus ensuring that the number of computations will be halved at each level of the transform. The algorithm used to calculate SR is described graphically in figure 4.4. Here the highest contributing band after each wavelet transform has been applied, is then selected for further decomposition, thus ensuring that no more than the necessary computations are done at each step.

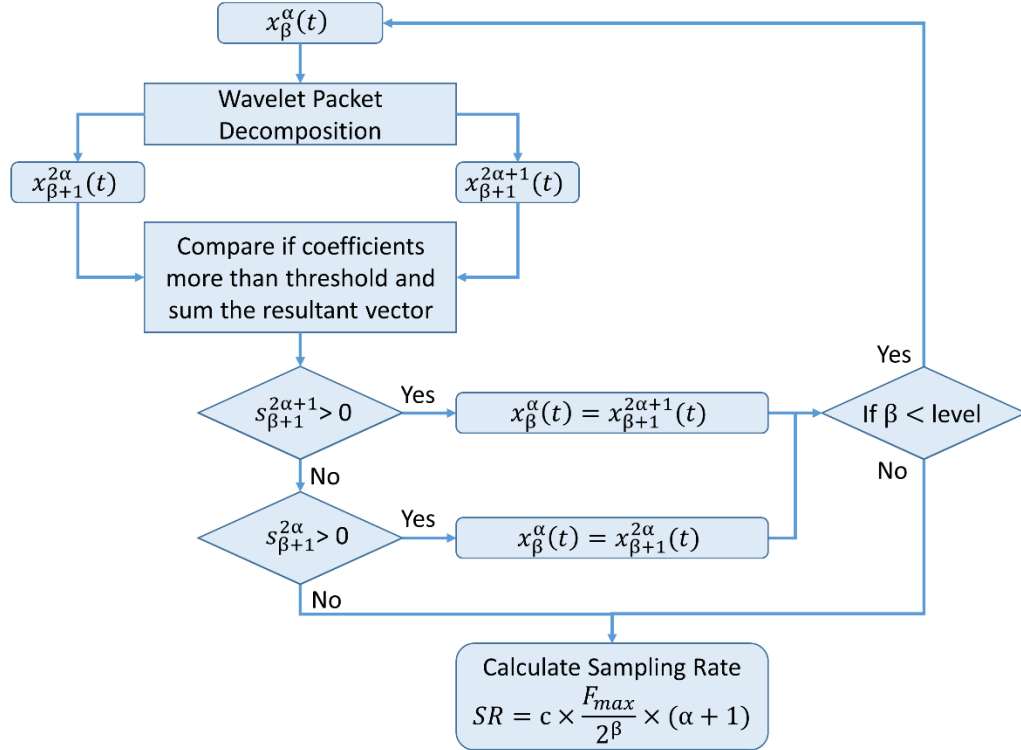


Figure 4.4: Scheme of calculating the adjusted sampling frequency.

In Figure 4.4, the first signal given to the algorithm will be the sampled signal $x_0^0(t)$. This will then be passed through a low-pass filter $G(\omega)$ and a high-pass filter $H(\omega)$ to produce $x_1^0(t)$ and $x_1^1(t)$, respectively. The algorithm will then select the band with the coefficients above the threshold and break it down into two more signals to be filtered using wavelet transforms. This way, only one wavelet transform is required per level and the signal length on which we will apply the wavelet transform would get reduced by a factor of 2 at each level. This process will go on until we reach the desired decomposition level or the magnitude of the coefficients in both bands is below the calculated threshold. The threshold is calculated using the formula described below [117]:

$$\gamma = \varepsilon \sqrt{2 \log m}$$

$$\varepsilon = \frac{\text{median}(\{x_1^1(t) | \forall t \in x_1^1\})}{0.6745} \quad (27)$$

Here m is the size of the data, and $x_1^1(t)$ is the result of the first high-pass filtered coefficients of the wavelet transform. Any coefficient below γ is set to 0 using the hard threshold operation and any coefficient above the threshold is set to 1, with the resultant vectors being then summed to give $s_{\beta+1}^{2\alpha+1}$ for the high-pass band and $s_{\beta+1}^{2\alpha}$ for the low-pass band. The hard threshold operation is described by the equation below [73]:

$$X_{th} = \begin{cases} x & |x| \geq \gamma \\ 0 & |x| < \gamma \end{cases} \quad (28)$$

This way the coefficients of each calculated sub-band are compared against the threshold in (27), if any coefficient is higher than the threshold for the sub-band then the sub-band is selected for further decomposition.

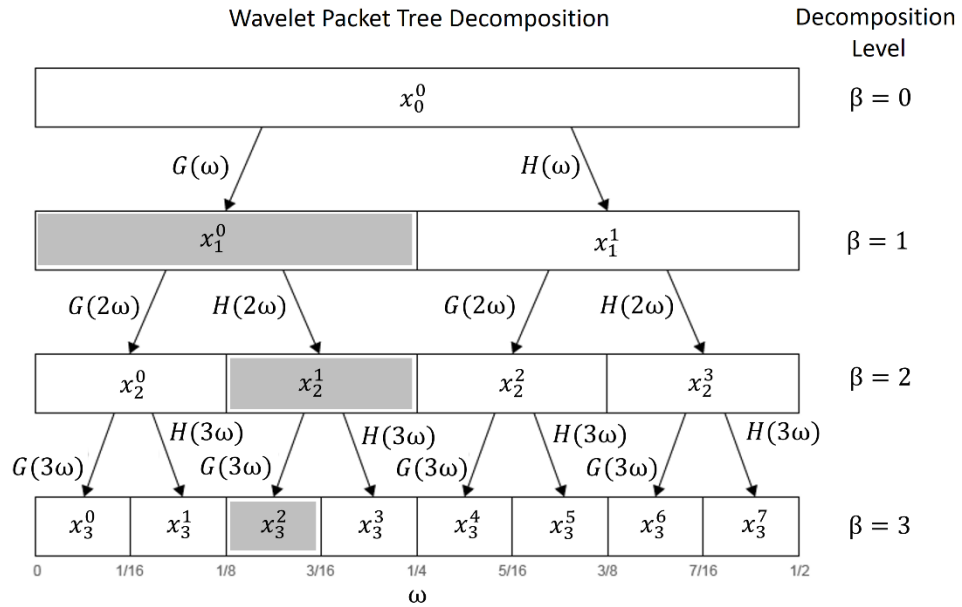


Figure 4.5: Example 3 level Wavelet Packet Decomposition and band selection according to the scheme presented in figure 4.4. The shaded boxes represent sub-bands which are higher than the calculated threshold and are selected for further decomposition.

The band selection and decomposition process is shown in Figure 4.5 for 3 levels. In the figure we are showing an example in which at level 1 the low pass filtered coefficients are selected because they are higher than the calculated threshold, after this step the high pass and low pass filters are again applied to this band and the resulting sub-bands are again compared to the threshold, if both bands have coefficients higher than the threshold then the higher band is chosen for the next filtering operation as shown in Figure 4.5, otherwise the lower band is chosen. This will go on till either the required level has been reached or the coefficients of the sub-bands are lower than the calculated threshold.

After the algorithm has finished running, the adjusted sampling frequency SR is calculated by the formula shown below:

$$SR = c \times \frac{F_{max}}{2^\beta} (\alpha + 1) \quad (29)$$

Here β is the highest decomposition level the algorithm achieved, F_{max} is the bandwidth that is covered due to sampling at SR_{max} rate, c is the constant which is greater than or equal to 2 as per Shannon sampling theorem, which is taken here to be 2 to satisfy the minimum requirements and α is the sub-band number with coefficients above the threshold at the decomposition level β . The corresponding computational complexity of the adaptive sampling algorithm is given below in expression (30) for the decomposition.

$$O(m) = \sum_{r=0}^{\beta} 2^{\lfloor \frac{m}{2^{r-1}} \rfloor} \times Q + Q(Q - 1) \quad (30)$$

Here m is the size of the data array on which the wavelet transforms are applied, β is the decomposition level and Q is the size of the filter array used for the wavelet transform. The

filter array size varies from 2 to upwards of 40 coefficients for some transform pairs, these would correspondingly increase the computation complexity. Computation complexity is directly proportional to the energy consumption.

Along with setting the sampling frequency using the adaptive sampling algorithm described above, we will also be trying to detect the leak using the vibration sensors while using the adaptive sampling approach. This approach is explained in the next subsection.

4.3. Adaptive Threshold Algorithm

To detect leaks in a dynamic environment of a pipeline network using vibration sensors an adaptive threshold based algorithm is needed as the value of the vibration will be changing dynamically based on the pipeline conditions. For detecting the leak in the pipeline using vibration sensors there are various ways in which the threshold can be applied to distinguish between different conditions, [25] used the standard deviation of the vibration signal to determine if leak is occurring in the pipe, likewise [118] used the raw vibration readings but based the leak detection on the change in vibration magnitude, also the data was being transmitted in real time to the sink. In [9] the authors used the threshold calculated at the sensor node to detect a burst event using the low energy sensors and then confirm it using high energy sensor by using the vibration median. There are many methods to which leaks can be detected in the pipeline as shown in the literature, and these are adaptive based on the vibration conditions.

We have two vibration sensors as described previously, one of which is a high-precision sensor (HSVS) and consuming more energy and the other, being a low-precision sensor (LSVS), and hence consuming less energy. These are used in an adaptive threshold based

scheme [9], [31] which was previously used to detect bursts in pipelines. A hierarchical scheme is used in which LSVS are used for extended time periods to detect leaks and then wake the HSVS for event confirmation. The median of the vibration signal at these high frequency sampling instances will be needed to detect the leak, when the median is higher than the previous readings a leak is most likely present in the system.

For the LSVS, we will be using the median of the signal after performing the wavelet packet decomposition (WPD) of the acquired signal as in [119] where the authors used the same technique to distinguish between different events in the pipe when using vibration sensors. WPD would divide the signal into different bands of equal bandwidth which would allow to compare the medians of the individual bands, this method would give a better metric for leak detection as the leak signal would have a higher energy in some bands of the signal as the leak signal is not broadband in nature.

When using a HSVS, the median would be calculated without any additional operation as the HSVS has very low noise characteristics and would recreate the signal with fewer distortions. The median of the signal would be calculated at each HSVS sampling duration and LSVS sampling duration separately and compared to the respective previous ones to detect leaks in the system. The median of the LSVS signals would be reset after each frequency readjustment period because a new sampling frequency would be calculated which may not be equal to the last one. Using these steps, the final algorithm is shown in Figure 4.6 below.

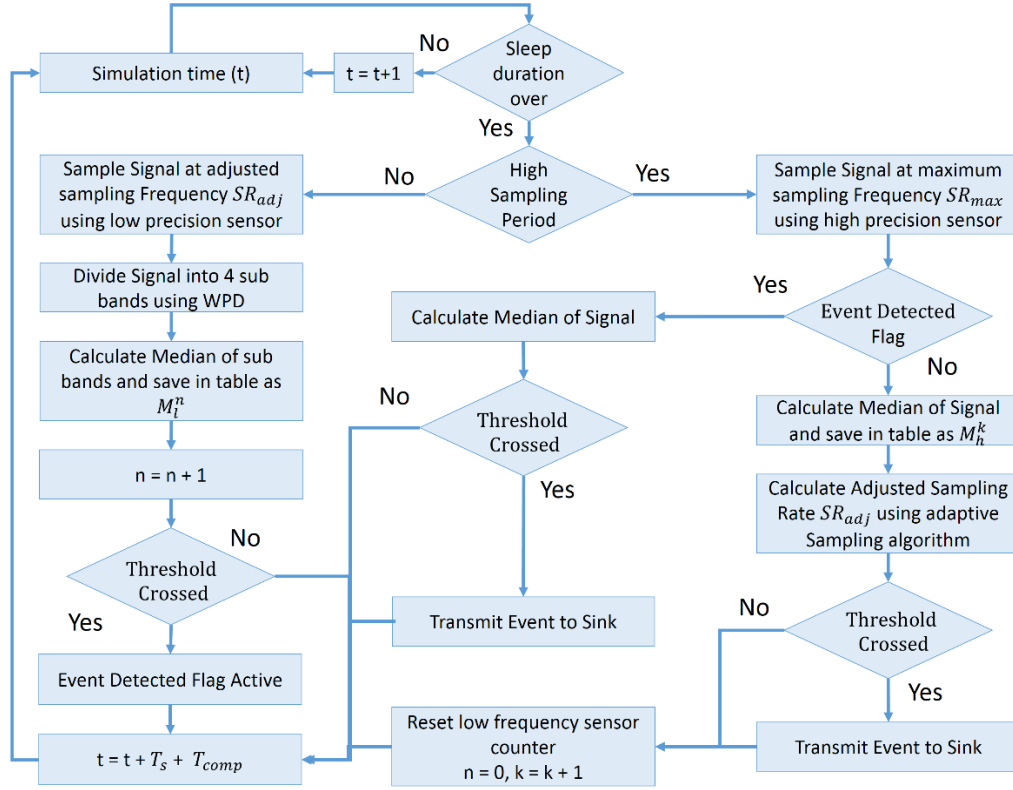


Figure 4.6: Complete Scheme of the simulation for leak detection using adaptive threshold

Figure 4.6 shows the complete scheme that will be used to run the algorithm. The first step is running the simulation with time step t and checking if the sleep duration is over for the duty cycling scheme, and if so, the algorithm checks whether to sample using the HSVS or using the LSVS. If the time is for a HSVS, the sensor samples at SR_{max} . It then calculates the median of the signal M_h^k and compares it to the previous medians. If it is higher than the previous two medians by a factor of η_h , it then transmits an event to the sink that it has detected the leak. The equation for the comparison is shown below (31):

$$Leak = (M_h^k \geq \eta_h \cdot M_h^{k-2} \text{ and } M_h^k \geq \eta_h \cdot M_h^{k-1}) \quad (31)$$

The next step is to calculate the adjusted sampling frequency SR using the adaptive sampling algorithm described in Figure 4.6. The algorithm then resets the counter for the

adjusted frequency sampling sensor and increments the simulation timer counter by the sampling period T_{sample} . This process is different on the LSVS side where the signal is sampled using the low-frequency sensor and then divided into 4 sub-bands using the WPD algorithm. The algorithm passes the signal through one bank of low-pass and high-pass filters like the ones shown by equation (26) and generates one signal with low-pass coefficients and one with high-pass coefficients. These are then passed individually through the filter bands again to generate the low-pass and high-pass components and to generate 4 separate bands of the signal. The medians M_l^n of each sub-band are computed and are compared against the previous medians as shown by (32) and again if they are found to be higher in any sub-band compared to the first reading by a factor of η_l then the high frequency sensor is used to confirm the event by comparing the current and previous high-frequency medians with each other, otherwise the low-frequency sensing routine counter increases by 1 and adds the algorithm computation time to the simulation time and then goes back to the start of the loop.

$$Leak = (M_l^n \geq \eta_l \cdot M_l^2 \text{ and } M_l^n \geq \eta_l \cdot M_l^1) \quad (32)$$

The algorithm computation time is dependent on the operations that the processor does, for the high-frequency sampling routine, the operations of the algorithm include the adaptive sensing algorithm and finding the median, and for the low-frequency sampling routine, the operations include WPD and median calculation operations. The median calculations is done by sorting the values using a mergesort algorithm [120] which is computed in $O(m \log m)$ time. And the WPD takes the computations shown by equation (33)

$$O(m) = R \times 2 \left[\frac{m}{2^{-1}} \times Q + Q(Q - 1) \right] \quad (33)$$

Here R is the number of levels the signal is decomposed into, in this case it is 2 and m is the length of the signal and Q is the length of the filter.

Along with this adaptive sampling routine, a monitoring solution will also be implemented in a parallel loop, which would transmit the data back to the sink even if there is no leak detection in the system. This would allow further data processing at the backend and would help in leak detection by applying more computationally expensive procedures on the gathered information if for some reason the proposed monitoring solution is unable to detect the leak. Before transmission of data the signals that are acquired by the onboard sensors will be processed by the compression algorithm, this is described in the next subsection.

4.4. Data Compression Scheme

Data compression is used to reduce the amount of bits to represent a certain amount of data. This is beneficial both in a sense that for storage purposes and for transmission purposes the amount of data required to represent a signal gets lesser than usual. Mostly for WSN based systems there are many different types of compression algorithms present in literature which are mostly based on the two broad categories of lossless compression and lossy compression. Lossless compression mostly is focused on retaining all the information in the signal while reducing the size, on the other hand lossy compression usually has an acceptable distortion of the signal from the original while reducing the size more than what lossless compression methods are usually able to achieve, it should also be noted that while both of them are computationally extensive, lossless compression methods use less computations but in the case of vibration signals provide less compression than the lossy compression methods [68], this is important because when transmitting the more

compressed signal the energy saved is higher as the transmission is a more energy intensive process than the computation. We can, from an energy standpoint, afford to use more time on computation as opposed to more time on data transmission.

Regarding lossless compression methods there are algorithmic approaches like Huffman coding [121] in which the most occurring variable values are represented by the least amount of bits and the least occurring values are represented by the higher amount of bits, it is a form of variable bit length coding and there are many methods which exploit this while using the entropy of the signal. Using the entropy of the signal is most efficient in data where the signal is not varying very much and the difference of consecutive values is concentrated around zero. Dictionary-based Huffman coding is used in these scenarios [69] as well as adaptive dictionary based methods with multiple dictionaries to maximize the compression [122] around other values where the entropy of the signal may not be concentrated around zero. Again, these methods are effective only when the entropy of the signal is low and in vibration compression [68] it didn't give better performance compared to lossy algorithms. Usually these methods are combined in an algorithm where both lossy and lossless compression methods are combined to give a better result than any one method alone.

When discussing lossy signal compression, there is extensive work done in compressing ECG signals [123]. There are many methods that the authors have used to tackle the problem of signal compression while compressing the signal as much as possible while achieving the least amount of distortion possible. The most used method is the wavelet transform when compressing signals as it is computationally relatively simple compared to other transform-based methods like the discrete cosine transform [68] and the lapped

orthogonal transform [70]. Applied alongside the wavelet transform, in most of the works, entropy coding and quantization methods, are also applied. It was noted that most works focused on decomposition levels between 4 and 6 and that Wavelet Packet Analysis was not better compared to Wavelet Transform for the compression purposes as most of the frequency content of the signal is in the lower bands and the Wavelet Transform already does an effective job of dividing the signal into sub-bands. Most of the techniques in literature were focused on techniques which combined two or more techniques to compress the ECG signals, for example [124] removed a certain percentage of coefficients below a threshold and then quantized the signals using a fixed 8 bit quantization. In a similar fashion, [125] thresholded and quantized the wavelet coefficients using a different number of quantization bits and compared the change in compression ratio (CR) based on a user-defined Percentage Root Mean Difference (PRD), which are two common methods to measure compression performance. Other methods used energy packing schemes [126] to reach a particular CR and PRD to set the threshold. Some other ways to improve the CR is by employing entropy coders like Huffman coding [124] and arithmetic coding [127] to reduce the size on the thresholded wavelet coefficients. Run-length coding [126] and significance map [124] are other methods used very frequently to increase the CR. Many different quantization techniques are also available in literature that aim to set the optimum number of bits to have the least PRD with the best CR, most of the works use a fixed quantization scheme or try to reduce the number of bits near zero to compress the signal. Sub-band adaptive quantization (SAQ) [128] is another method which is used to compress signals based on the variances of the different sub-bands.

In vibration signals, the simple Wavelet Transform thresholding technique [70] is also used to compress the signal of plane engines where it was found that at high compression ratios the wavelet transform does not perform as well compared to the Discrete Cosine Transform and the Lapped Orthogonal Transform with adaptive quantization like the one described in [128], the authors had used quantization with the lapped orthogonal transform to compress the vibration signals. The Discrete Cosine Transform [68] is also successfully used to compress vibration signals and it mostly gives a better compression performance than wavelets based methods for stationary signals in both the compression ratio and error in signal. This was compared against entropy-based methods [69] which were optimized for use in WSNs and was shown to be 10 times computationally expensive compared to it, but the data compression performance was better and the data transmission time was also less so it was better overall in terms of energy consumption. The downside is that the DCT is computationally expensive [129]. In [130] the authors introduced a method to compress vibration signals based on a combination of wavelet transform, adaptive quantization and arithmetic coding that gave a CR of up to 27 times that of the original signal with acceptable signal distortion. The adaptive quantization scheme is described in detail in [128] and is based on the variance of the coefficients of the wavelet transform. In [131] the author has introduced a method for choosing the most optimal wavelet coefficients using a Genetic Algorithm to reduce the Mean Squared Error (MSE) in the signal. While such a method will give the most optimal coefficient given certain restraints and will work better than the simple truncation of coefficients that is mostly used based on magnitude and energy methods it is more difficult to implement in an energy-constrained environment of WSN

nodes as Genetic Algorithm is an artificial intelligence based search method for the most optimum parameters for any given cost function.

Based on the literature, a wavelet transform based approach will be used with sub-band adaptive quantization and an algorithmic approach to reduce the number of bits used to represent information. This is discussed in the next sub-section.

4.4.1. Algorithm Details

The proposed algorithm works in 3 steps, first the wavelet coefficients are thresholded according to the percentage of the absolute sum of the coefficients, then the remaining coefficients are quantized using a sub-band adaptive quantization scheme. The third and final step is the second main part of the compression in which the significance map is generated, this indicates the location of the non-zero coefficients in the transform and allows us to only transmit the non-zero coefficients. After this step, the significant coefficients are also coded using a Rice coding scheme. This will be described later. The algorithm procedure is shown in figure 4.7.

Signal Compression Schemes

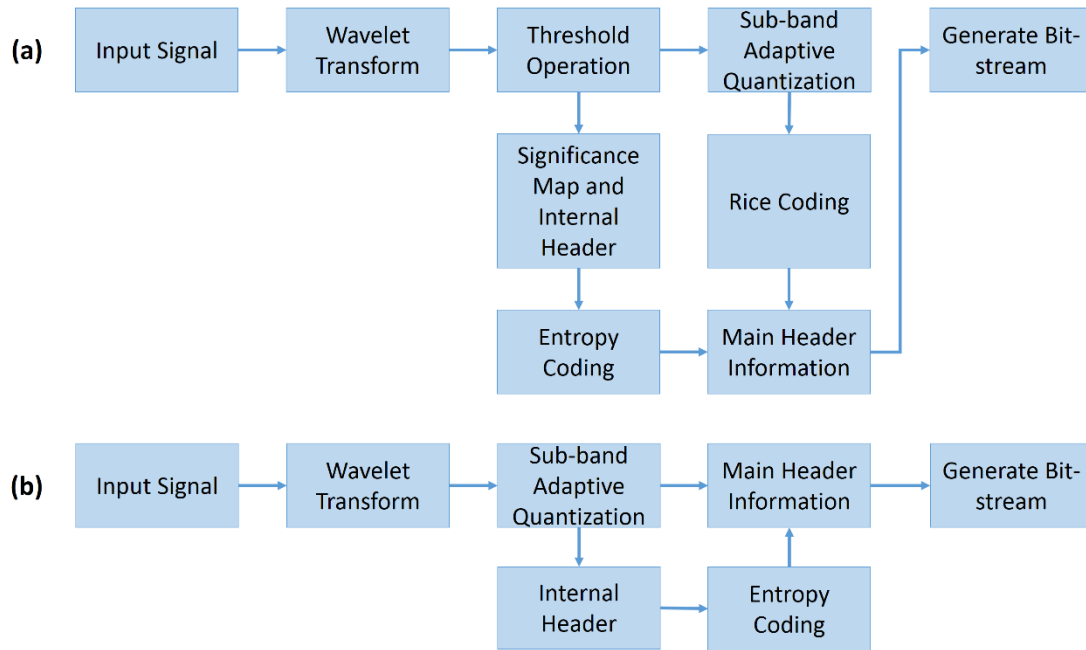


Figure 4.7: (a) Compression and Encoding Scheme for HSVS and (b) for LSVS at WSN Node.

In the compression scheme the daubechies 5 wavelet is being used in the wavelet transform. Similar to the procedure shown in figure 4.7 an inverse process will be applied to the received bit-stream at the sink to decode the signal. This is shown in figure 4.8.

Signal Decoding Scheme

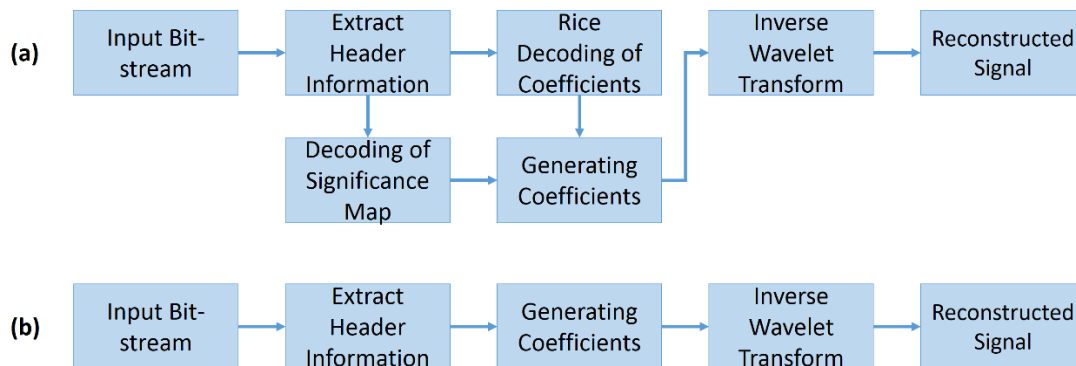


Figure 4.8: Decoding procedures for (a) HSVS sampled data and (b) LSVS sampled data.

In figure 4.7(a) looking at the first step we will apply the wavelet transform. Here the filter banks used are similar to the ones used in WPD for the adaptive sampling algorithm, the only difference is that only the low-pass filtered coefficients are passed through the wavelet transform iteratively to get the result. For example if there are 3 levels in the wavelet transform the result of the coefficients from the original signal would be like the one shown in figure 4.9.

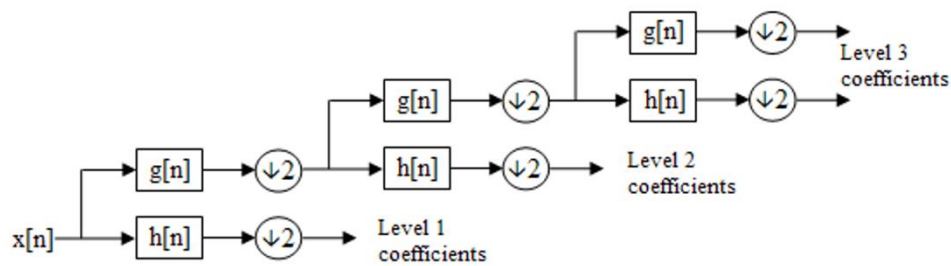


Figure 4.9: 3 level Wavelet Transform of signal $x[n]$.

This way for the signal, however many levels of decomposition are for the wavelet transform, there would be that many detail coefficient vectors of the signal and one approximation coefficient vector of the signal. After this step, the threshold operation will be applied to the signal, the threshold operation in this case would be based on the magnitude of the signal, [131] did a comparison of the number of retained coefficients and the mean squared error (MSE) of the vibration signal and found out that the relationship between them is logarithmic in nature, as such in literature there are many ways to define the threshold, [68] based the threshold on the energy of the coefficients that 95% of the energy of the signal is retained. Another work [70] did a comparison of different levels of compression by varying the amount of coefficients selected among other parameters, it was again found that more coefficients yielded better results. In [73], the authors discussed that over 90% of the wavelet coefficients are less than 5% magnitude of the highest coefficient

value. Based on the light of the literature, we have proposed a simple to implement algorithm which would select the threshold level based on the magnitude of the signal while being computationally simple, first the coefficients are generated from the signal by applying n-level wavelet transform, in our case we have done 5 level wavelet transform so there will be 5 detail coefficient vectors and 1 approximation coefficient vector in the wavelet coefficient vector WC .

$$WC = (A_5, D_5, D_4, D_3, D_2, D_1) \quad (34)$$

This step will take up the computational complexity shown by equation (30) for the high frequency signal, but with the first decomposition vector already done by the adaptive sampling algorithm and saved and the subsequent decompositions being done by the signal compression scheme.

The magnitude of all the vector elements are summed together as shown by the equation below.

$$M_T = |A_5| + |D_5| + |D_4| + |D_3| + |D_2| + |D_1| \quad (35)$$

From this we can apply a value for example we need enough coefficients that make up $Z\%$ of the signal, we can say this value is the threshold so the threshold will be.

$$Magnitude_{threshold} = \frac{Z}{100} \times M_T \quad (36)$$

This will usually be lower than the number of coefficients required to complete this particular energy requirement because energy is squared of the coefficient value. After this step the magnitudes of the coefficients are arranged in one single vector and the number of coefficients selected is based on the cumulative sum of the arranged vector.

$$V_{sort} = \text{sort}(|A_5|, |D_5|, |D_4|, |D_3|, |D_2|, |D_1|) \quad (37)$$

The sorting in (37) is again done by the mergesort algorithm of $O(m \log m)$ complexity [120].

Then the coefficient value is selected by the following algorithm. Here the coefficient below which all the remaining coefficients will be zeroed is chosen based on the magnitude threshold defined above in (36).

while $i < \text{size}(V_{sort})$

if $V_s < \text{Magnitude}_{threshold}$

$$V_s = V_s + V_{sort}[i]$$

$$i = i + 1$$

else

$$T_c = V_{sort}[i]$$

endif

endwhile

From the above algorithm all values below the magnitude of T_c in the original coefficient vector WC are set below zero using the hard thresholding operation defined in (28) to generate $NZC(A_5)$ up to $NZC(D_1)$. After this step since all the sub-bands will have varying magnitudes as well varying amounts of significant coefficients (coefficients which are non-zero after the threshold operation), sub-band adaptive quantization [128] will be applied to the vectors containing the remaining elements. The wavelet coefficients are already

ordered into their respective sub-bands, the method described in [128] calculates the variances of each sub-band and then distributes the available bits to each sub-band, a variation of the algorithm was used to calculate the bits based on simplifying the computations and is shown below:

1. Compute Ra_k for each sub-band of the wavelet transform, where Ra_k = range of the wavelet coefficients in the k_{th} sub-band.
2. Set $R_k = 0$ for all k and let $R_b = M.R$, in which R_b = total number of bits available for distribution; R =desired average number of bits per sample to be used; and M =total number of sub-bands in the wavelet transform.
3. Sort the ranges $\{Ra_k\}$. Suppose Ra_q is the maximum.
4. Increase R_q by one and divide Ra_q by two.
5. Decrease R_b by one. If $R_b = 0$ then stop; otherwise, go back to step 3 and repeat.

In the algorithm shown above the bands with the highest ranges are allocated the highest number of bits and so on. This way it is ensured that quantization error is minimized for bands with high ranges. This differs from the original method that range is used instead of variance, since calculating variance is an extensive computation requiring multiple steps, this approach is proposed in which instead of calculating the variances the range of the sub-bands are calculated. This way the computation cost will be minimal and we will get approximately the same result. Apart from this change the original bit allocation algorithm works in the same way as shown in the algorithm above. After the bits have been assigned, we will quantize the values of each sub-band based on the maxima and minima of the signal by the formula shown below:

$$NZC_q(A_5) = (2^q - 1) \frac{NZC(A_5) - NZC(A_5)_{min}}{NZC(A_5)_{max} - NZC(A_5)_{min}} \quad (38)$$

Here $NZC(A_5)$ are the non-zero coefficients that were generated from the thresholding operation for sub-band A_5 , $NZC_q(A_5)$ are the quantized coefficients for sub-band A_5 and are quantized for q bits. $NZC(A_5)_{min}$ and $NZC(A_5)_{max}$ will be different for each sub-band as each of them have different ranges, here we have taken values for A_5 . This will ensure most of the information is coded as near to the original value as possible for each sub-band because setting the range of the highest values for the sub-bands, which will usually be the approximation vector will distort the information for sub-bands which would have an entirely different range. This step takes up to 4 computation cycles for each coefficient in each sub-band. After this information has been generated the sub-bands will be Rice coded [132]. This is a lossless signal compression technique which reduces the amount of bits used to represent the data.

To apply this scheme first we will find out the median value of the sub-band and see how many significant bits it uses, after that we would assign that bit number to be used for coding the sub-band. This bit value q_r will be usually less than q , which were the bits originally used to quantize the signal using (37). After we have determined q_r from the median we will then encode the coefficients of the vector using rice coding. This is done by the algorithm shown below as an example for sub-band A_5 .

find $m = median(NZC_q(A_5))$

find maximum bits q_r required to encode m

while $i < \text{length}(\text{NZC}_q(A_5))$

$$j = \text{floor}(\text{NZC}_q(A_5)[i]/2^{q_r})$$

write j binary 1's, one binary 0 and add last q_r bits of $\text{NZC}_q(A_5)[i]$.

endwhile

In this algorithm again the median is found by applying mergesort [120]. And each step takes up to 5 computation cycles for each nonzero coefficient.

From the above algorithm we will be able to generate the rice coded bit-stream $A_{5,r}$ for the $\text{NZC}_q(A_5)$, for example 1000 requires 10 bits to encode using normal bits shown like 1111101000, but using for example 8 bits to encode it we will write it down like 111011101000, this way values higher than the median would require more bits than usual to encode but if there are more values below the median then it results in a net gain in terms of bit conserved, as they will be encoded in 8 bits rather than 10 bits throughout the stream. After rice coding the coefficients for each sub-band and finding the maxima and minima for each of them we will be able to generate the final bit stream that is to be transmitted, for this we would need the significance map as we are only transmitting the significant non-zero coefficients of the signal, it is simply the length in bits of the whole vector, for example if there are 1000 coefficients then the significance map will be 1000 bit long and will contain 1 where the coefficient is non-zero and 0 where it is below the threshold. Then the significance map will be appended with the sub-header in the beginning which will

indicate how many bits are required to encode each sub-band and the maxima and minima of each sub-band. This is shown in the expression below:

$$SubHeader = (q_r(A_n), q_r(D_n), \dots, q_r(D_1), \max(A_n), \min(A_n), \dots, \min(D_1)) \quad (39)$$

Here q_r are each of 4 bits each and show that up to 16 bits can be assigned to each sub-band using the SAQ algorithm described previously, and the maxima and minima for each sub-band are of 24 bits as per the ADC resolution and settings. This sub-header and the significance map will be appended in one bit-stream which will then be divided into bytes and coded using a fixed dictionary Huffman by the encoding procedure described in LEC [69]. This operation takes an approximate 57.86 instructions per byte. Another header would then be appended to this sub-header which would have the lengths of each of the components of the whole stream which comprises the LEC coded sub-header stream and the significant coded coefficients of the sub-bands. The final output of the whole compression scheme would look like the one shown below:

$$BitStream = (SubHeader_{length}, A_{n_{length}}, \dots, D_{2_{length}}, SubHeader, A_{n_r}, \dots, D_{1_r}) \quad (40)$$

In the *BitStream* the $SubHeader_{length}$ is of 16 bits, and all the other lengths are 10 bits each, $D_{1_{length}}$ is not required because it is the last and remaining vector of the whole bit-stream.

The scheme for the low energy sensor would be similar but there would be no thresholding operation so no significance map would be required, so no dictionary based Huffman coding will be applied to it. Only sub-band adaptive quantization will be applied and since the signal will already be of low resolution and low length no Rice coding will be applied either. This way we would be able to reduce the amount of data to be transmitted by the

high energy sensor drastically but also reduce the amount of data that would be transmitted by the low energy sensor routine while keeping acceptable signal distortion rates. The signal distortion is shown by Percentage Root Mean Difference (PRD) which is commonly used to gauge compression algorithm performance apart from the compression ratio. PRD is given below as:

$$PRD = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{\sum_{i=1}^N (x_i)^2}} \times 100 \quad (41)$$

Here x_i is the original signal and \hat{x}_i is the reconstructed signal. We will be calculating it in the simulation for different compression ratios as it would give us an approximation as to how much distortion there will be for a certain compression ratio, the compression ratio is given by:

$$CR_{high_rate} = \frac{bits \times signal_{length}}{16 + (n + 1) \times 10 + length(SubHeader) + k} \quad (42)$$

$$k = q_r(A_n) \times length(A_{n_r}) + \dots + q_r(D_1) \times length(D_{1_r})$$

The expression for CR_{low_rate} will be similar with k similar to the expression shown below and no other value in the denominator as only SAQ will be applied.

$$k = q(A_n) \times length(NZC_q(A_n)) + \dots + q(D_1) \times length(NZC_q(D_1)) \quad (43)$$

The process of decoding the stream will be similar but in reverse like the one shown in figure 4.8, first examine the header and extract the lengths and then go in reverse and reconstruct the coefficients and then go on to apply the inverse discrete wavelet transform to get the reconstructed signal. Before this could happen the encoded and compressed data

will be needed to be transmitted. For this the data transmission scheme is described in detail in the next subsection.

4.5. Data Transmission Scheme

For transmissions purposes, since we will not be transmitting at every cycle of the adaptive sampling algorithm, a different routine will be chosen. This can be at every hour or when the sink requires network information regarding monitoring data. Monitoring data in this case will be the pressure data which will be acquired for monitoring purposes and will be measured at a comparatively low rate of hourly or half hourly periods. The radio will be using significant energy only when pressure data reporting time will be present, and in such a transmission we would transmit all the medians of the LSVS and HSVS and the frequency at which the vibration sensors were sampling, along with the high-frequency vibration sensor and low-frequency vibration sensor sampled data. If the pressure monitoring period is every 30 minutes, then the transmission would include the pressure reading and the data sampled by the HSVS and the LSVS in the previous 30 minutes, along with the medians of the HSVS reading and the medians of the LSVS along with the sampling rates. Such a transmission scheme is described in Figure 4.10.

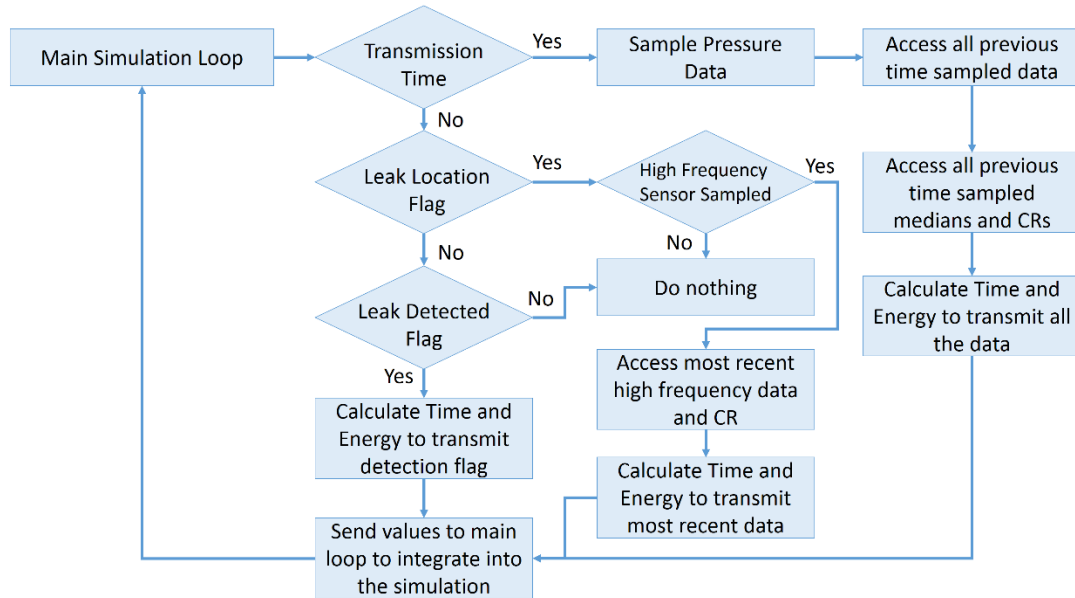


Figure 4.10: Radio Transmission Scheme for the overall simulation.

In figure 4.10 there are 3 cases described for the transmission conditions, first if the duty cycle time for the transmission time is up and it needs to transmit the data, second if the radio has received the leak location flag and need to transmit data and third if the radio needs to transmit the leak detected flag. For all the three cases the radio transmissions are going to be different, in the first case, all the time stamped compressed vibration data between the previous radio transmission time and current radio transmission time will be transmitted along with the time stamped pressure data sampled at that time. In the second case, only the most recent high frequency sampled data will be transmitted, since the nodes are synchronized, all of them would sample the high frequency sensor at the same time when not detecting any leaks, this would have them send the most recent high frequency sampled data as the nodes would receive transmission if the sink requires data to locate leaks. The third and final condition will be when the node will be transmitting leak detection flag to the sink, this would happen at any sampling instance if the conditions of the threshold based leak detection algorithm have been satisfied. For all three conditions

the amount of data transmitted would be different, for the last case only the detection flag need to be sent so it wouldn't require more than one packet of information so it is relatively quick to transmit that, the second case only requires the compressed and time stamped high frequency data so it will also be quicker than usual and the last case requires all the data to be transmitted between the previous and current transmission time. An expression for the approximate number of bytes B transferred in the transmission time is shown by the expression below:

$$\begin{aligned}
B = & \sum_{i=1}^N 3 \times \left(Tstamp_h[i] + \frac{Sample_h[i]}{CR_h[i]} + M_h[i] \right) \\
& + \sum_{i=1}^M 3 \times \left(Tstamp_l[i] + \frac{Sample_l[i]}{CR_l[i]} + 4 \times M_l[i] \right) + 3 \times Sample_p + 3 \\
& \times Tstamp_p \tag{44}
\end{aligned}$$

In the expression above everything is multiplied by 3 because the ADC gives data in 24 bit resolution, which is equivalent to 3 bytes. N is the number of times the high- frequency vibration sensor has been used to sample the vibrations, $Sample_h[i]$ is the sample vector of the high frequency sampled data and $CR_h[i]$ is the corresponding compression ratio, using both of these we can find the size of the coded coefficient bit-stream that we will be transmitting, $M_h[i]$ is the median that will be calculated for each high-frequency vibration sampling instance and $Tstamp_h[i]$ is the corresponding timestamp. Similar is the case for the low frequency data where M is the amount of vectors for the low frequency data.

From the number of bytes, we can then find out the transmission time T_{tx} and thus calculate the energy used by the radio for transmissions and T_{radio} , this was previously mentioned

in chapter 3 while discussing the energy model for the WSN node. Apart from this, the nodes would be receiving data every resampling time in the adaptive sampling algorithm and when the sink has confirmation from only one node in the network for the leak and would be requiring more data from other nodes to perform cross correlation to localize the leak. Other than this, the nodes would not be receiving data and will transmit only for 3 possible conditions, when transmitting data for periodic monitoring, event detection and when specific data has been required from them for leak localization.

This completes the discussion of the multi-faceted energy conservation scheme that we have developed for monitoring leaks using both vibration and pressure data in pipelines. The next section discusses the development of the cost function related to the proposed scheme.

4.6. Solution for Optimal Cost

Related to the cost analysis of the scheme there are different parameters in the algorithm that can be set which would give different distortion rates as well as different leak detection and localization times for any number of different parameter combinations. In an attempt to consolidate all of this information regarding the cost of the algorithm in terms of energy, time and distortion a cost function was developed which would give the conditions for the most appropriate parameters based on the information. There are many ways to approach a certain problem when solving it using a cost function [133] and in this scenario the promptness at which the leak is detected and located is under study along with the distortion in the signal. These are the three parameters which will be optimized as per the proposed scheme keeping in mind the energy cost of the algorithm. For calculating the energy cost the equations are already given in chapter 3 of the thesis in the WSN Node Model section.

This section further builds upon the equations given there and the corresponding costs of the adaptive sampling and signal compression algorithms in the following chapter. Before implementing the solving method some initial parameters of the node will need to be known, for example the controller type, its corresponding power consumption, the sensor types and their corresponding power consumption and finally the radio type and its respective power consumption. After these three things are known it would also be required to know the wavelet filter which is being applied to the algorithm, this is because the length of the filter varies and it usually takes a certain amount of time to run calculations based on the filter length as the number of computations is directly proportional to the filter length, also the signal length is to be known because both energy conservation algorithms are applied on the signal and again the longer the signal the higher the computations. Using equation (30) the computation cost can be evaluated for the high frequency sensing cycle for the adaptive sampling and signal compression scheme and using equation (33) the computation cost can be evaluated for the low frequency cycle that uses the WPD for signal compression as well as the threshold computation. The computation time in the high frequency cycle will set a limit to how low the minimum duty cycle time T_t can be set as the computations are higher in this cycle. This computation is shown by the expression below:

$$C_c^h = C_{CR} - 2 \left[\frac{m}{2^{-1}} \times Q + Q(Q - 1) \right] + \sum_{r=0}^{\beta} 2 \left[\frac{m}{2^{r-1}} \times Q + Q(Q - 1) \right] \quad (45)$$

Here C_{CR} are the different computation operations that are executed in the signal compression scheme, for C_c^l for the LSVS sampling the expression will be similar as only WPD is being performed.

$$C_c^l = C_{CR} + \sum_{r=0}^2 2 \left[\frac{m}{2^{r-1}} \times Q + Q(Q-1) \right] \quad (46)$$

The corresponding time T_c^h can be found by first finding the energy consumption of C_c^h by using equation (17) and then dividing by the power consumption of the controller to find the time, the case is similar for T_c^l . This is shown by the equation below for the HSVS.

$$T_c^h = \frac{C_c^h \times \mu \left(\frac{f}{k} + \beta \right)}{e_{comp}^w} \quad (47)$$

Finding T_c^h will give us the minimum duty cycle time the overall scheme can achieve as this the approximate time taken for the algorithm to complete the computations for the given sensing period and wavelet filter type employed in the adaptive sampling algorithm and signal compression algorithm. The time T_c^l can be calculated similarly. After this has been determined the energy consumption of the node for each high energy sensor sampling cycle and low energy sensor sampling cycle is calculated, this is based on the sensor energy, computation energy and radio energy and the active and sleeping times of the nodes under study, it is explained in detail in Chapter 3.4. Using equation (16) E_j^h is the complete energy consumption of the HSVS sampling period for both the active and sleep time of the sensor and E_j^l is correspondingly for the LSVS, both of these are a function of sampling time, duty cycle time and active time of the controller, the active time is shown in equation (48). Using the energy consumption for HSVS duty cycle period and for LSVS duty cycle period the following expression is constructed for one complete resampling cycle of the adaptive sampling algorithm.

$$E_j^{adap} = E_j^h + (n-1) \times E_j^l \quad (48)$$

During the HSVS sampling cycle the radio will also receive transmission request through radio and will either transmit data or not transmit based on the information. The adaptive cycle j is made of n cycles of sampling after which the algorithm needs to resample. Consequently E_j^{adap} is a function of T_c^h , which is time for computation for HSVS cycle, T_c^l , which is for LSVS, T_t , which is time for one duty cycle duration (22) and n , which is the resampling time for the adaptive sampling algorithm, this can be written down in the form shown below:

$$E_j^{adap} = f(T_c^h, T_c^l, T_t, n) \quad (49)$$

Every i seconds the radio will be transmitting the data acquired in the previous i seconds as per equation (44). From this the radio energy consumption will be computed using equation (18) and the radio transmission time as per equation (21). This will give a basis for average energy that the algorithm uses in one transmission cycle and using these equations for the whole simulation time this will give a cost for the energy required for the whole simulation. For E_{radio}^w from (18) the energy consumption is proportional directly to the number of bytes transmitted, which is directly proportional to T_{sample} , Sampling rate SR_{lf} from equation (25), transmission time i and CR_{TH}^B , which is the B number of bits for SAQ and TH threshold for the compression scheme. So E_{radio}^w can be written as.

$$E_{radio}^w = \sum_k E_k^{radio} = f(T_{sample}, SR_{lf}, CR_{TH}^B, i) \quad (50)$$

Moving on the final expression is constructed for the energy consumption for both the sampling energies and transmission energies. If a simulation is run for t seconds the total energy consumption for those t seconds will be given by the expression shown below.

$$E_T = \sum_{j=0}^{\frac{t}{n \times T_D}} E_j^{adap} + \sum_{k=0}^{\frac{t}{i}} E_k^{radio} \quad (51)$$

Here again based on (49) and (50) we can write E_T as:

$$E_T = f(T_{sample}, SR_{lf}, CR_{TH}^B, i, T_c^h, T_c^l, T_t, n) \quad (52)$$

Since T_{sample} and i are fixed they can be set as constants and T_c^h and T_c^l are based on wavelet filters decided beforehand and T_{sample} , they can be set as constants. SR_{lf} can be estimated at an average for a certain scenario and set as a constant too. Finally the parameters that can be varied are the Duty Cycling time, the Resampling Rate for the adaptive sampling algorithm and the bits and threshold set for the compression algorithm, based on this equation (52) can be written down as:

$$E_T = f(CR_{TH}^B, T_t, n) \quad (53)$$

These will be the parameters which would need to be optimized for best leak detection and localization times based on the energy consumption as well as the maximum distortion in the signal since the distortion is a function of the compression ratio, the higher the compression the higher will be the distortion. For this the following conditions are given for defining the bounds of the solution.

Minimize

T_D, T_L, PRD_A

Subject to

$E_T \leq E_{target}, PRD \leq PRD_A, T_t \leq T_D$ and $n \times T_t \leq T_L$

To solve such a scenario given the constraints of the algorithm regarding the minimum duty cycle time T_t , the number of times the adaptive sampling algorithm runs before it needs to resample n and the optimum compression for a given signal based on PRD , a cost function solving method using an artificial intelligence based technique of backtracking algorithm [134] is proposed, the algorithm has constraints given to it according to which it will try to calculate the most optimum solution for the given data. It uses the maximum acceptable energy consumption E_{target} , maximum acceptable distortion PRD_A , maximum acceptable leak detection T_D and maximum acceptable leak localization times T_L to calculate the best parameters falling within the parameters. The algorithm pseudo-code is given below:

Take maximum possible node energy as input E_{target} for simulation time T_{sim} .

T_D is the maximum possible Leak Detection Time.

T_L is the maximum possible Leak Detection Time.

PRD_A is the maximum possible acceptable signal distortion.

Set Minimum possible duty cycle time from T_c^h as T_t .

Set $n = 4$, the minimum adaptive sampling algorithm resampling period.

Set 100% Threshold and 24 bits for signal compression.

1. Set duty cycle time.

2. Set n for adaptive sampling algorithm.

3. Set threshold level and bits for compression algorithm.

4. Calculate energy for set parameters using equation (51).
5. If $E_T > E_{target}$ then reduce threshold and/or bits and repeat from 3 until $PRD \geq PRD_A$ else exit and give calculated threshold parameters, T_t and n .
6. Else if $E_T > E_{target}$ then increase n by 1 and repeat from 2 until $n \times T_t \geq T_L$.
7. Else if $E_T > E_{target}$ then increase T_t by T_{st} and repeat from 1 until $T_t \geq T_D$.

In the algorithm shown above PRD is the distortion in the signal due to the compression, this can be estimated by using a relationship between compression parameters and distortion for a particular set of wavelet filters used, for example the graphs of the relationship between distortion and compression are given in Chapter 5.3 for a daubechies 5 wavelet, other wavelet types would give different relationships. Also T_c^h will depend on the type of wavelet and the signal length used and T_t derived from it for the first sampling duty cycle would be at least 15 seconds higher than this to allow the radio to transmit the data in between as per the scheme. For the first energy reduction scheme optimized as per the solving scheme in step 5 the algorithm would iteratively set lower threshold levels and number of bits for the signal and calculate the CR and PRD for the signal using the relationship for the particular wavelet type used and if E_T is within E_{target} for any setting the algorithm would terminate the solving process and give the calculated threshold setting and number of bits and n , which is the resampling period of the adaptive sampling algorithm, also it would give T_t , which is the duty cycle time which it would calculate, if the E_{target} is not met for any of the conditions then the algorithm will move on to the next step after resetting the compression parameters. The second level of the energy

optimization process is in step 6 which is increasing the resampling period n until it reaches a maximum threshold level of $n \times T_t > T_L$, at each iteration n would be increased by 1 and the algorithm re-run from step 2. Even if by using this method the energy consumption does not fall below E_{max} , the final step is to increase the duty cycle period T_t by T_{st} and then going back to step 1 to re-run the whole solver again. This solving process gives priority to leak detection time, then leak localization time and then signal compression given that they are within the constraints that are given to the solver and E_{target} is achievable. For the algorithm shown the solver would go about the solving process as shown in figure 4.11.

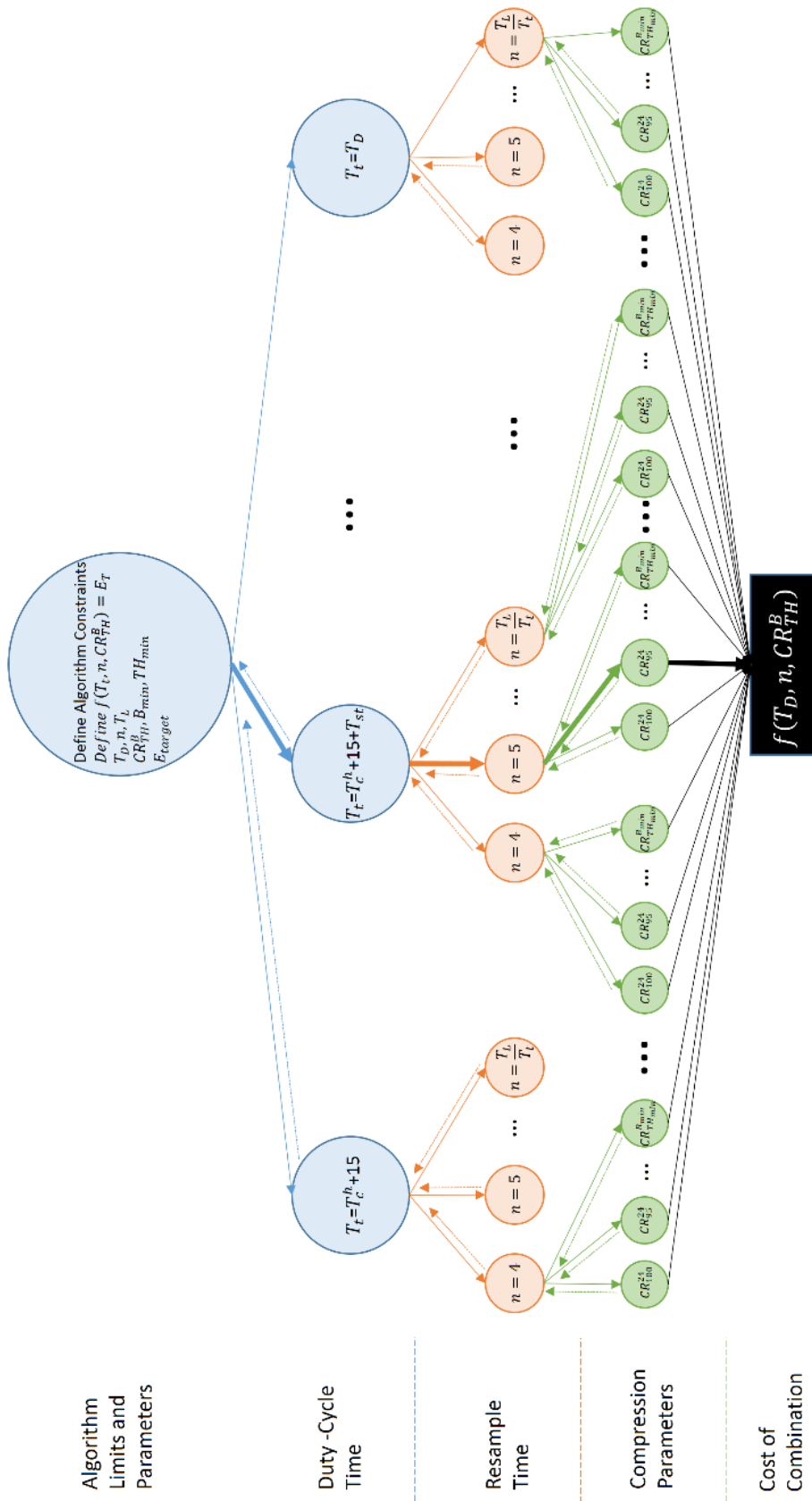


Figure 4.11: Paths the backtracking algorithm will take to find the most optimal solution. The solid arrows represent forward paths and the dotted arrows shows the backtracking when the cost function is not satisfied for the set parameters.

In figure 4.11 the scheme for setting the parameters of the proposed algorithm is shown, the parameters would be set from the minimum duty cycle time, number of resampling periods and the compression parameters and iteratively increased as per the scheme shown until the cost function is satisfied, the algorithm will backtrack to the previous level, increase the parameters with step size and then again calculate cost of the new scheme. First the compression parameters would be iteratively increased in the third level of the algorithm, then the resampling period in the second level and lastly the duty cycle time in the first level, the motivation being to increase the duty cycle time at last to have a minimal effect on the leak detection time. The lines in bold give one possible choice which complete the constraints on the leak detection times, signal distortion and maximum allowed energy consumption and these parameters may be chosen for a given scheme. There are other solving methods also available and these would give different results based on the parameters used and the priority given to each of the energy efficiency techniques employed. For example artificial intelligence techniques like genetic algorithm or particle swarm optimization can also be used to solve the above mentioned optimization problem and would give a different result based on the cost function used for the solving process.

Moving on, the next section is related to the simulation where the pipeline simulation model discussed in chapter 3 will be used to test the developed energy conservation algorithm in different cases of the leaks.

CHAPTER 5

RESULTS

This section is related to the testing of the whole scheme and the advantages in energy consumption it brings as a whole to the leak detection scheme, it will be seen if the changes are successful in maintaining acceptable leak detection and localization performance. First it will be checked how the adaptive sampling scheme that is described in the previous section calculates the sampling frequency and then selects the sampling frequency based on the median of the last three sampling frequency calculations, the sampling rate selected will be compared to the spectrum of the signal at that point, it will also be seen how much different leak magnitudes effect the signal as different leak strengths will be affecting the vibrations differently. Then the adaptive threshold based algorithm in which the leak is detected by using the medians of the vibration data will be discussed and its performance when detecting leaks of different magnitudes will be shown. Afterwards the compression algorithm will be discussed and the relationship between the CR and PRD of the signal will be discussed as it will be an important indicator when the algorithm will be used for localizing leaks. Afterwards the energy consumption gains that all three energy conservation schemes will be giving to the monitoring scheme will be discussed for different leak sizes and how much they will affect the timeliness of the leak detection and how will they be affecting the quality of the leak localization that will be done by cross-correlation of vibration signals. The cross-correlation will be done to localize the leak at the end of the scheme at the sink, the quality of the cross-correlation result will greatly depend upon the strength of the leak and the compression level that will be used. It should be noted that we will be running the simulations for similar scenarios for all the different

energy conservation schemes, the simulation time is fixed at 12000s and the gain in energy savings will be compared regarding monitoring in this time period. Before analyzing the complete algorithm we will first see how the adaptive sampling performs for different leak sizes and different distances from the leak. This is shown in the next sub-section.

5.1. Adaptive Sampling Algorithm

The adaptive sampling scheme is designed such that it would sample the signal once using the high frequency sensor, calculate the frequency, use the frequency with the two most recent frequency calculations and then select the low energy sensor sampling rate using the median of the three readings. For the low energy sampling, we will use the sensor for n cycles as specified in the algorithm shown in figure 4.6. During these low energy cycles the sensor would sample at the selected rate until conditions for resampling at high frequency will be met. Before discussing the adaptive sampling scheme however it is to be seen how the leak affects the vibration spectrum at different points in the pipeline, this was already discussed in chapter 3, figures 3.15 to 3.18 detail the effect of the leak size on the spectrum of the signal for each individual node. It was found that the closer the node is to the leak the higher the bandwidth of the signal reaches it, also the intensity of the signal will be higher. Consequently it is assumed that the Nodes nearer to the leak i.e. Nodes 10 and 14 would react more with regards to change in sampling frequency as opposed to Nodes further away, i.e. Nodes 7 and 18.

Related to this we will consequently be looking at how the sampling rate changes before and after the leak for different leak sizes, n or the number of times the low energy sensor will sample the signal before resampling is required is 9. The daubechies 5 wavelet is used with 10 coefficients in the low pass and high pass filters. It should be noted that the wavelet

filters with higher the amount of coefficients will give better separation of the frequency bands when calculating the frequency but will give a higher amount of computation cost as a result. Figure 5.1 shows the performance of the adaptive sampling scheme for Node 10 for a 1mm leak. Node 10 is 23 m from the leak location.

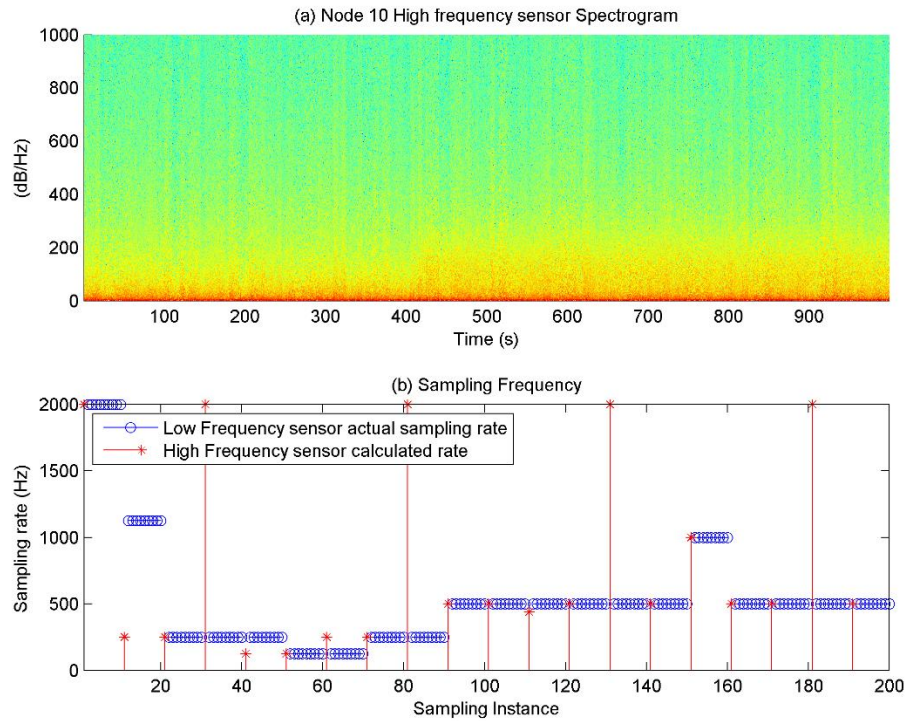


Figure 5.1: (a) The frequency spectrum and (b) the corresponding sampling frequencies 23 m from 1mm leak. In figure 5.1(a) the frequency spectrum of the vibration signal can be seen for the whole simulation. From the whole 12000s simulation a total of 1000s of vibration data has been generated over 200 sampling instances of 5s sampling periods each. In (a) the signal that is being sampled by the sensors is shown, it can be seen that after 400s the spectrum of the signal slightly changes, this is due to the 1mm leak which has been introduced in the system. For the corresponding signal in part (b) the calculated and actual sampling rates have been plotted, it is seen that the adaptive sampling algorithm initially calculated higher than usual sampling frequency of 2KHz but due to the medians of the last 3 readings being

used settled to a level of 125 – 250 Hz, this is consistent with the frequency spectrum which shows significant energy around 50Hz before 400s. Also due to the medians being used, in subsequent sampling periods where higher than usual frequencies are calculated for one cycle due to noise in the signal the algorithm keeps the sampling rate consistent. After 400s as the frequency spectrum changes and reaches up to 200Hz, the algorithm also adapts and starts calculating the sampling frequency to be readjusted up to 500Hz. This can be seen in figure 5.1(b). Again it is observed that higher than usual sampling frequencies are suppressed by the algorithm as we take the median of the last 3 calculations to set the final sampling rate. Figure 5.2 is plotted similarly for Node 14.

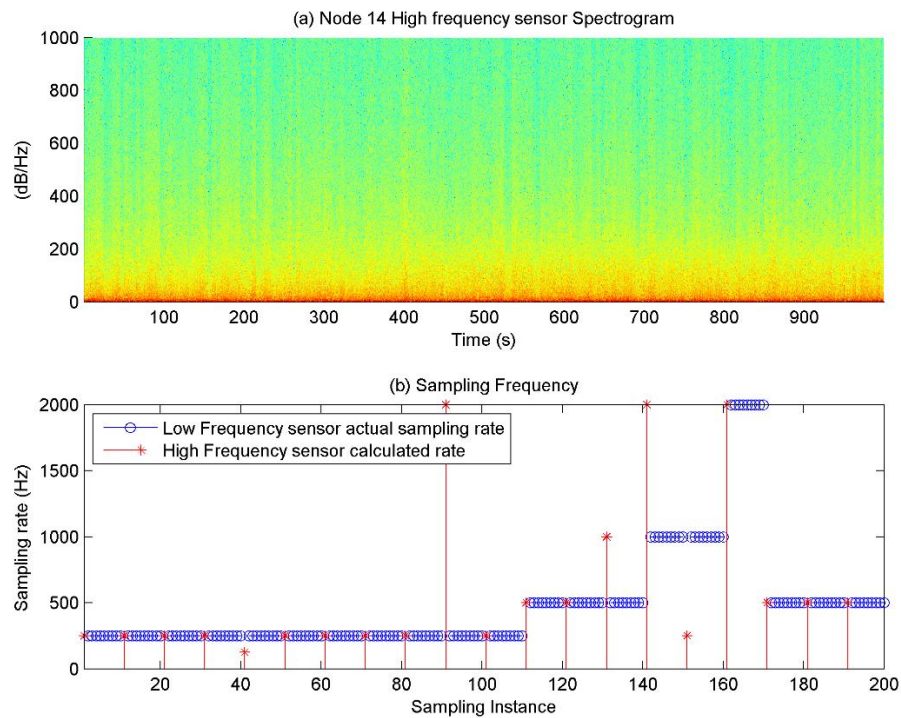


Figure 5.2: (a) The frequency spectrum and (b) the corresponding sampling frequencies 27 m from 1mm leak. In figure 5.2(a) the frequency spectrum of the signal for the simulation is shown again, here the scenario is the same but the node is further away from the leak, likewise it can be seen that the change in the frequency spectrum is not very visible after 400s but a slight change

in frequencies up to $\sim 200\text{Hz}$ can be observed. The adaptive sampling algorithm that was introduced in Chapter 4 is again used here and it keeps the sampling frequencies fairly consistent up till 500s, after which the sampling algorithm starts calculating frequencies higher than the previous average and because of it the sampling frequency shift upwards to 500Hz and in some instances higher because of multiple calculation results of high frequency.

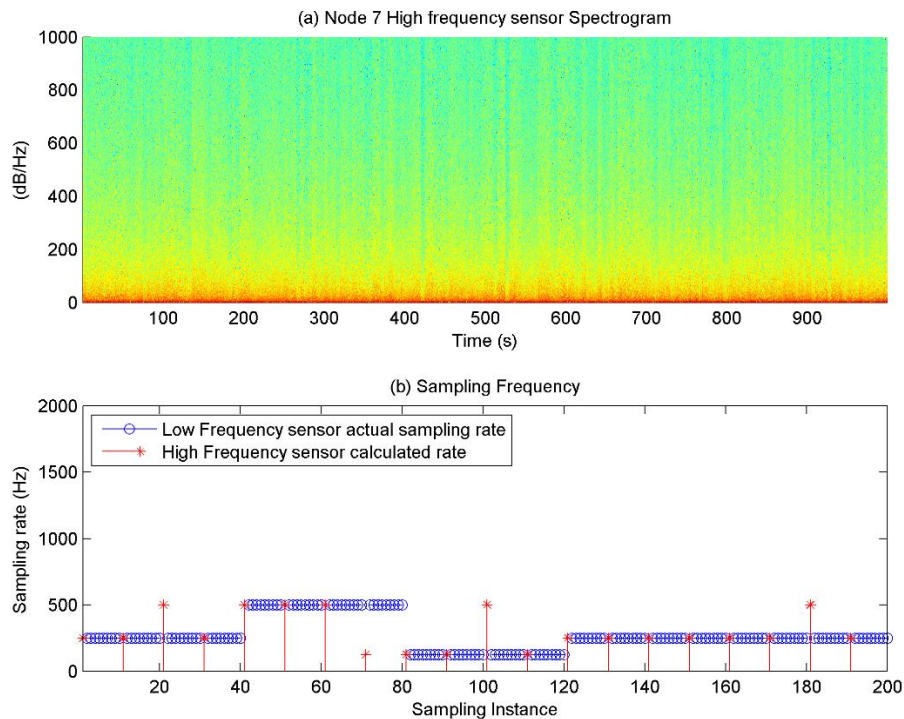


Figure 5.3: (a) The frequency spectrum and (b) the corresponding sampling frequencies 73 m from 1mm leak. It should be noted that the frequencies that reach one location is a direct result of the pipe frequency response and the longer the distance of the pipe the more the signal would be attenuated and the more narrowband it would become as a result. Going by the observations and by the results shown in the figures above it can be safely concluded that weak leak signatures would not affect vibrations on the measurement points which are further away,

to see this effect we will plot the vibrations at nodes 7 and 18 and observe the effect of vibration over time for the leak in figures 5.3 and 5.4.

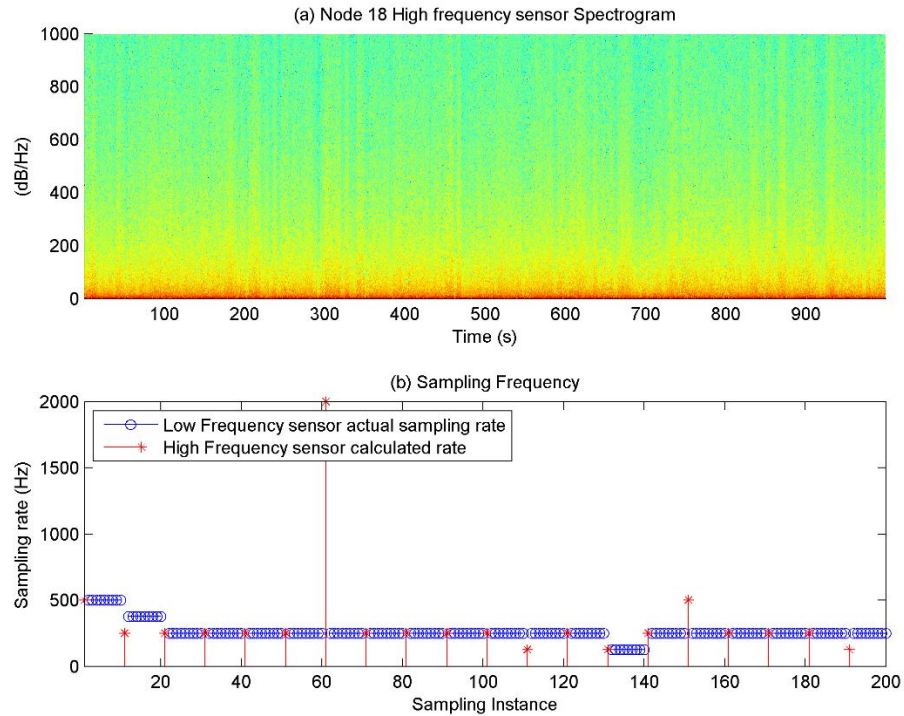


Figure 5.4: (a) The frequency spectrum and (b) the corresponding sampling frequencies 77 m from 1mm leak.

In figures 5.3 and 5.4 it can be observed from the frequency spectrums that the vibration spectrum is not changing before and after the leak. This is due to the attenuating and band-limiting characteristics of the pipe. Likewise the sampling frequencies are fairly stable throughout the whole simulation for these two sensors at around 250Hz.

In the figures shown above the effect of 1mm leak on the sampling frequencies calculated by the adaptive sampling algorithm was discussed, after this the effect of 3mm leak will also be seen. Figure 5.5 shows the effect of 3mm leak at Node 10 which is 23m from the leak.

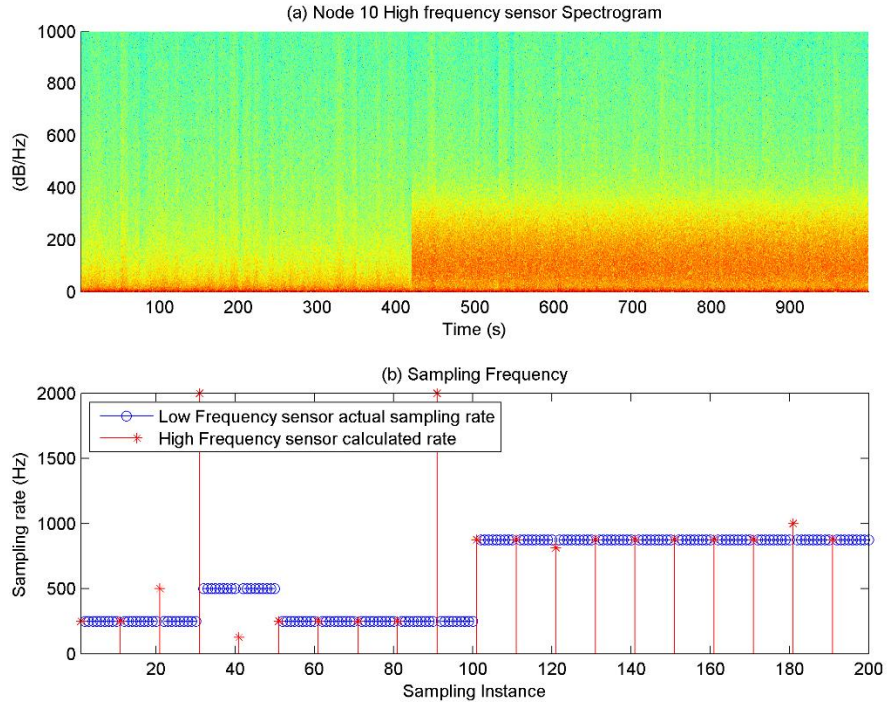


Figure 5.5: (a) The frequency spectrum and (b) the corresponding sampling frequencies 23m from 3mm leak. Here the effect of leak on the frequency spectrum is more visible, this is mainly due to the fact that the leak is stronger and as a result is propagated further through the network, due to this it can be observed that up till 350Hz in the spectrum shown in figure 5.5(a) after 400s the leak signature is very strong and due to this the algorithm calculates the sampling frequencies between 800Hz and 1000Hz in the leading cycles of sampling using the high frequency sensor as shown in 5.5(b). Similarly for Node 14, 27m away from the leak locations the vibrations are plotted and the performance of the adaptive scheme observed.

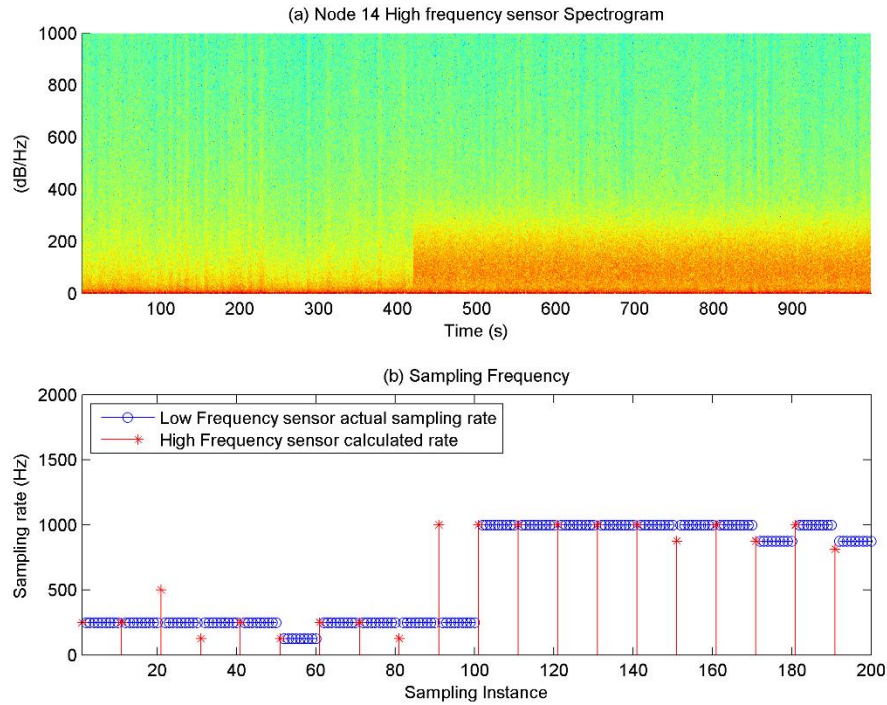


Figure 5.6: (a) The frequency spectrum and (b) the corresponding sampling frequencies 27m from 3mm leak.

In figure 5.6(a) the spectrum is again changing drastically after 400s as the 3mm leak is occurring at that mark, the bandwidth however is lower, concentrated at $\sim 300\text{Hz}$, this would still increase the sampling rate however and in figure 5.6(b) we can observe the algorithm sets a sampling rate of 1KHz, however due to the median of three previous calculations it takes up to 10 cycles to sample at the adjusted rate. Similar to this the sampling rate for Nodes 7 and 18 are plotted in the following figures.

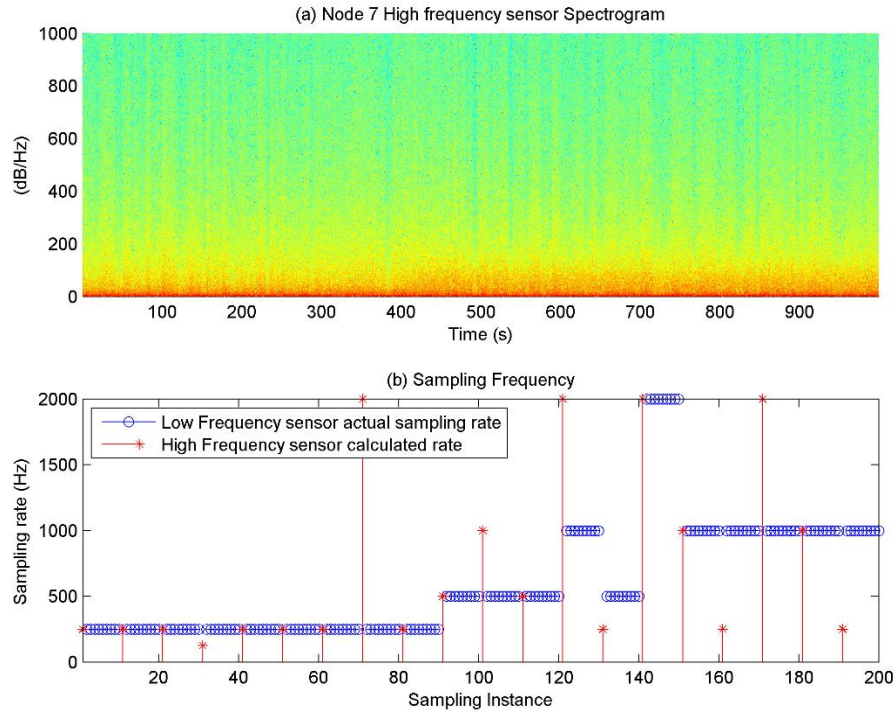


Figure 5.7: (a) The frequency spectrum and (b) the corresponding sampling frequencies 73m from 3mm leak.

In figure 5.7(a) it is observed that the frequency spectrum reaches around 150Hz after the leak, this is again reflected in the sampling frequencies that the algorithm calculates, but in some instances the algorithm is consistently calculating higher frequencies than observed, this is due to the effect of the wavelet transform such that it doesn't exactly divide the frequency spectrum in half due to this effect the adaptive sampling algorithm is not able to calculate the true sampling frequency which should be around 500Hz for this scenario, still keeping in mind the Nyquist sampling rate for such a signal the sampling demands are satisfied. Similarly the signal at Node 18 is plotted along with the sampling rates in figure 5.8.

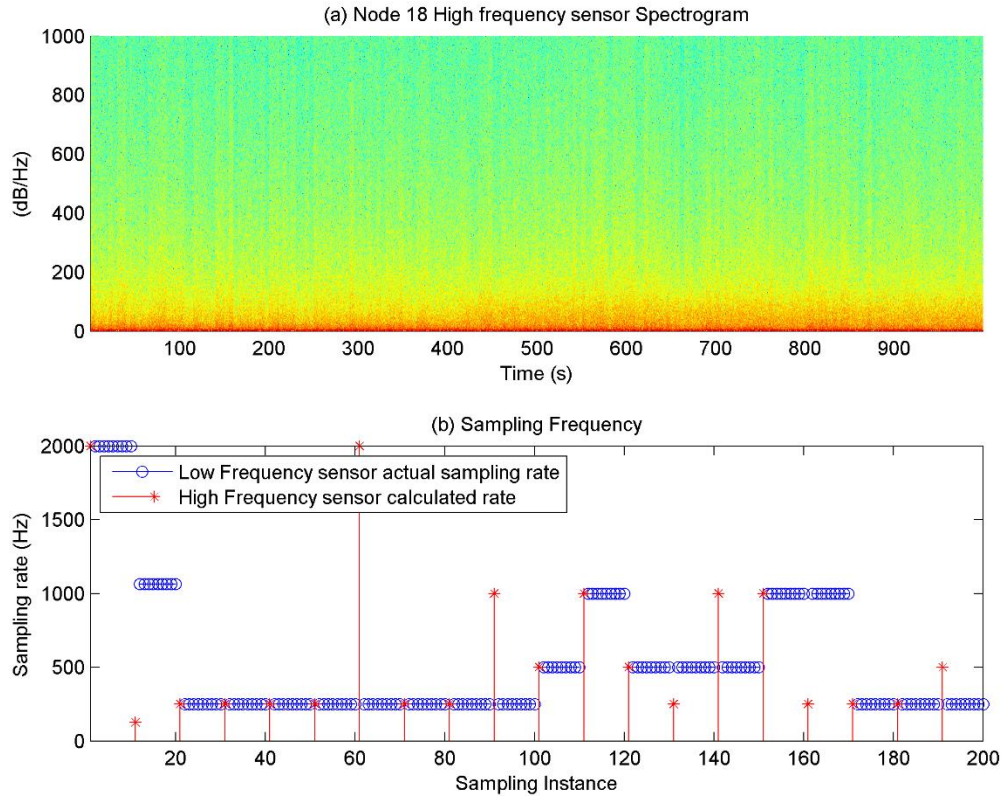


Figure 5.8: (a) The frequency spectrum and (b) the corresponding sampling frequencies 77m from 3mm leak. Again in figure 5.8(a) the minimal increase in bandwidth up to 150Hz is observed after the leak, it can be observed that the moderately strong leak effects the vibration further away from the leak position. In figure 5.8(b) the calculated frequency reflects this change but again the issue is with the wavelet type used in the algorithm, a filter pair which better divides the frequency into half will give a better result but would result in more computations. The results for the adaptive sampling for 5mm leak for the 4 Nodes under observation are plotted next, it should be noted that 5mm leak will be significantly stronger than either leak so the bandwidth of the signal will be larger across all Nodes that under observation in the simulation.

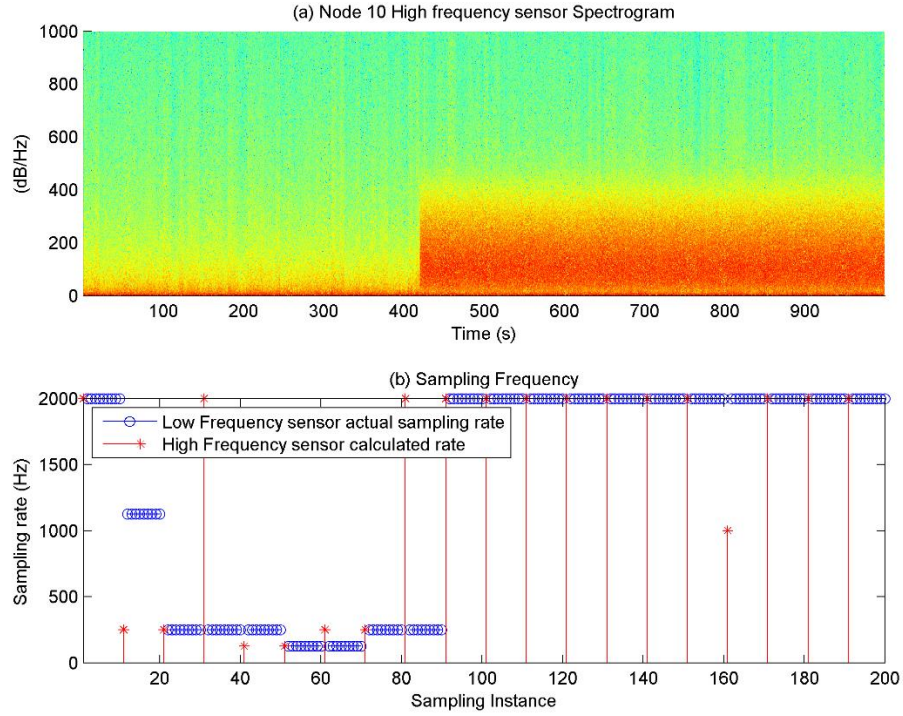


Figure 5.9: (a) The frequency spectrum and (b) the corresponding sampling frequencies 23m from 5mm leak.

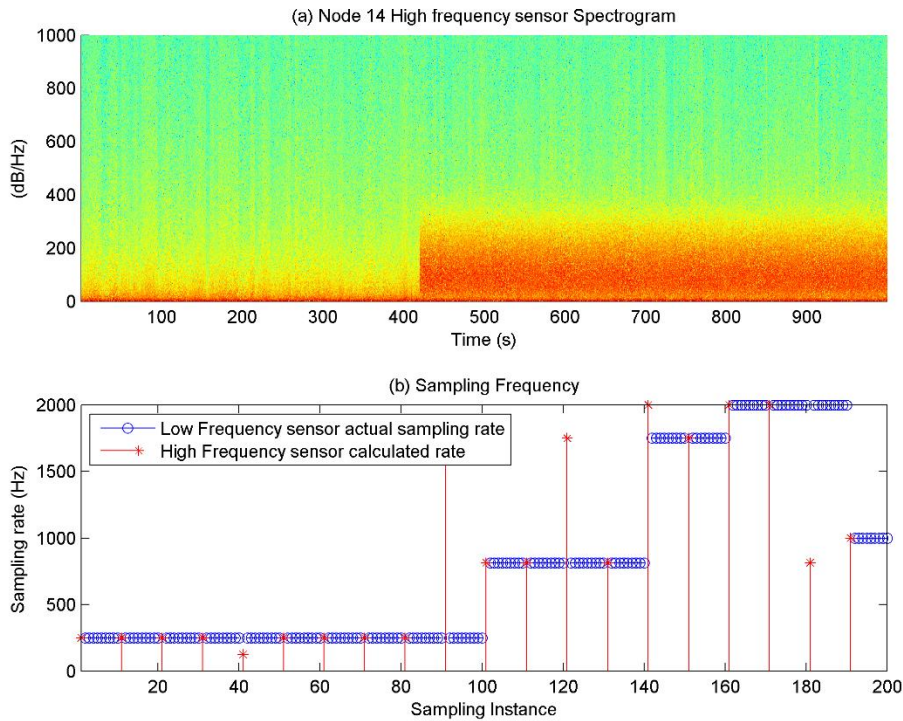


Figure 5.10: (a) The frequency spectrum and (b) the corresponding sampling frequencies 27m from 3mm leak.

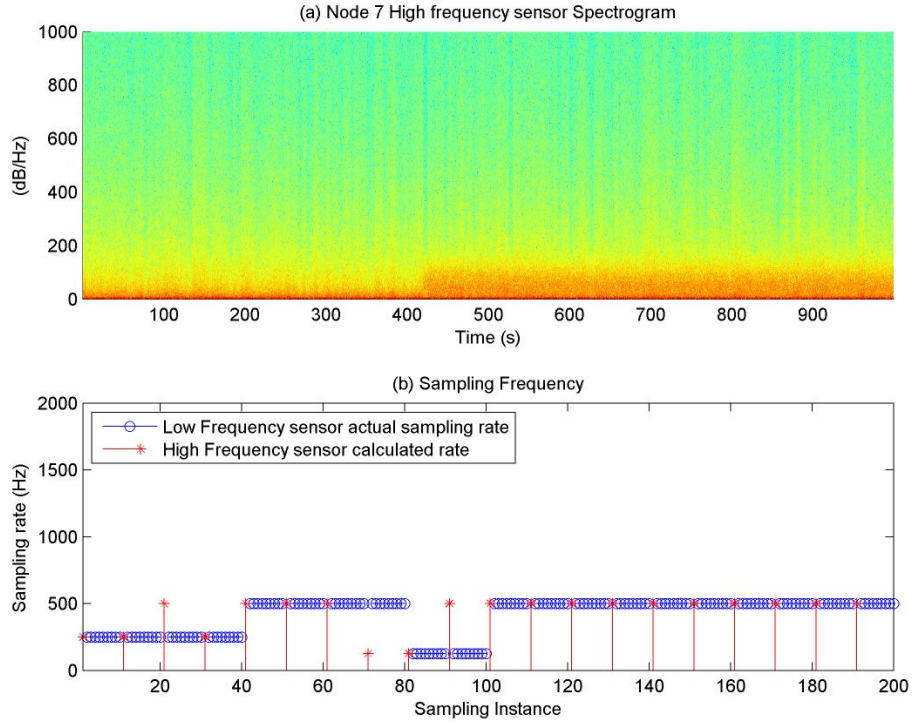


Figure 5.11: (a) The frequency spectrum and (b) the corresponding sampling frequencies 73m from 3mm leak.

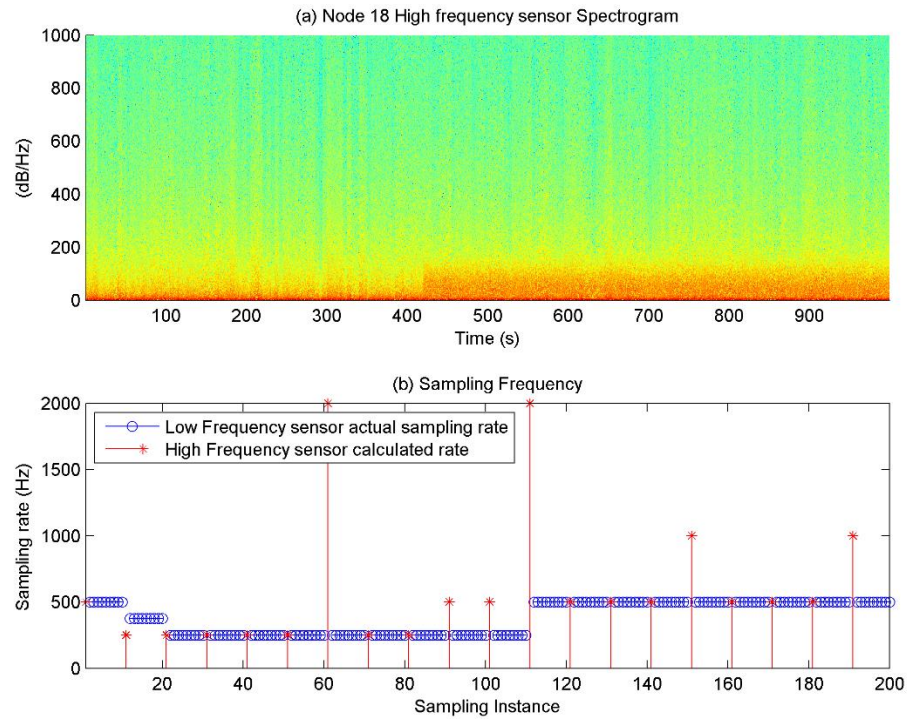


Figure 5.12: (a) The frequency spectrum and (b) the corresponding sampling frequencies 77m from 3mm leak.

From the figures shown above it is observed that the adaptive sampling algorithm formulated is largely following the increase in bandwidth of the signal and sampling at higher than the previous 3mm case, also the sampling rate selected is satisfying the Nyquist criteria in that it is sampling at more than twice the rate of the bandwidth of the signal. One point to be noted is that in figures 5.11 and 5.12 the sampling rate is lower than the sampling rate in figures 5.7 and 5.8, this is mainly due to the fact that after the first level of decomposition the algorithm doesn't find any more significant coefficients in the lower sub-bands, due to this it selects the sampling frequency at 1KHz, on the other hand the signal is stronger for the 5mm leak in the lower sub-bands, because of this reason the algorithm selects the lower sampling frequency of 500Hz.

Next the adaptive threshold based algorithm is discussed. The median of the signal was taken among the multiple statistical parameters that can be used to quantify leaks or anomalies in pipelines, the results are discussed in the next chapter.

5.2. Adaptive Threshold Algorithm

In this section the performance of the proposed adaptive threshold based scheme is discussed, equations (31,32) proposed in Chapter 4 are designed such that the effect of noise on the signal will be minimized as the leak signature is noisy in nature and will be varying in a certain range given other conditions are kept constant. If the current reading is higher than two consecutive previous readings the leak is detected as per the algorithm. First the effect of the leak on the medians of the signal measured at the nodes will be plotted and compared.

To judge the performance of the algorithm first the effect of the leak on the medians of the vibration signal at different distances would be found out for different leak sizes, figure 5.13 shows the plot of 1mm leak effect on the medians of the vibration signal. The median magnitude doesn't change and follows the same trend as the effect of the leak is not very high. In this case the adaptive threshold based algorithm would not be able to detect the leak as the magnitude of the leak is insignificant enough that it doesn't affect the vibration magnitude.

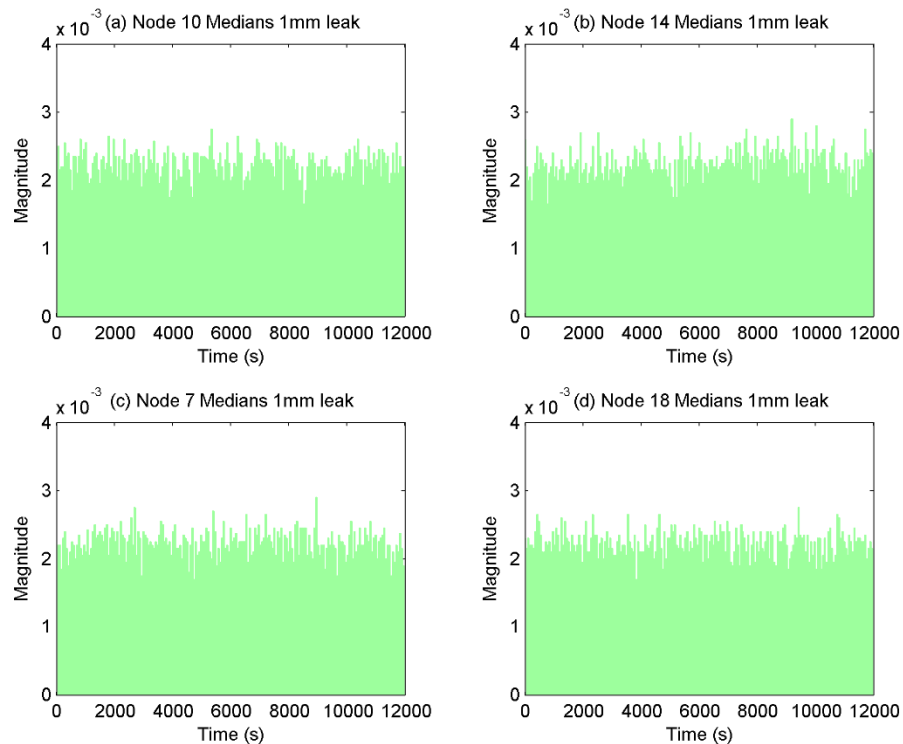


Figure 5.13: Median magnitudes of vibration readings different distances from 1mm leak for before and after leak.

For the other case of 3mm leak the vibration magnitudes are again plotted as shown in figure 5.14. Here it is observed that the leak is significant enough that it is affecting the signal at both Nodes 10 and 14, these two nodes are nearer to the leak and signal is not highly attenuated at these points, again further out the vibrations are not affected by the

leak as the larger distances have highly attenuated the leak signature, as such there is high chance of detecting leak at either nodes 10 or 14.

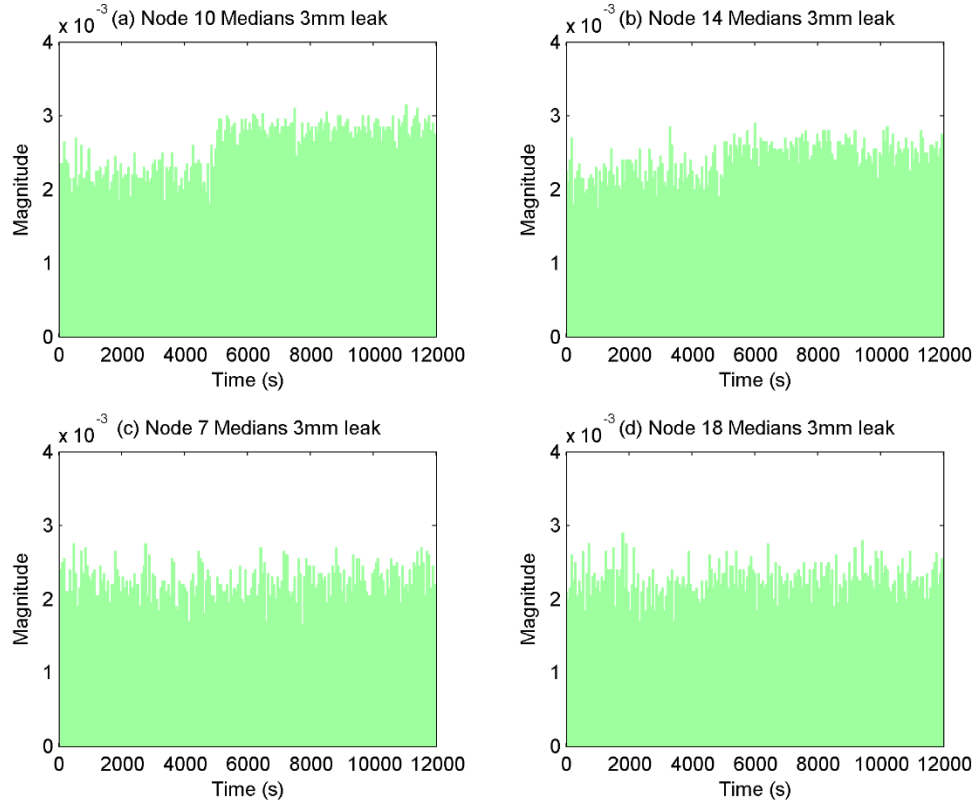


Figure 5.14: Median magnitudes of vibration readings different distances from 3mm leak for before and after leak.

On these scenarios the proposed adaptive threshold based leak detection algorithm will be applied. It should be noted that the threshold based algorithm cannot detect leaks of very low magnitude as evident from figure 5.13. And as shown in figure 5.14(c) and (d), the vibration sensors further from the leak wouldn't be able to detect the leak either, this can be improved by installing a denser sensor network on the pipeline at the cost of increasing the system hardware cost. Furthermore for very weak signals filtering of the signal will be required to filter out the event from the noise to detect the signal, this would increase the detection time of the leak and increase the computations at the node. This result also

necessitates the need of a WSN monitoring solution for vibration sensing with multiple nodes to effectively cover the pipe network and detect leaks. The adaptive threshold scheme described previously will be applied with the adaptive sampling scheme and will show the effectiveness of the leak detection in the following figures. The adaptive sampling algorithm explained previously will sample at high frequency for 1 cycle and then sample at n cycles using the low frequency sensor, in these two different sampling cycles the medians would be calculated, and for the low frequency equation (31) will be used to detect the leak and for the high frequency event equation (30) will be used to confirm the leak using the algorithm presented in figure 4.6. For the 3mm leak the leak detection by the algorithm using medians is shown in figure 5.15 for Node 10.

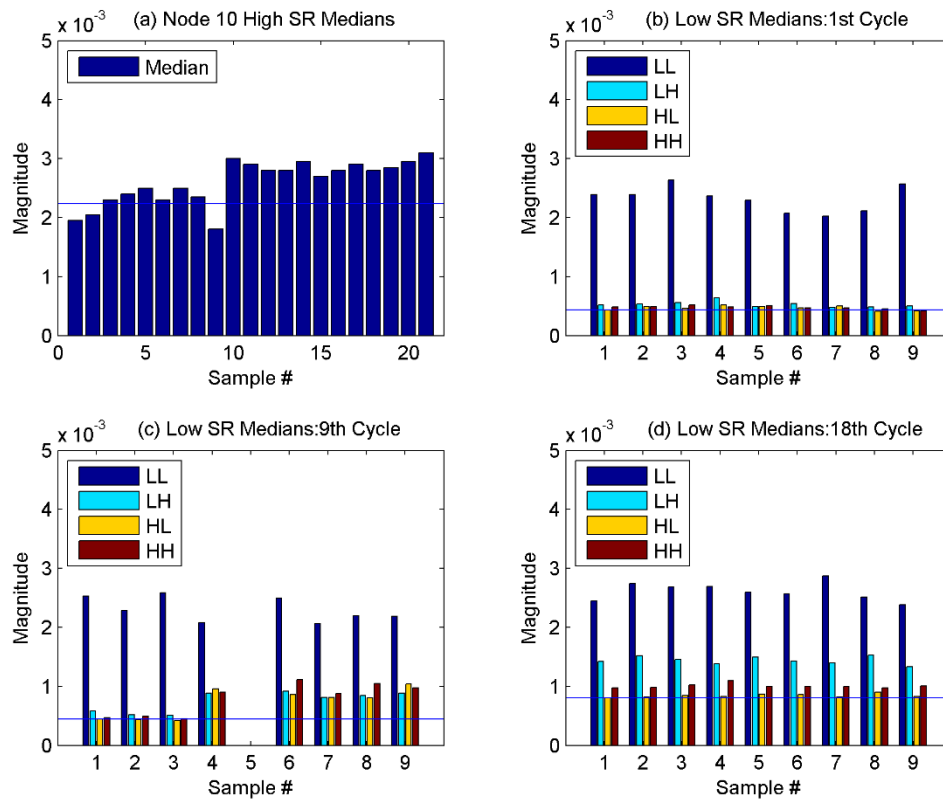


Figure 5.15: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 10 in 3mm leak.

In figure 5.15 the medians of the vibrations measured by the sensors at different times by both (a) high energy and (b), (c) and (d) low energy sensors is shown, the low energy sensors have medians taken after the signal has been passed through 2 level wavelet packet decomposition to better divide the sub-bands, these sub-bands are designated LL, LH, HL and HH in order from lowest frequency sub-band to highest frequency, the range of each sub-band is 1/8 of the sampling rate. The percent which any band of the low energy data may be higher than the previous corresponding data is 35% and the median of the high frequency sensor is set at 17.5%. It was seen from data that this much change was detectable by the sensors. Here it can be seen that algorithm detects the leak in the 4th sampling instance in 5.15 (c) and subsequently calls the high frequency sensor in the 5th sampling instance to confirm the leak event, after this it continues sampling using the low frequency sensor until the resampling time occurs.

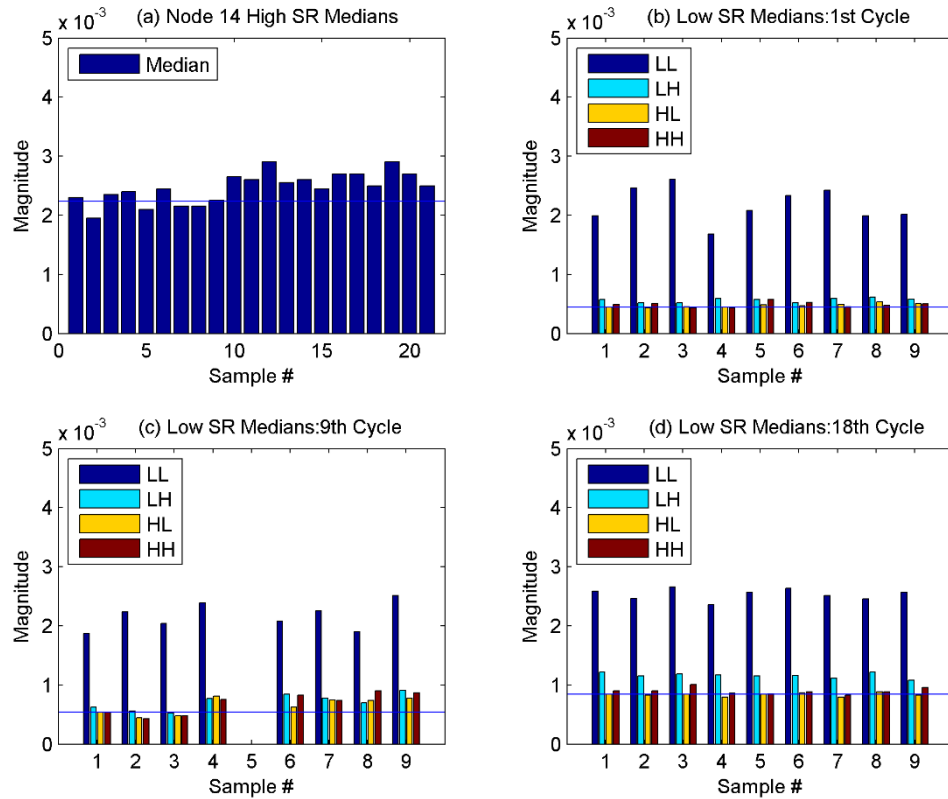


Figure 5.16: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 14 in 3mm leak.

Similarly the sensor at Node 14 detected the leak and the result is shown in figure 5.16. Here the result is also similar to figure 5.15 in which the node detects the leak at the same time instance. The leak detection and localization flags are shown in figure 5.16 for this particular scenario, any node which detects the leak will send out a leak detection flag to the sink and the sink will then ask the other nodes at their next high frequency sampling cycle to transmit the relevant data in order to localize the leak. This is shown in figure 5.17.

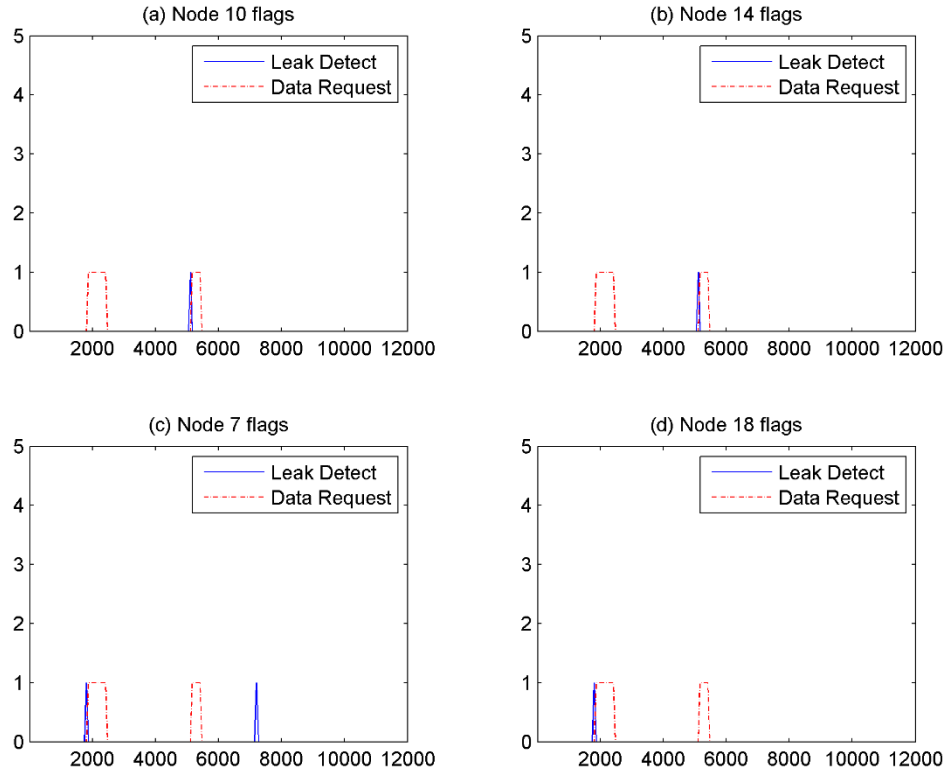


Figure 5.17: Leak detection and data request flags from the 4 Nodes in the system.

In figure 5.17 the nodes which transmit the data to the sink can have pairs programmed into them in which the sink requests data from specific adjacent nodes when the leak detect flag has been transmitted to it, this flag then triggers the leak localization request from the sink which it requests at the next high sampling routine from the Nodes, this is shown by the red dashed line which goes back to zero after the Nodes have transmitted the data to the sink. At this point the sink has enough information to localize the leak if it is occurring, in 5.17(c) and (d) the nodes detected a false positive and had localization request from the sink but in this case since there was no leak there would have been no result, on the other hand in figure 5.17(a) and (b) the nodes detected the leak at the time it occurs and sent the event to the sink which then sent the data request to the nodes in the next high frequency sampling cycle. The radio wakes up each high frequency sampling cycle to see if there is

any request from the node and goes to sleep so localization of leaks can only occur when the high frequency sensor wakes up, in this case it is every 10 minutes or 600 seconds and it depends upon the parameters used for the adaptive sampling algorithm like the ratio between the high frequency and low frequency sensors and the duty cycle.

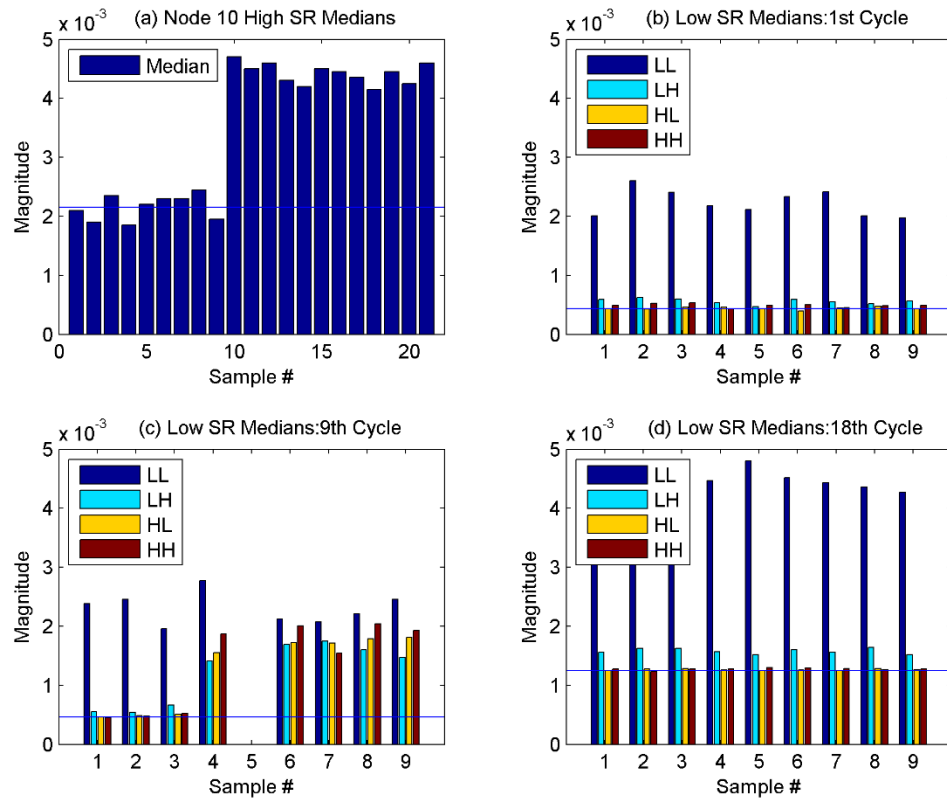


Figure 5.18: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 10 in 5mm leak.

Leak detection however in this case can be done in 120 seconds or only 2 minutes as the duty cycle is set at 60s and a minimum of two sample periods are required as per the current proposed algorithm. Moving on the case for 5mm case was also considered and it has the same result, in fact leak detection was easier as the change in median was more pronounced than before. In figures 5.18 and 5.19 the effect of the leak on the medians is shown and it is fairly high compared to the 3mm leak case for both nodes 10 and 14. It can be seen that

the higher magnitude of the leak produces a higher response in both the low energy sensor and high energy sensor.

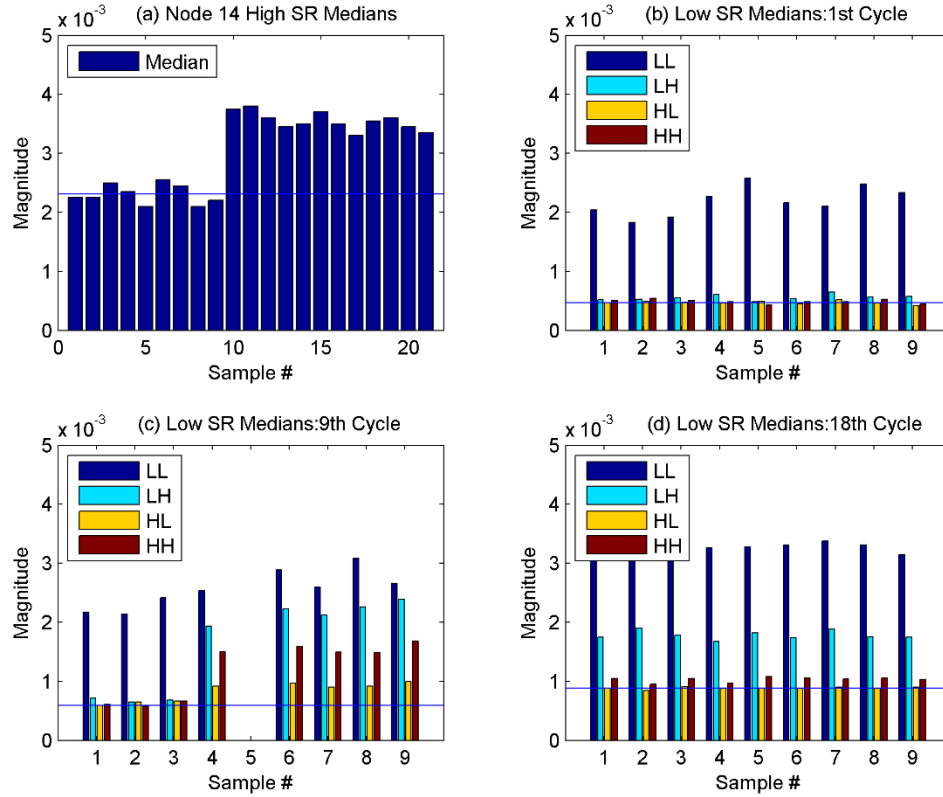


Figure 5.19: Medians calculated by the adaptive threshold algorithm and the leak detection using both low (LSVS) and high (HSVS) frequency sensor for Node 14 in 5mm leak.

Moving on from this aspect of the algorithm the next energy saving aspect will be discussed related to signal compression in the next sub-section.

5.3. Signal Compression

Signal compression is done to reduce the size of the data, this is beneficial for both storage and transmission of the data, it entails in extra computation but results in a net gain in terms of energy saved as the radio energy expended is lesser than previously. Compression usually entails a distortion in signal for different compression ratios as the compression used in this work is lossy in nature. The algorithm has already been explained in the

previous chapter in detail, first the wavelet transform is carried out up to level 5 on the high frequency data, then the threshold is applied to the coefficients and then sub-band adaptive quantization is applied. After this the signal is encoded in a bit-stream. Figure 5.20 shows the effect on the signal before and after compression has been applied to it sampled using high frequency sensor, 3mm leak case is considered.

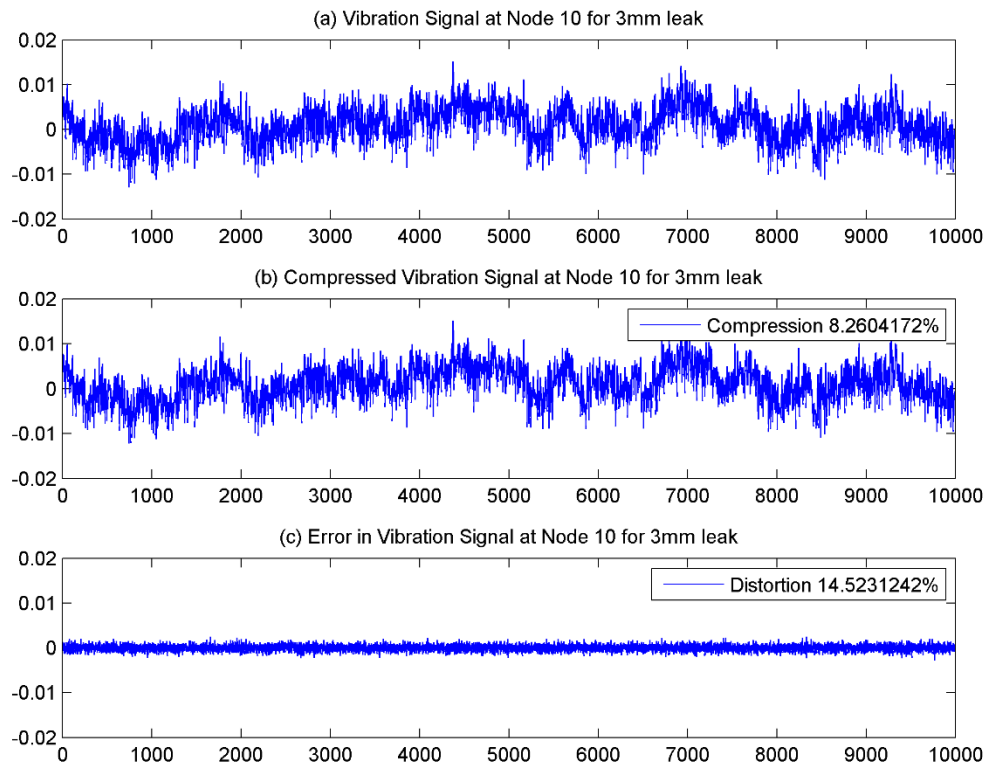


Figure 5.20: Compression and distortion for 3mm leak signal at Node 10.

It can be seen in figure 5.20 that the signal has distortion up to 15% for a compression just above 8% of the total signal length, transmission wise this shows a gain of 92% in transmission cost but computation costs will be increasing on the other hand and will be taking up most of that 92% gain achieved but due to lower computation power there would still be net gain, for the above signal the threshold was set at 80% of the signal magnitude

sum and 4 bits for the quantization. 3 bits were used instead of 4 to see the effect on CR and PRD. The result is shown in figure 5.21.

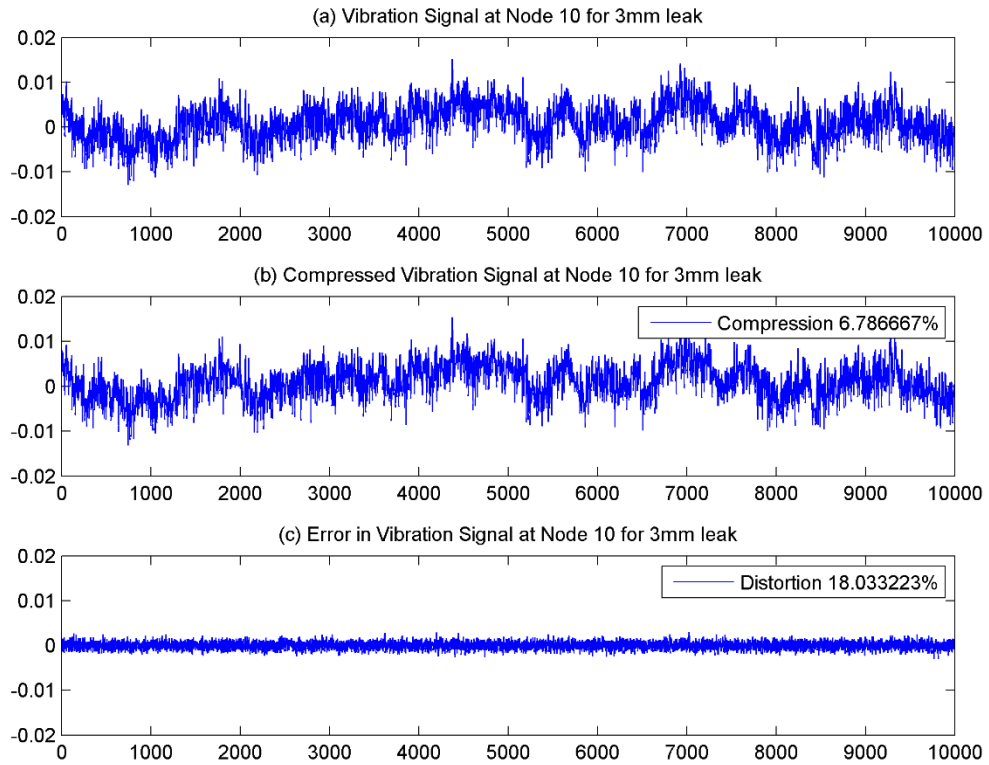


Figure 5.21: Compression and distortion for 3mm leak signal at Node 10, 80% threshold, 3 bits SAQ.

In figure 5.21 it can be seen that the compression increased at the cost of slightly more distortion. Again decreasing the threshold to 65% we can see that the distortion increases in figure 5.22, this way we can create an approximate relationship between compression ratio and distortion which would be helpful in determining later how much distortion is expected from the signal for a given CR, as CR is easily calculated at the node and the distortion is computationally expensive.

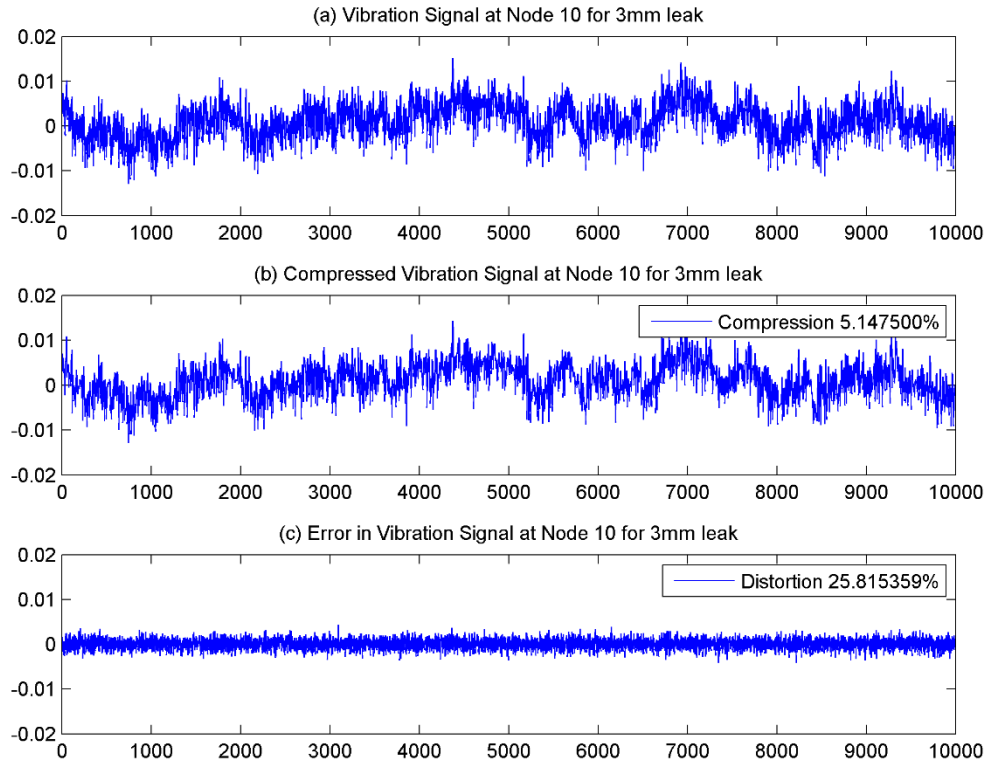


Figure 5.22: Compression and distortion for 3mm leak signal at Node 10, 65% threshold, 4 bits SAQ.

Using information from multiple leak magnitudes and different compression rates results are generated showing the trend of the compression ratio versus the distortion. Data was generated for a 3mm leak scenarios and the results plotted in the figures shown below.

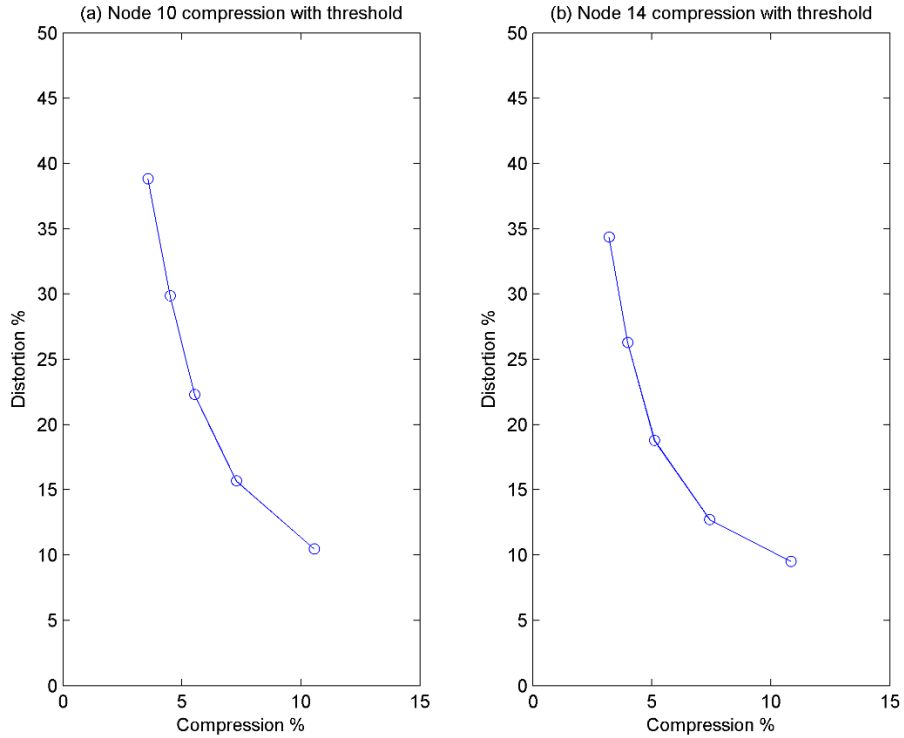


Figure 5.23: (a) Compression at Node 10 with different compression thresholds, (b) similar compression at Node 14 for 3mm leak.

In figure 5.23(a) and (b) the effect of compression threshold can be observed, the threshold value was started at 90% and then reduced in 10% steps till 50% of the maximum sum using a 4 bit sub-band adaptive quantization (SAQ) scheme. It can be seen in both graphs that 90% of the value is contained in ~12% of the data with approximately 10% PRD, compressing the data further gives a logarithmic curve in which the PRD increases exponentially while achieving compression of ~3.5%. The further the Node is away from the leak the less leak signature is present and less distortion is present for a particular compression method as evident from graphs in figure 5.23. Next the effect of varying the bits is observed on the compression performance. The threshold level was set at 80% of the sum and the bits were varied from 6 bits to 2 bits. The result is shown in figure 5.24.

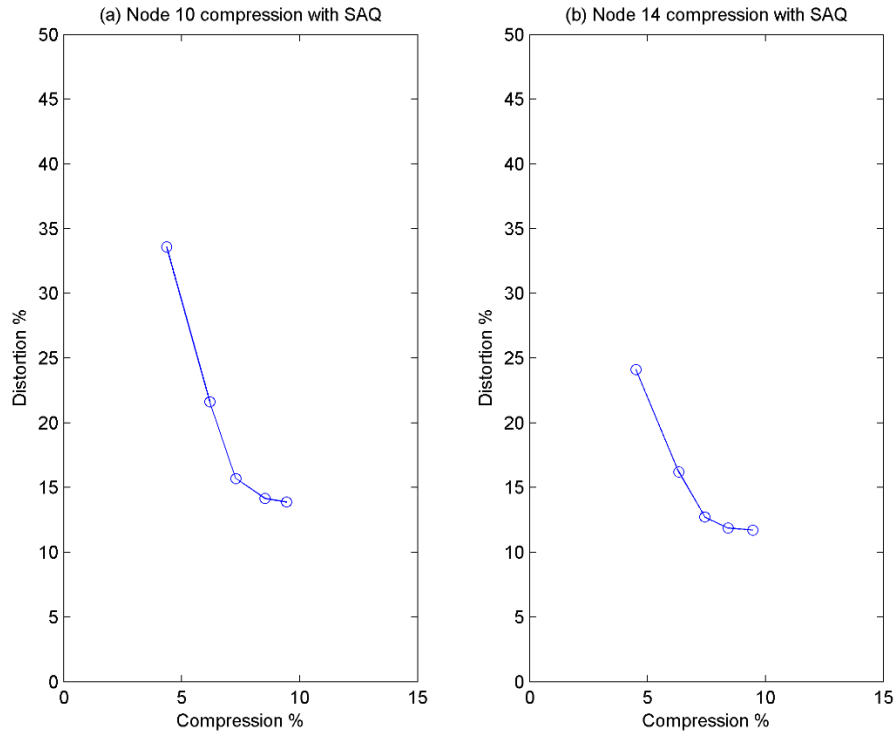


Figure 5.24: Effect of varying SAQ from 6 bits to 2 bits.

In figure 5.24 the effect of varying the number of bits is plotted, the bits started from 6 bits and again a logarithmic effect is seen in which after 4 bits the distortion increases significantly when 3 and 2 bits are used. Again it is seen that the signal nearer to the leak has more distortion for a given compression setting.

To observe the effect of compression on the leak magnitude another graph was plotted for a 5mm leak with similar compression settings as in figure 5.23 for the 3mm leak.

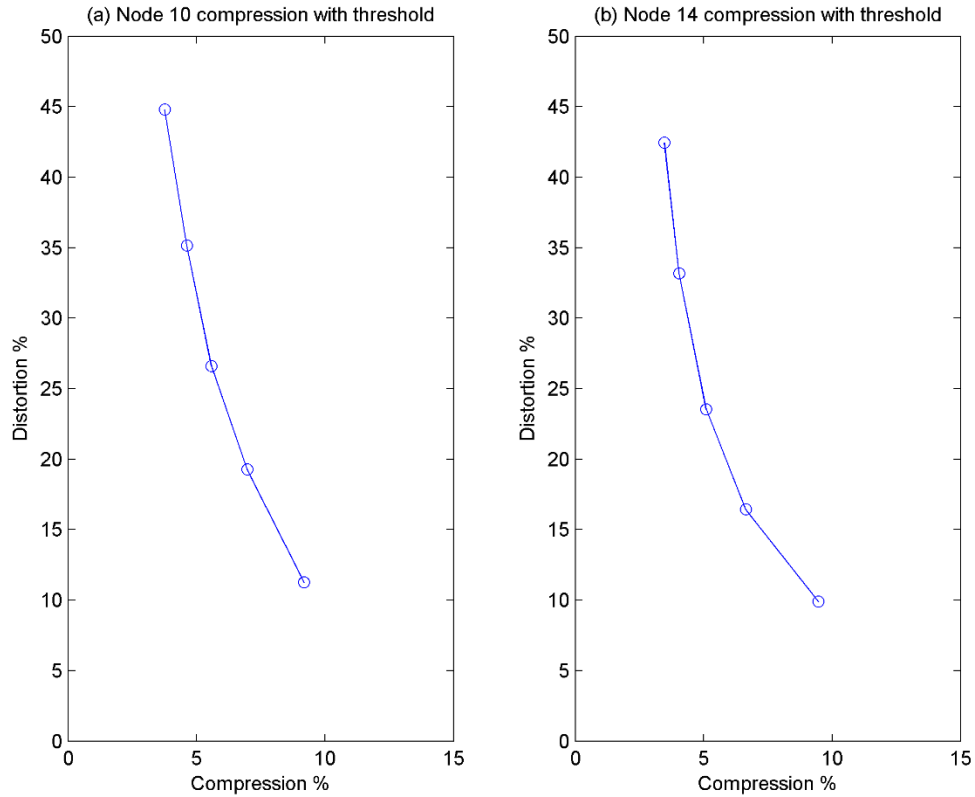


Figure 5.25: (a) Compression at Node 10 with different compression thresholds, (b) similar compression at Node 14 for 5mm leak.

In figure 5.25 for similar compression settings the performance is plotted and the trend is similar to figure 5.23. For distortion between 10 – 15% for most of the signals threshold level of 80% of the coefficient sum and 4 bits SAQ setting can be chosen based on the results shown in the above figures. Similarly for the low energy sensor a 4-bit SAQ scheme is shown to give a distortion of ~10% and a compression percentage of ~17%, this is optimal as reducing the number of bits further increases the distortion to ~25% which is fairly high. Next the effect of each energy efficiency technique on the overall energy consumption of the Node energy is discussed. There are 4 Nodes under study in the system, 2 Nodes are nearer to the leak and 2 Nodes are further away from the leak. As discussed previously these are the nodes nearest to the leak as per the network at which the signal would be present the most and the algorithms will be effective in these nodes the most.

5.4. Effect of Algorithms on Energy Consumption

To study the effect of energy efficiency techniques first the basic version of a monitoring scheme will be simulated, in which the data is continuously being sampled and sent to the sink for analysis, this is most similar to a case in literature of PipeTECT [9] in which near continuous data was streamed from the server in real time with approximately 80% duty cycle, in this simulation for a similar setting the data was sampled for 5s, the medians calculated and it was transmitted which approximately took 7s, further 3s the node was asleep and after 10s the cycle repeated for a 70% duty cycle rate. In this scenario the energy consumption was plotted for without leak for 12000s and the result is shown.

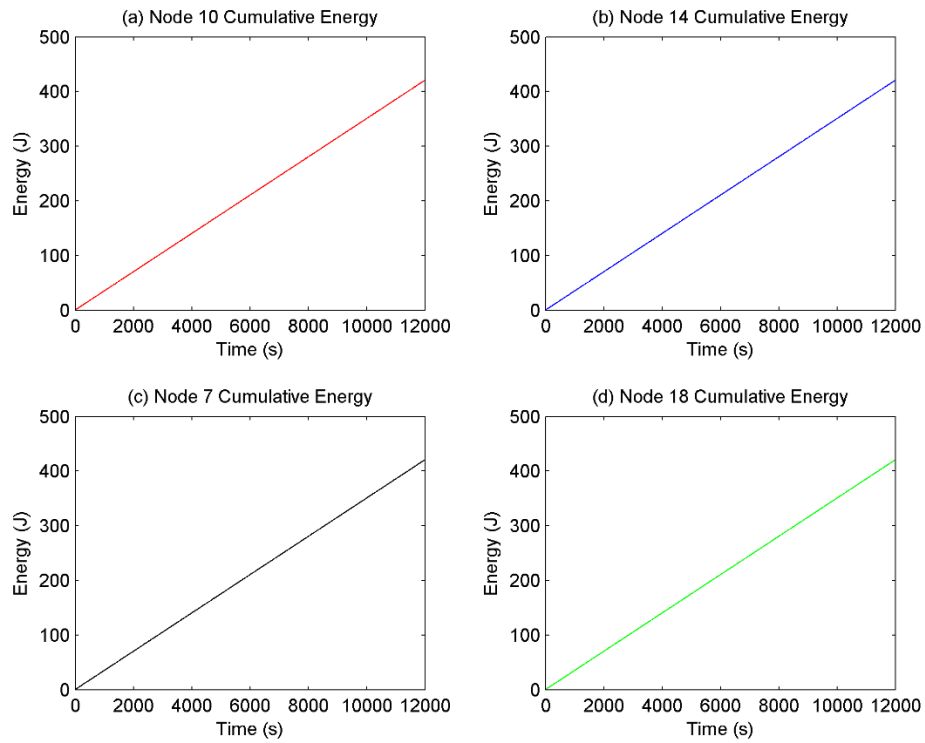


Figure 5.26: Energy consumption over time for 4 Nodes under study for 10s duty cycling.

Figure 5.26 shows the energy consumption over time for all the four nodes under study and the energy consumption of such a scheme is constant over time for each as the sampling

rate is constant, the complete data is also being transmitted, the energy consumption is approximately 420.5 Joules in such a case. Applying duty cycling to such a case, where the sampling is also duty cycled and the transmission is also duty cycled, a case is considered in which the sensor samples the vibrations data only every 60s and then transmits it at that time and another case in which the data is transmitted hourly along with the pressure data at that time. The result for the latter case is shown in figure 5.27.

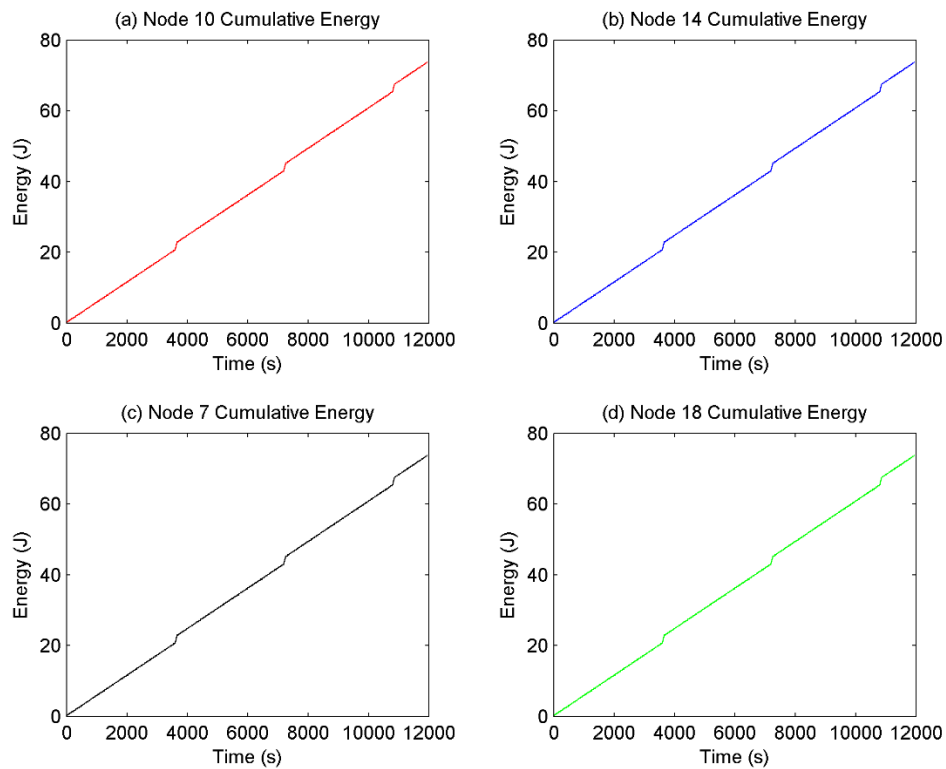


Figure 5.27: Energy consumption over time for 4 Nodes under study for 60s sensor duty cycling and 3600s radio duty cycling.

In the result shown in figure 5.27 gives significant energy savings over the basic version simulated before, the energy cumulative consumption over 12000s is approximately 73.75 Joules for a 10% duty cycle rate and this equates to 82.5% decrease in energy consumption. It should be noted that such a scheme would however give a delay in leak detection and localization as the sampling is done after every 60s instead of after every 10s in the previous

case. Furthermore the hierarchical sampling based adaptive sampling scheme was applied to the system in which the data is being sampled by a higher energy sensor and the frequency is calculated after which the lower energy sensor samples the data in the following sampling instances, this is expected to save further energy as the data will be sampled at lower rate for a longer period of time than being sampled at high frequency all the time. Figure 5.28 shows the results when the low energy sensor was used to sample every 9 cycles after the high energy sensor had sampled the data and calculated the sampling frequency. First the 1mm case was considered.

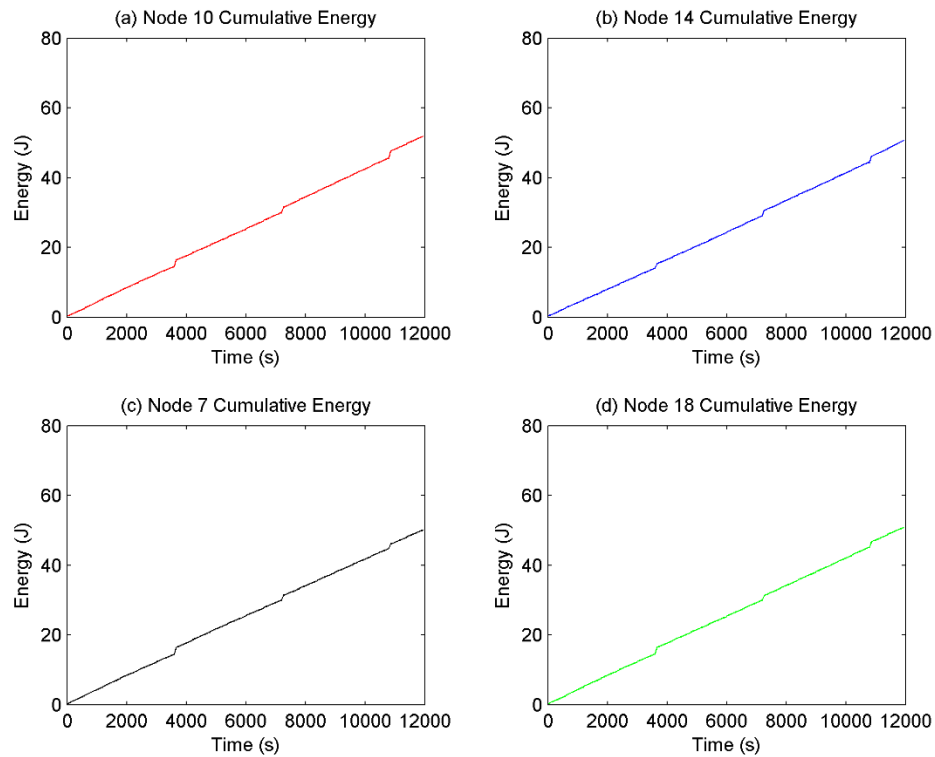


Figure 5.28: Energy consumption over time for 4 Nodes under study for a 9 cycle adaptive sampling scheme for 1mm leak.

Here it can be observed that the energy consumption has dropped further down keeping the same conditions as the previous case shown in figure 5.27. The average energy over the 4 nodes being monitored is 50.9 Joules, this represents a 31% reduction in energy

consumption over the previous scheme, here it is observed that as further energy conservation schemes are being applied to the system the rate at which the energy is saved is dropping, this is mainly due to the fact that adaptive sampling is a CPU intensive operation, but due to the CPU taking less energy than the sensor per time it gives a net gain in energy. Moving on, for similar sampling scheme setting the savings in energy for the 3mm and 5mm leak are also plotted and shown below.

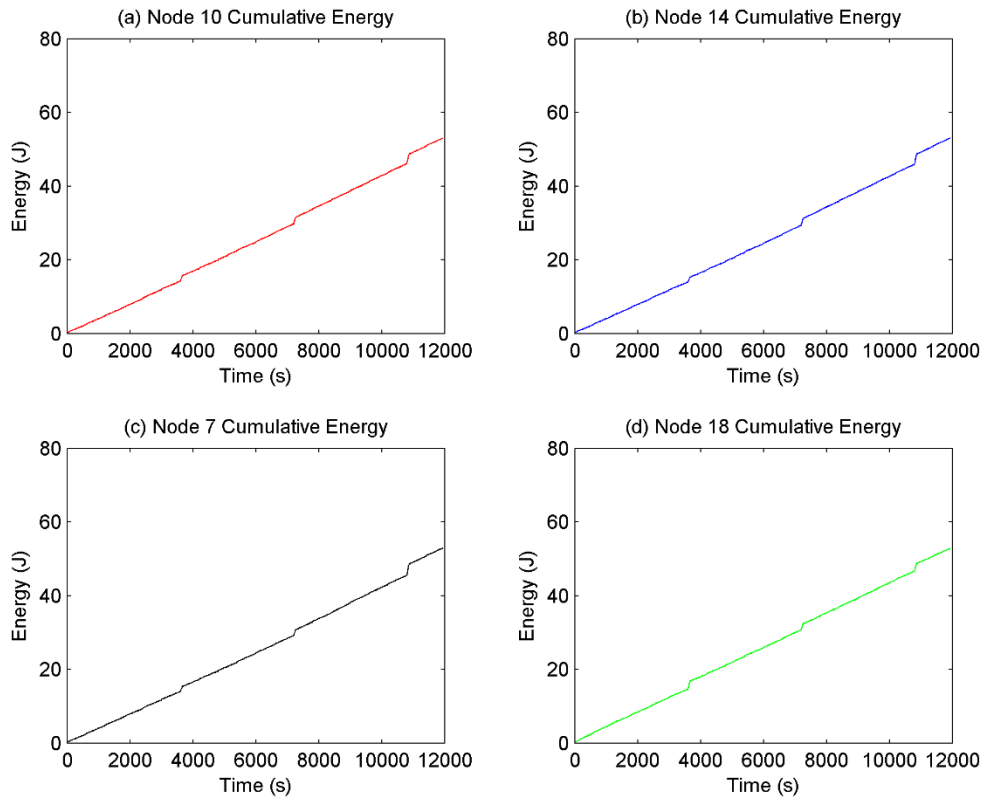


Figure 5.29: Energy consumption over time for 4 Nodes under study for a 9 cycle adaptive sampling scheme for 3mm leak.

In figure 5.29 the energy consumption is plotted over time, the leak is occurring at 5000s so the leak is present for approximately 58% of the simulation duration. The average energy consumption of the nodes is 53 Joules which is still an energy conservation of 28% over the previous non-adaptive scheme, next the 5mm leak is plotted.

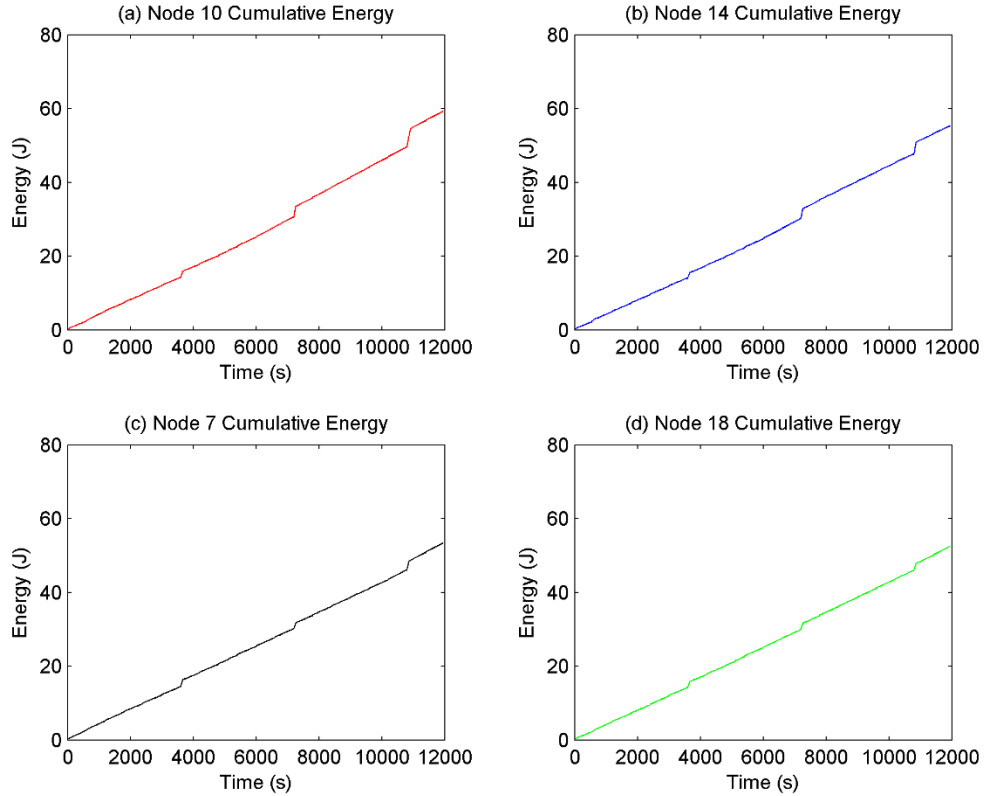


Figure 5.30: Energy consumption over time for 4 Nodes under study for a 9 cycle adaptive sampling scheme for 5mm leak.

The 5mm leak is the final case because it represents a maximum in what the adaptive sampling algorithm can set as the sampling frequency for the system which is 2000Hz. Even here it can be observed that the energy consumption over time is below the average consumption of the non-adaptive case, which is 55.17 Joules, this is again, 25% below the energy consumption of the previous scheme. This then concludes the part where energy conservation schemes were applied to the sensor of the system, for further energy conservation the radio energy is conserved by applying the compression scheme to reduce the energy consumption further. First the 1mm leak case was considered in which the 80% threshold and 4-bit SAQ scheme was applied as discussed in the previous section.

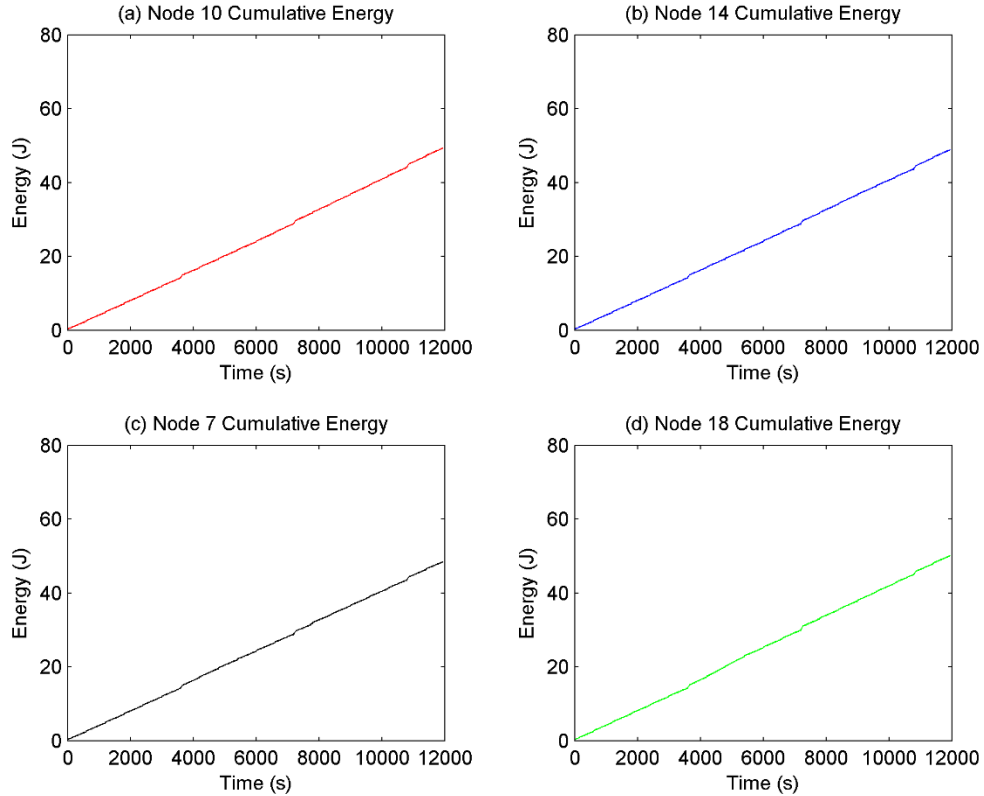


Figure 5.31: Energy consumption over time for 4 Nodes under study for compression applied using 80% threshold and 4-bit SAQ for 1mm leak.

In figure 5.31 the results in figure 5.28 have the compression algorithm applied to them, this results in reduction of energy when transmitting data from 50.9 Joules to 49.2 Joules, this is a very low change in energy considering the conditions because at this time the data is already reduced by a fair margin as vibration magnitudes are not changing significantly, still there is a 3.34% reduction in energy from the previous scheme and gives a net gain in terms of energy conserved. This change is expected to be higher in leaks of larger magnitude. The energy consumption when the algorithm is applied to a 3mm leak is shown in figure 5.32 next.

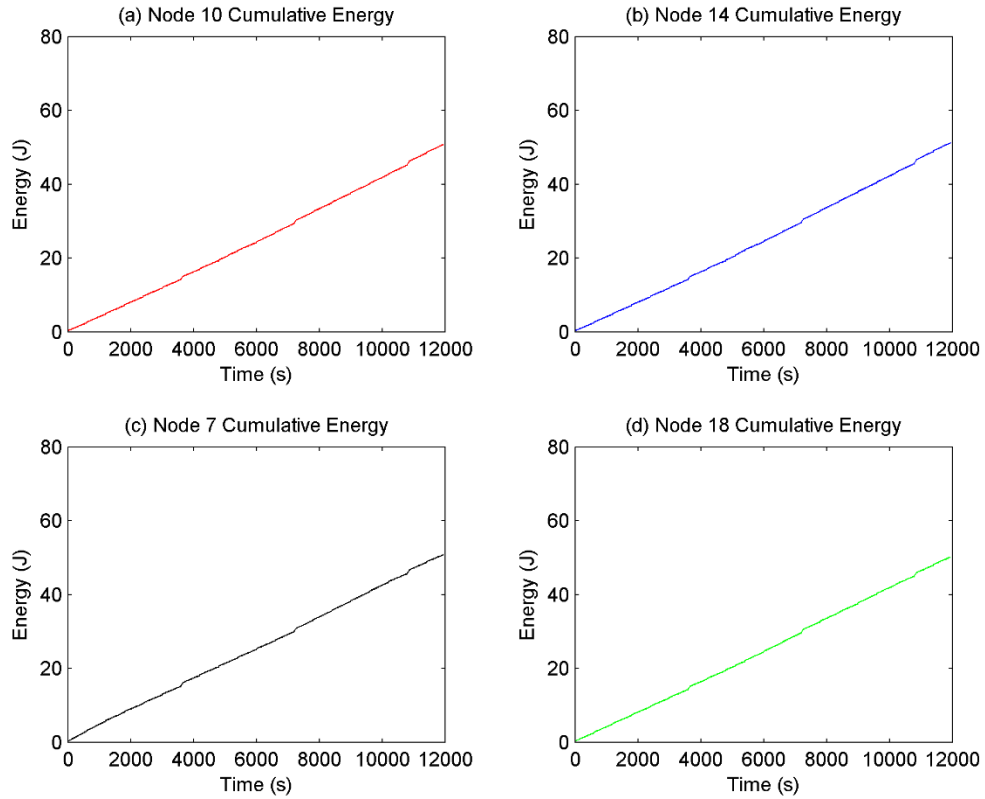


Figure 5.32: Energy consumption over time for 4 Nodes under study for compression applied using 80% threshold and 4-bit SAQ for 3mm leak.

In figure 5.32 the compression algorithm is applied to the results of figure 5.29 and it can be seen that the energy consumption has reduced to an average of 50.8 Joules which is less than the 53 Joules before, this represents a 4.15% reduction in energy consumption compared to the previous case. Next the 5mm leak case is plotted and the results plotted.

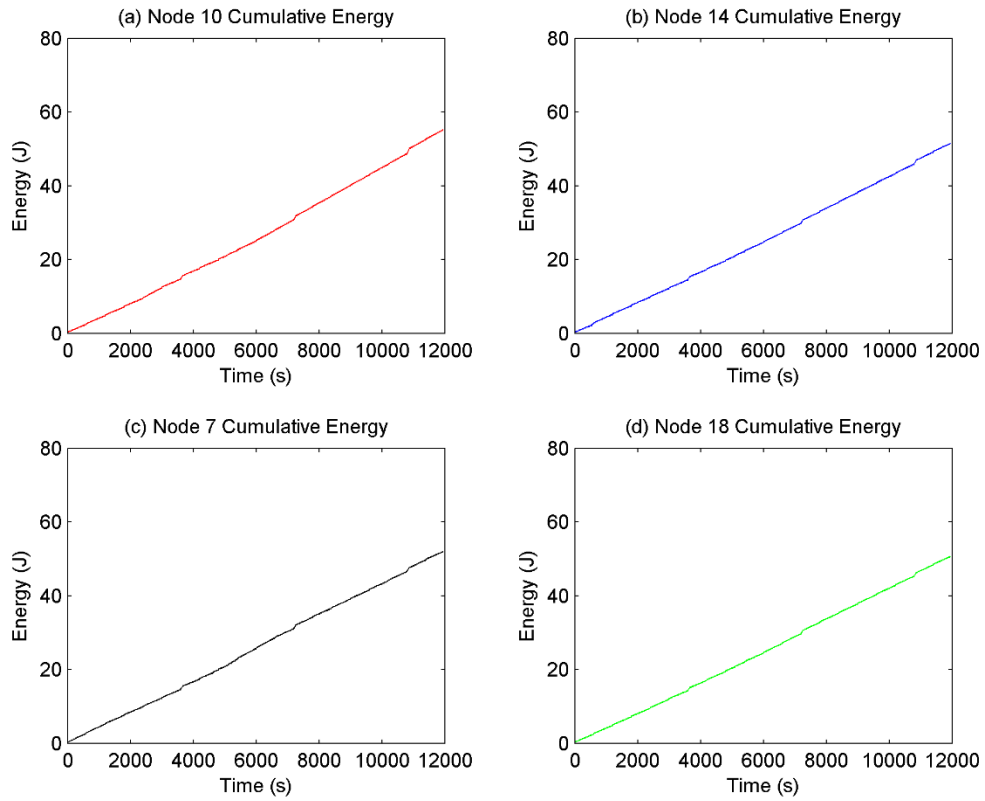


Figure 5.33: Energy consumption over time for 4 Nodes under study for compression applied using 80% threshold and 4-bit SAQ for 5mm leak.

In this case the average energy of the system drops to 52.6 Joules which represents a 4.65% reduction in energy over the previous case. Moving on the percentage energy conserved by each step is plotted in a graph for each leak.

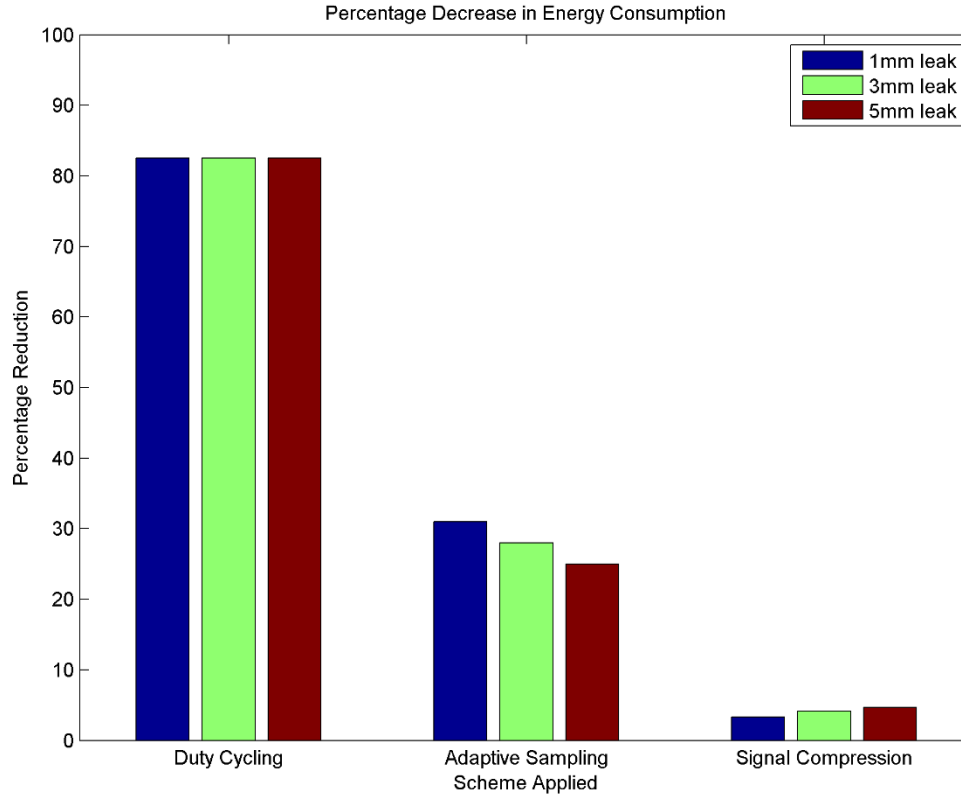


Figure 5.34: Percentage reduction in energy consumption across all energy conservation schemes for different leak sizes.

The next section deals with the effect of the compression ratio on the leak localization which will be the final step in the algorithm.

5.5. Effect of Compression on leak localization

The final part of the algorithm is the leak localization which is done by applying cross-correlation to the vibration signal from two sensors across the leak location, the leak is found by finding the difference in time one signal takes to reach the other and using the known sensor distance and the speed of sound in the water the leak can be localized using the formula shown below:

$$d_2 = \frac{d - cT_0}{2} \quad (54)$$

Here c is the speed of sound propagation in pipe and T_0 is the time delay at which the sensor correlation reaches its peak value. The time delay T_0 is estimated at the peak value of the cross correlation function between the two signals measured by the sensors on either side. The vibration data for leaks is already generated in Chapter 3 of the thesis, and the results for energy and algorithm performance shown in the chapter above are for the same signals. Leak localization is done by cross-correlating data from Nodes 10 and 14 as these are the nearest to the leak location, in fact the main advantage of deploying a WSN to such a pipeline network would be the benefit it provides in having readily available data from nodes nearest to the leak which would enable localization in such a scenario. Before applying the cross correlation the signal is high pass filtered [20] to remove low frequency noise and improve the cross-correlation result. First the cross-correlation result of the 1mm leak is plotted, it is expected to not give a result as the leak signature is minimal at the sampling nodes.

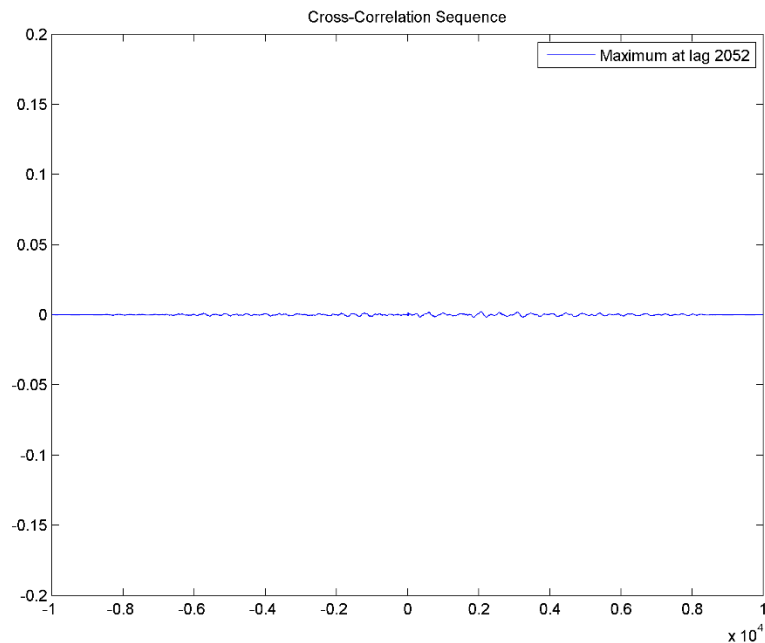


Figure 5.35: Cross-correlation result of 1mm leak for Nodes 10 and 14.

Next the 3mm leak case is considered and the cross correlation peak plotted.

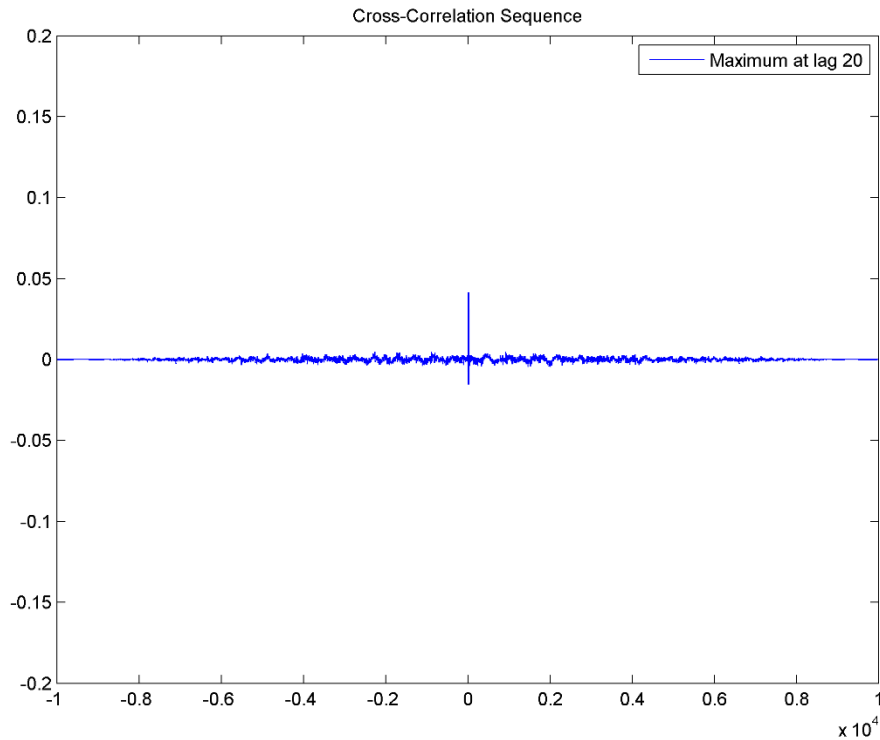


Figure 5.36: Cross-correlation result of 3mm leak.

From the result in figure 5.36 the leak was localized at 22.9645m from Node 10 using equation (44), in the simulation it is set at 23m. The maximum value of the peak is 0.0415 and it is 8.95 times higher than the next highest value in the cross-correlation series. Next different compression ratios that were previously discussed applied to the same signal and the effect of compression on the peak of the cross-correlation examined for the 3mm leak case.

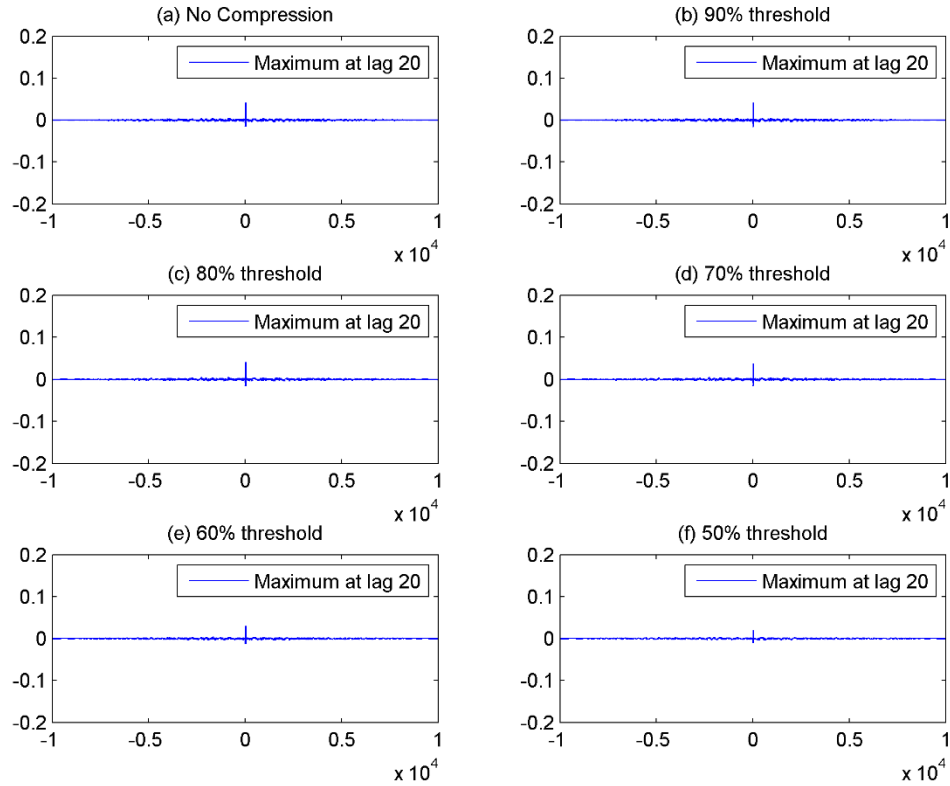


Figure 5.37: Effect of compression on the cross-correlation peak for 3mm leak.

In figure 5.37 the effect of compression on the cross-correlation peak is described, the peak is decreasing in magnitude from figure 5.37(a) to (f). In 5.37(b) the peak remains the same at 0.0415, in (c) it decreases to 0.402, in (d) it is 0.0369 at 70% threshold setting, in (e) it is 0.0295 and in (f) it is 0.0198 which is still 5 times higher than the next highest value. From the results it can be concluded that 90% threshold settings produces the least amount of difference in the result but it may not be the best compression scheme as the 80% threshold compresses the signal further but still manages to be within 95% of the maximum possible value of the cross correlation peak, interestingly even at 70% threshold setting the peak is at 88% of the maximum possible threshold and gives an attractive alternate to a no compression scheme based system. Next results were plotted for the 5mm leak and are

shown in figure 5.38. The results are similar to the 3mm leak case but the peak is much higher due to the fact that the signal magnitude is higher.

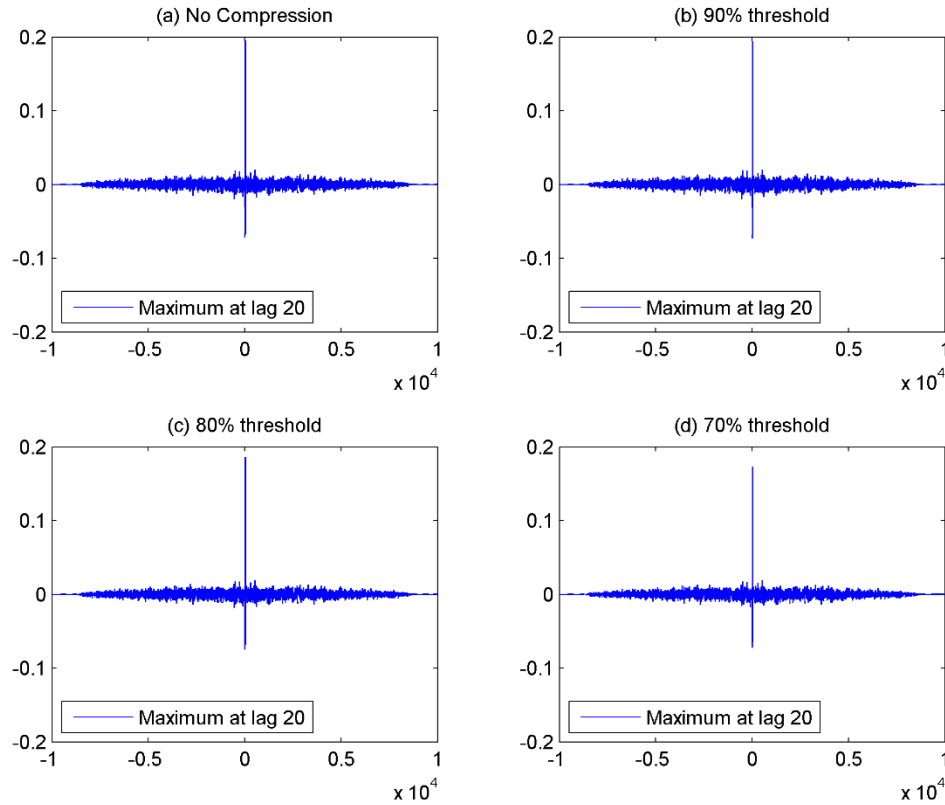


Figure 5.38: Effect of compression on the cross-correlation peak for 5mm leak.

The maximum peak value in this case is 0.1951 which is 9.7 times higher than the next highest value in the correlation series and as the compression threshold level is reduced the signal magnitude decreases to 0.1937 in (b) and 0.1862 in (c), in (d) it is 0.1731 which is again 88% of the original value of the peak. The results suggest from the purely leak localization based standpoint the vibration signal can be compressed up to 50% threshold level but this brings about high levels of signal distortion which would make the signal potentially not suitable for other types of analysis which might be applied, in such a case a threshold level of 80% is best suited for the requirements of signal compression and leak localization as it retains more than 95% value of the original peak of the cross-correlation

as well as provides reasonable amount of compression and keeps the distortion below 20% for almost all the signals under study. Finally the sensitivity of the scheme to drops in packets is compared for different packet losses and the effect on the peak computed. It is expected the distortion of the signal will increase and it would result in a reduction in the peak of the result. This is shown in the figure below.

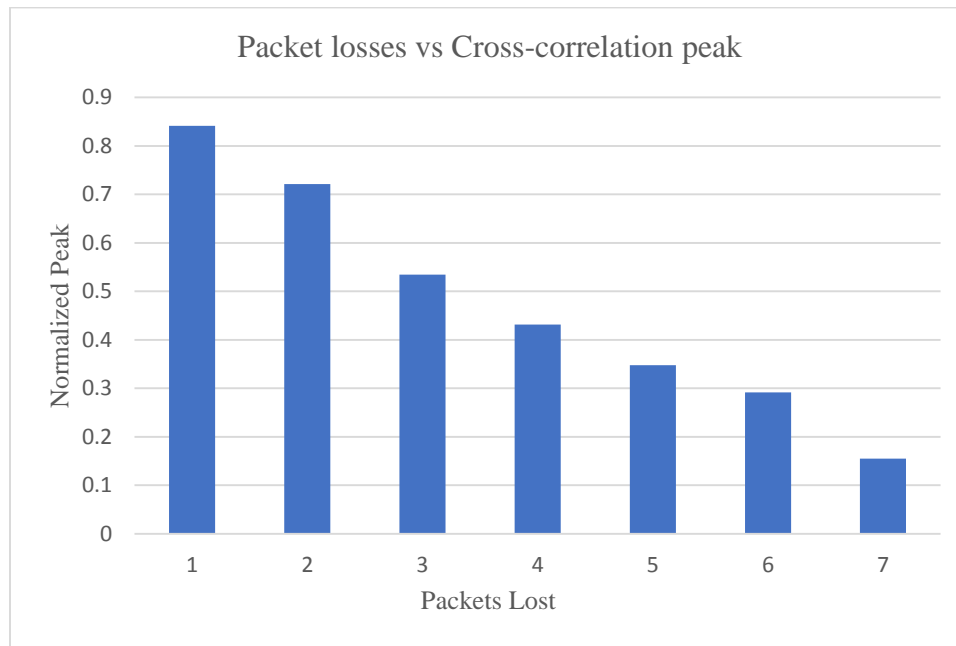


Figure 5.39: Reduction in normalized cross-correlation peak with different packet losses.

In figure 5.39 the effect of packet losses on the cross-correlation peak was investigated. Usually it was found the number of packets in which the signal content was around 10 packets. That way if one packet was lost around 10% information was lost from the wavelet coefficients as these are transmitted to the sink. All the results are for 80% threshold and 4 bit SAQ setting. Random packets were dropped for 5 cases to see the effect on the peak and the average value is shown for each number of packets dropped. It can be seen that the Normalized value of the cross-correlations reduces as the number of packets dropped increases. But it was found that even if 6 packets were dropped the cross-correlation result

gave a correct value of the location of the leak ~23m at a reduced peak of 40% of the original value. But at 7 packets the leak localization result gave erroneous readings of 10m, which were 13m from the actual leak location. The peak is also diminished to around 15% but it is at wrong position. So for environments in which the packet loss rate is high the scheme will eventually give a wrong result for the leak location. It should be noted that the scheme only uses 18 to 19 packets to transmit all the signal data including the coefficients and their positions, so 7 packets represents an approximate 35% packet loss rate which is fairly high. Keeping the results in the previous sections under consideration with regards to energy conservation techniques as well as leak detection and localization times a cost analysis of the whole scheme is done in the next sub-section.

5.6. Cost Analysis of the Scheme

In this section the comparative cost analysis of the leak detection scheme proposed above is presented. Previously the energy savings comparison was carried out and it was found that all three energy saving techniques in the proposed scheme and algorithm parameters were successful in reducing the energy consumption of the overall system to ~12% of the original scheme. This reduction in energy however comes at a disadvantage to other aspects of the system, in this case the leak detection and localization times of the leak detection scheme have increased as well as the signal received at the sink is distorted by up to 20% compared to the original signal depending on the conditions.

Taking an example of T_t of 60 seconds and different resampling period n of 5, 10 and 15 sampling instances and a threshold of 80% and 4 bits for the SAQ for a 3mm leak the following result can be plotted for the 4 Nodes under study.

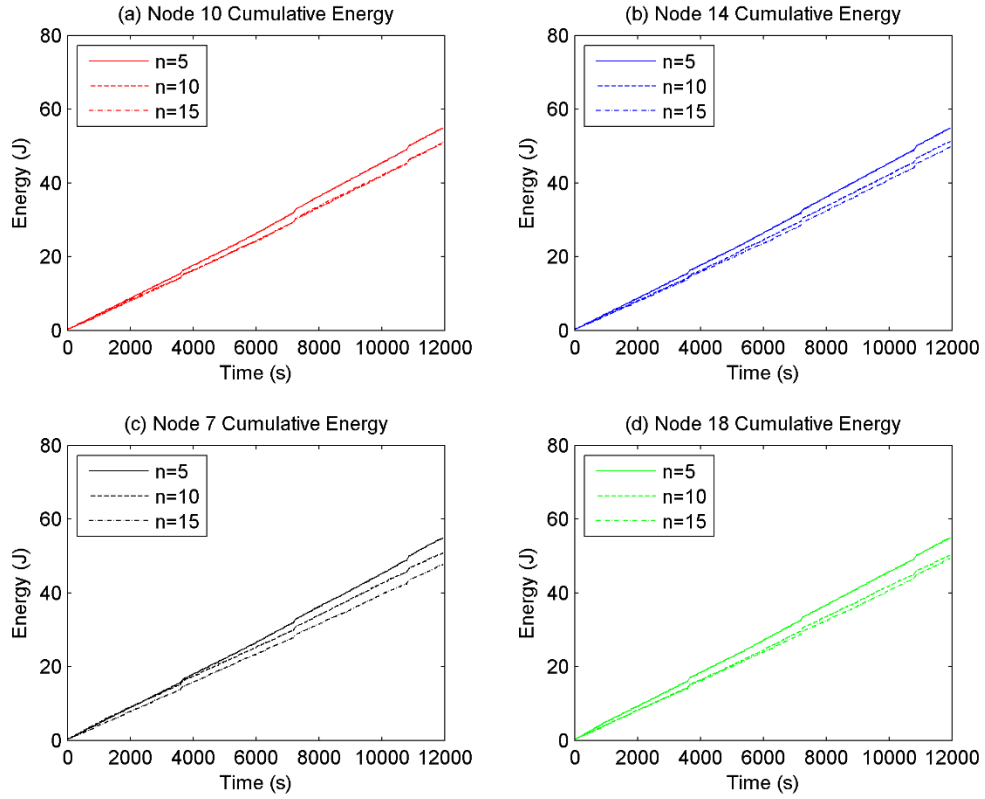


Figure 5.40: Cumulative energy comparison between the three adaptive sampling schemes with varying n .

In figure 5.40 the effect of varying the adaptive resampling period can be clearly observed, as per the settings the leak detection time is the same at 60s for each scheme and leak confirmation will be 120s but the localization times will be minimum of 300s, 600s and 900s for each scheme. Next the effect of increasing the duty cycle period while maintaining T_L at 600 will be observed.

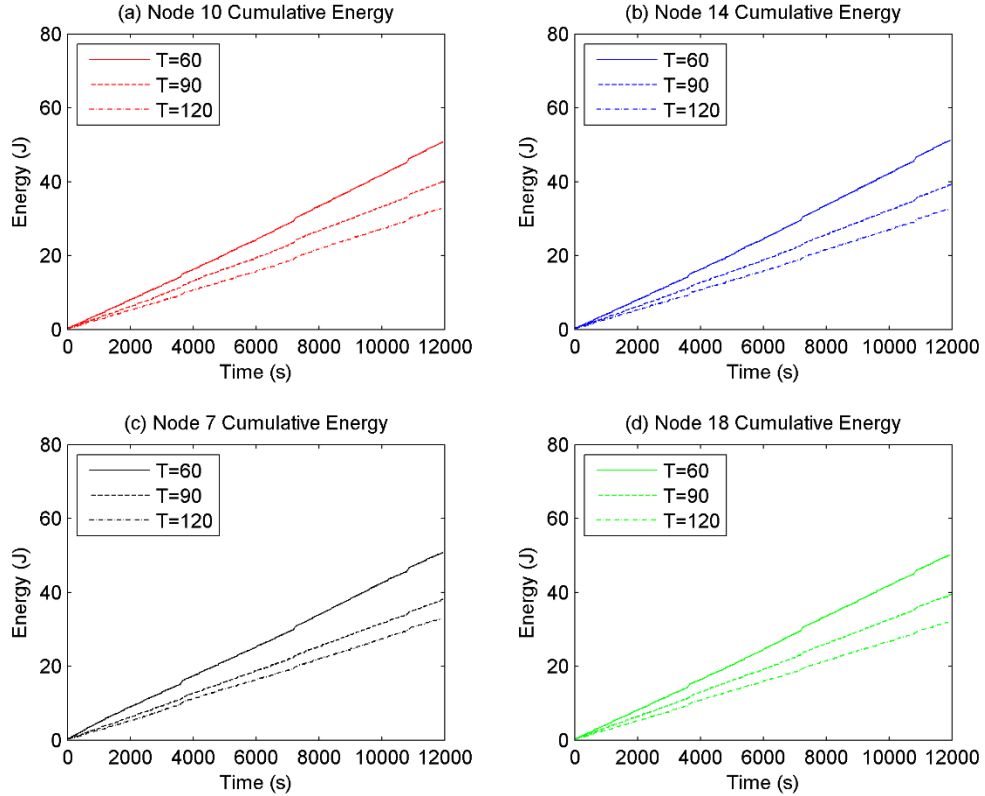


Figure 5.41: Effect of increasing duty cycle on energy consumption for fixed T_L .

In figure 5.41 the effect of increasing the duty cycle time on the energy consumption for individual nodes is observed, this would affect the most on the energy consumption due to the fact that the sampling instances would be reduced and consequently the data for processing and transmission would be reduced as well. It is observed that the change in energy is much greater changing the duty cycle time as compared to increasing the adaptive sampling resampling instances. Because of this reason this is optimized last in the algorithm proposed in this section, also this would increase the leak detection time and in our algorithm the leak detection time would be minimized if it is last. Lastly the effect of compression on the energy consumption performance is observed. The duty cycle is set at 60s with n equal to 10 and different compression ratios are applied to the system.

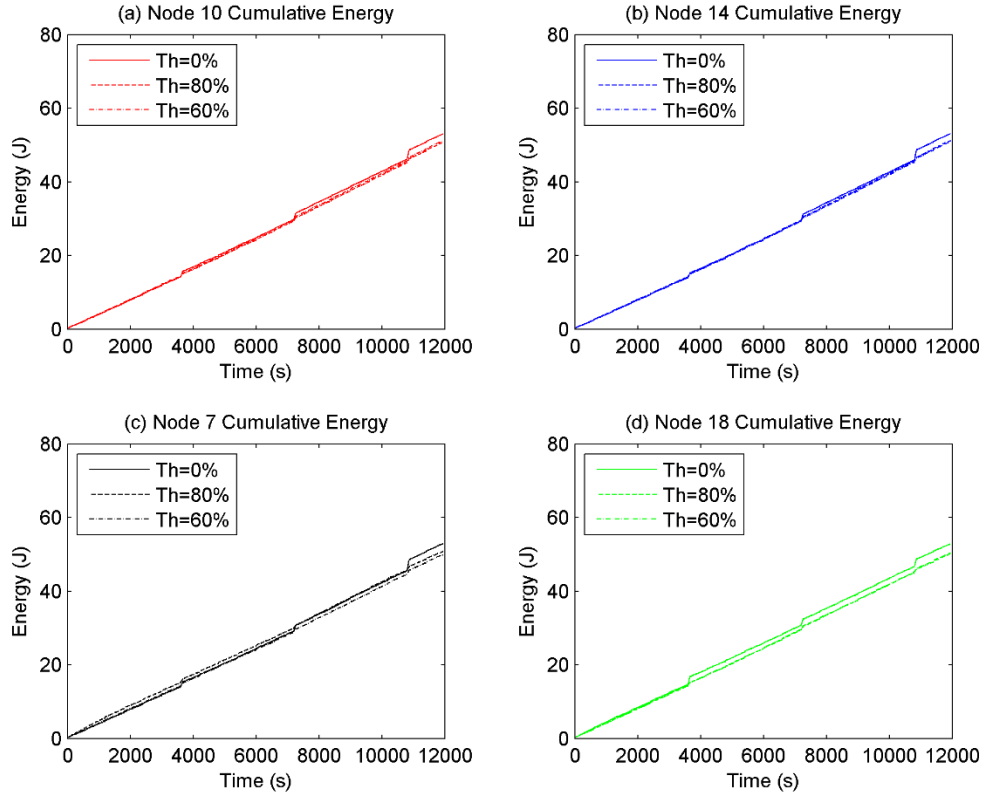


Figure 5.42: Effect of compression on cumulative energy consumption for the nodes under study.

In figure 5.42 the effect of the compression for the nodes under study is observed, it is seen that over time the energy consumption of the nodes with compression is lower than when no compression is applied. Also different compression levels for the time under simulation did not generate sufficient differences in energy consumption for the small time duration simulated, but over a longer period of time 60% threshold will have a marginally lower energy consumption than 80% threshold. This warrants the compression to be first one to be optimized as per the algorithm as it effects the energy consumption the least but still gives a net positive gain in energy savings. Lastly the effect of the leak on the detection and localization times due to different schemes is shown in the next figure along with the leak detection indications.

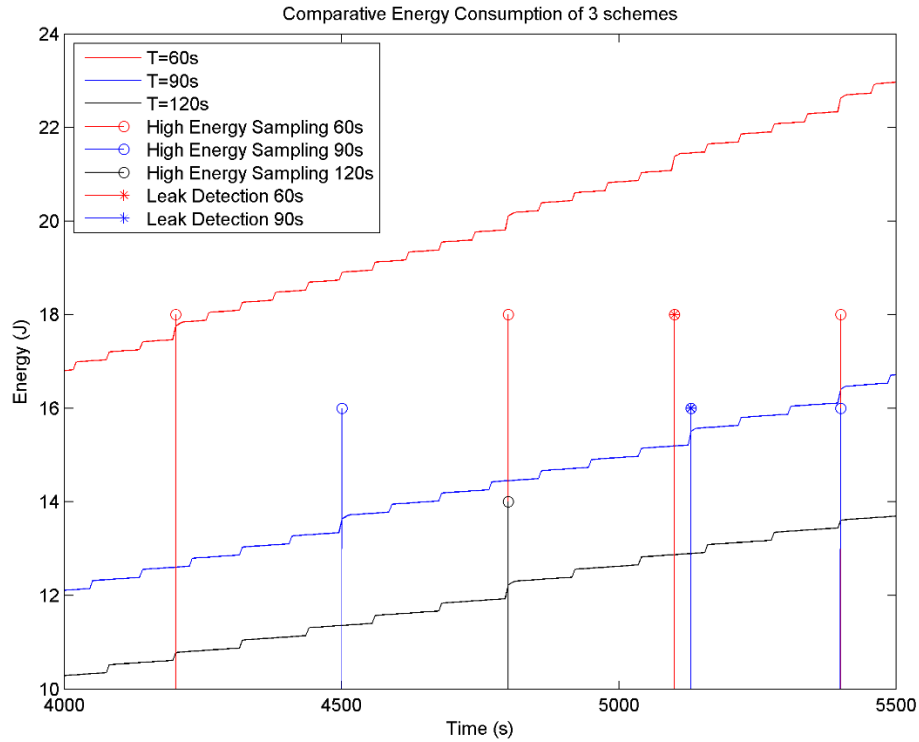


Figure 5.43: Effect of leak detection with different duty cycle times.

In figure 5.43 the sampling by high energy sensors is indicated for three different duty cycle times of 60s, 90s and 120s for a resampling ratio of 10 for the adaptive sampling algorithm. This means that every 600s, 900s and 1200s the adaptive sampling algorithm will use the high frequency sensor. This is also shown in the figure above by the stem plot of the high frequency sampling instances. It is also observed that after 5000s when the leak occurs the 60s and 90s duty cycle schemes sample the high frequency sensor earlier than the resampling duration, this is done when the low energy sensors detect the leak and the high energy sensors in the next cycle are activated to confirm the leak. This way within 2 sampling cycles the leak detection occurs or 120s and 180s in these cases. For the 120s duty cycling period the leak is detected later in the next high frequency sampling instance as the leak occurs within the low frequency sensor sensing period, as a result the worst case

performance of the algorithm is also known for leak detection which is 1200s or the whole resampling time. The leak detection is shown in the figure below.

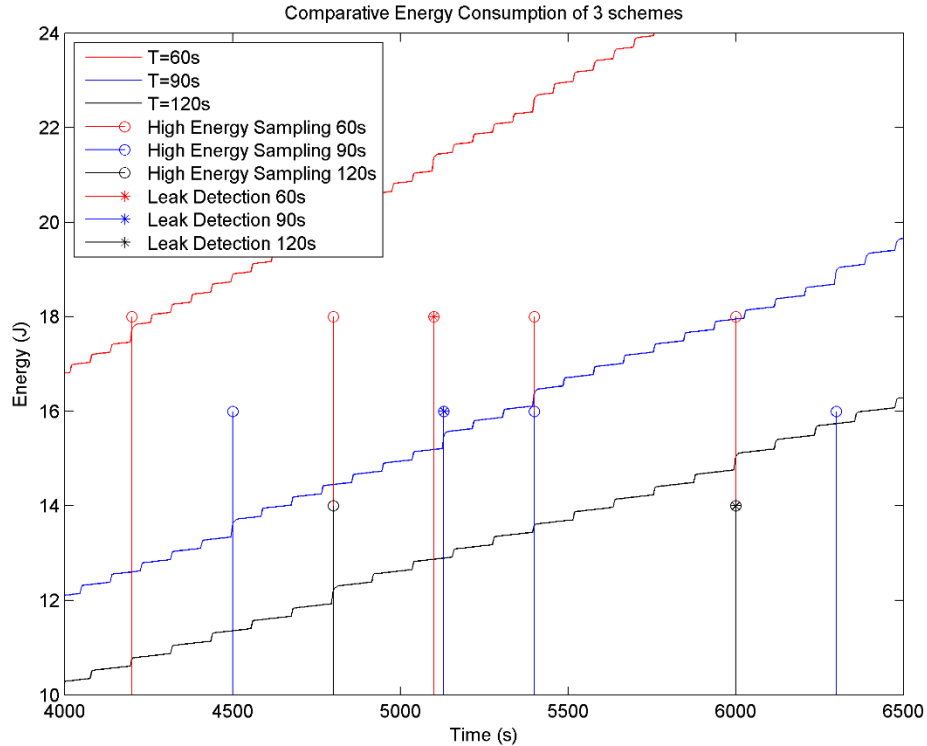


Figure 5.44: Worst case leak detection of 120s scheme.

In figure 5.44 the leak detection is within the 1200s worst case leak detection time though this can be rectified by using a different threshold setting than the one proposed in this work.

After the leak has been confirmed as indicated in figures 5.43 and 5.44 the leak confirmation indication is transmitted at that instance of the high frequency sampling instance. This way the sink knows about the leak and at the next high frequency cycle the sink requests the data from the node.

A solving method is introduced in chapter 4.6 for giving the optimal monitoring conditions for a given energy consumption and leak detection time. Using the method some scenarios

were run to solve the cost function to give us the parameters, for example our simulation is running for 12000s so we take the 12000s case for solving the cost function.

Table 5-1: Solutions given by optimization algorithm for different target energies

Parameter Type	Case No.	Case 1	Case 2	Case 3
	Parameter			
Input Constraints	Target Energy	50	40	60
	Maximum T_D	300s	300s	300 s
	Maximum T_L	1200s	1200s	1200s
	Maximum PRD_A	25%	25%	25%
Algorithm Outputs	Calculated T_D	60s	80s	50s
	Resampling Period n	9	8	7
	Calculated Threshold	86%	79%	91%
	Calculated Bits	3 bits	3 bits	3 bits

In Table 5-1 the results for 3 different cases are discussed. Case 1 is for 50 Joules case, it was found that the algorithm returned a duty cycle time of 60s and a resampling period of 9, with an 86% threshold and 3 bits for compression. This result closely matches the simulation parameters used in the simulation. Similarly for 40 Joules and 60 Joules target energy it was seen that the calculated duty cycle time is at 80s and 50s respectively. From the results it is apparent that given the constraints the most amount of change in energy consumption is due to the duty cycling scheme and for different energy consumption requirements in changes the most.

This wraps up the discussion of the proposed solver which minimizes the leak detection and localization times based on given energy. Next the conclusion is discussed.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

In this work we have introduced a novel multi-scale leak detection and localization scheme which uses vibration measurements for detection and localization of leaks in a pipeline network. This solution can be used without extensive knowledge of the network and is fairly accurate under optimal conditions, opposed to other techniques which would require extensive computation as well as relatively more knowledge of the network under study.

In the simulation, using our leak detection and localization algorithm, we have shown that leak can be effectively detected and localized by the vibration data using the novel multi-scale approach we are using here to solve the problem using sensors of different sensitivities. In our approach a duty cycling-based data-driven approach was used to reduce the energy due to the sampling operation for a WSN node. Furthermore for reducing the sampling energies further an adaptive sampling scheme was applied, the algorithm was able to successfully adapt the sampling rate according to the vibration conditions that are present and reduce energy consumption by sampling at the frequency required based on the bandwidth of the signal. Even though this has been used for different vibration monitoring scenarios, its application to pipeline vibration monitoring for leak detection has brought a significant improvement in energy saving to this important engineering problem. Before transmitting the signal is compressed, reducing the data further and optimizing the radio energy. We have shown that the proposed algorithm can successfully detect leaks in the system. Moreover, this work has also used an energy model for a WSN node for a simulated test-bed system using microcontrollers. Using this model, we have shown that significant savings of up to 88% in energy conservation can be achieved when using the

complete algorithm for the parameters we have used in the shown simulations, with a delay in confirming the leak compared to a non-adaptive approach. Finally the signal is cross-correlated at the sink to localize the leak and the effect of compression on the result is investigated, it was observed that for detectable leaks even at distortion levels of up to 50% the localization can be done effectively. Furthermore the algorithm is shown to be resilient for packet loss rates of up to 30% if the header information is sent correctly.

Based on the work a cost function approach to solving for optimum monitoring parameters is proposed which would allow to set the most appropriate parameters based on the given system conditions and parameters.

This system allows us to have a framework in which a thorough simulation based study can be carried out based on theory. There is much scope for future work in our proposed framework that also allows for the investigation of further optimization techniques for accurate leak localization. It is also noted that we have used a simple model for communication in which the sensor nodes communicate directly with the sink without any hops or a chance for collision in the network. This work can be expanded and applied to a WSN environment with multiple hops to the sink, as this would have an impact on the leak detection speed of response due to the different delays associated with the hops engendered by the protocols being used. Also it was observed there can be improvements in the detection scheme by carrying out trend analysis of high frequency vibration signals to detect leaks. This analysis can be carried out at the sink. Different compression algorithms can also be applied to the problem to further optimize the solution.

References:

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 40–50, 2002.
- [3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "How to prolong the lifetime of wireless sensor networks," *Mob. Ad Hoc Pervasive Commun.*, pp. 1–26, 2006.
- [4] I. Jawhar, N. Mohamed, and K. Shuaib, "A framework for pipeline infrastructure monitoring using wireless sensor networks," in *Wireless Telecommunications Symposium, 2007. WTS 2007*, 2007, pp. 1–7.
- [5] K. El-Darymli, F. Khan, and M. H. Ahmed, "Reliability modeling of wireless sensor network for oil and gas pipelines monitoring," *Sensors & Transducers*, vol. 106, no. 7, p. 6, 2009.
- [6] I. Stoianov, L. Nachman, S. Madden, T. Tokmouline, and M. Csail, "PIPENET: A wireless sensor network for pipeline monitoring," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, 2007, pp. 264–273.
- [7] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, and C. Shahabi, "SWATS: Wireless sensor networks for steamflood and waterflood pipeline monitoring," *IEEE Netw.*, vol. 25, no. 1, pp. 50–56, 2011.
- [8] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava, "NAWMS: nonintrusive autonomous water monitoring system," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 309–322.
- [9] M. Shinozuka *et al.*, "Nondestructive monitoring of a pipe network using a MEMS-based wireless network," *Proc. SPIE - Nondestruct. Charact. Compos. Mater. Aerosp. Eng. Civ. Infrastructure, Homel. Secur.*, vol. 7649, p. 76490P, 2010.
- [10] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [11] S. Oven, "Leak Detection in Pipelines by the use of State and Parameter Estimation Sindre Oven," 2014.
- [12] J. D. Blotter and A. G. Stephens, "Flow rate measurements using flow-induced pipe vibration," 2004.
- [13] M. T. Pittard, R. P. Evans, R. D. Maynes, and J. D. Blotter, "Experimental and numerical investigation of turbulent flow induced pipe vibration in fully developed flow," *Rev. Sci. Instrum.*, vol. 75, no. 7, pp. 2393–2401, 2004.

- [14] A. Martini, M. Troncossi, A. Rivola, and D. Nascetti, “Advances in Condition Monitoring of Machinery in Non-Stationary Operations,” *Lect. Notes Mech. Eng.*, vol. 5, pp. 535–544, 2014.
- [15] L. Molina-Espinosa, C. Verde-Rodarte, and O. Cazarez-Candia, “Mathematical model for pipeline leak simulation,” in *Experimental and Theoretical Advances in Fluid Dynamics*, Springer, 2012, pp. 303–311.
- [16] D. De Silva, J. Mashford, and S. Burn, “Computer Aided Leak Location and Sizing in Pipe Networks Urban Water Security Research Alliance Technical Report No. 17. Urban Water Security Research Alliance Technical Report NO. 17, 2011(17).de,” *Urban Water Secur. Res. Alliance Tech. Rep. No. 17*, vol. 2011, no. 17, 2011.
- [17] G. Ye and R. A. Fenner, “Kalman Filtering of Hydraulic Measurements for Burst Detection in Water Distribution Systems,” *J. Pipeline Syst. Eng. Pract.*, vol. 2, no. 1, pp. 14–22, 2011.
- [18] Y. Ishido and S. Takahashi, “A new indicator for real-time leak detection in water distribution networks: Design and simulation validation,” *Procedia Eng.*, vol. 89, pp. 411–417, 2014.
- [19] F. Almeida, M. Brennan, P. Joseph, S. Whitfield, S. Dray, and A. Paschoalini, “On the acoustic filtering of the pipe and sensor in a buried plastic water pipe and its effect on leak detection: an experimental investigation,” *Sensors (Basel)*, vol. 14, no. 3, pp. 5595–5610, 2014.
- [20] Y. Gao, M. J. Brennan, P. F. Joseph, J. M. Muggleton, and O. Hunaidi, “A model of the correlation function of leak noise in buried plastic pipes,” *J. Sound Vib.*, vol. 277, no. 1, pp. 133–148, 2004.
- [21] Y. Gao, M. J. Brennan, P. F. Joseph, J. M. Muggleton, and O. Hunaidi, “On the selection of acoustic/vibration sensors for leak detection in plastic water pipes,” *J. Sound Vib.*, vol. 283, no. 3–5, pp. 927–941, 2005.
- [22] M. J. Brennan, F. K. de Lima, F. C. L. de Almeida, P. F. Joseph, and A. T. Paschoalini, “A virtual pipe rig for testing acoustic leak detection correlators: Proof of concept,” *Appl. Acoust.*, vol. 102, pp. 137–145, 2016.
- [23] A. S. Papastefanou, P. F. Joseph, and M. J. Brennan, “Experimental investigation into the characteristics of in-pipe leak noise in plastic water filled pipes,” *Acta Acust. united with Acust.*, vol. 98, no. 6, pp. 847–856, 2012.
- [24] Y. a. Khulief, a. Khalifa, R. Ben Mansour, and M. a. Habib, “Acoustic Detection of Leaks in Water Pipelines Using Measurements inside Pipe,” *J. Pipeline Syst. Eng. Pract.*, vol. 3, no. 2, pp. 47–54, 2012.
- [25] A. Martini, M. Troncossi, and A. Rivola, “Vibration Monitoring as a Tool for Leak Detection in Water Distribution Networks,” *Ciri Din*, pp. 1–9, 2014.
- [26] A. Santos and M. Younis, “A sensor network for non-intrusive and efficient leak detection in long pipelines,” *IFIP Wirel. Days*, vol. 1, no. 1, 2011.

- [27] A. S. Almazayad *et al.*, “A proposed scalable design and simulation of wireless sensor network-based long-distance water pipeline leakage monitoring system,” *Sensors (Switzerland)*, vol. 14, no. 2, pp. 3557–3577, 2014.
- [28] C. Ok *et al.*, “Optimal transmission power in self-sustainable sensor networks for pipeline monitoring,” *Proc. 3rd IEEE Int. Conf. Autom. Sci. Eng. IEEE CASE 2007*, pp. 591–596, 2007.
- [29] Y. Guo, F. Kong, D. Zhu, A. S. Tosun, and Q. Deng, “Sensor Placement for Lifetime Maximization in Monitoring Oil Pipelines,” *Proc. 1st ACM/IEEE Int. Conf. Cyber-Physical Syst. - ICCPS '10*, pp. 61–68, 2010.
- [30] M. Mysorewala, M. Sabih, L. Cheded, M. T. Nasir, and M. Ismail, “A Novel Energy-Aware Approach for Locating Leaks in Water Pipeline Using a Wireless Sensor Network and Noisy Pressure Sensor Data,” *Int. J. Distrib. Sens. Networks*, vol. 2015, 2015.
- [31] H. Mustafa and P. H. Chou, “Embedded damage detection in water pipelines using wireless sensor networks,” *Proc. 14th IEEE Int. Conf. High Perform. Comput. Commun. HPCC-2012 - 9th IEEE Int. Conf. Embed. Softw. Syst. ICESS-2012*, pp. 1578–1586, 2012.
- [32] M. Zorzi and R. R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance,” *IEEE Trans. Mob. Comput.*, vol. 2, no. 4, pp. 349–365, 2003.
- [33] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 70–84.
- [34] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” *Wirel. networks*, vol. 8, no. 5, pp. 481–494, 2002.
- [35] A. Cerpa and D. Estrin, “ASCENT: Adaptive self-configuring sensor networks topologies,” *IEEE Trans. Mob. Comput.*, vol. 3, no. 3, pp. 272–285, 2004.
- [36] P. Godfrey and D. Ratajczak, “Naps: scalable, robust topology management in wireless ad hoc networks,” in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 443–451.
- [37] C. Schurgers, V. Tsiatsis, and M. B. Srivastava, “STEM: Topology management for energy efficient sensor networks,” in *Aerospace Conference Proceedings, 2002. IEEE*, 2002, vol. 3, p. 3.
- [38] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy-latency-density design space,” *IEEE Trans. Mob. Comput.*, vol. 99, no. 1, pp. 70–80, 2002.
- [39] X. Yang and N. H. Vaidya, “A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay,” in *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th*

IEEE, 2004, pp. 19–26.

- [40] E. Shih, P. Bahl, and M. J. Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices,” in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 160–171.
- [41] L. Gu and J. A. Stankovic, “Radio-triggered wake-up for wireless sensor networks,” *Real-Time Syst.*, vol. 29, no. 2–3, pp. 157–182, 2005.
- [42] A. Keshavarzian, H. Lee, and L. Venkatraman, “Wakeup scheduling in wireless sensor networks,” *Proc. seventh ACM Int. Symp. Mob. ad hoc Netw. Comput. - MobiHoc '06*, p. 322, 2006.
- [43] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TinyDB: an acquisitional query processing system for sensor networks,” *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, 2005.
- [44] P. Buonadonna, D. Gay, J. M. Hellerstein, W. Hong, and S. Madden, “TASK: Sensor network in a box,” in *Proceedings of the Second European Workshop on Wireless Sensor Networks, EWSN 2005*, 2005, vol. 2005, pp. 133–144.
- [45] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.
- [46] T. van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” *Proc. 1st Int. Conf. Embed. networked Sens. Syst. (SenSys '03)*, pp. 171–180, 2003.
- [47] B. Hohlt, L. Doherty, and E. Brewer, “Flexible power scheduling for sensor networks,” *Proc. third Int. Symp. Inf. Process. Sens. networks - IPSN'04*, p. 205, 2004.
- [48] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, “Delay efficient sleep scheduling in wireless sensor networks,” in *Proceedings - IEEE INFOCOM*, 2005, vol. 4, pp. 2470–2481.
- [49] IEEE Computer Society, *Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, vol. 2007, no. June. 2007.
- [50] R. Zheng, J. C. Hou, and L. Sha, “Asynchronous wakeup for ad hoc networks,” *Proc. 4th ACM Int. Symp. Mob. ad hoc Netw. Comput. MobiHoc 03*, pp. 35–45, 2003.
- [51] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, and S. S. Iyengar, “Random asynchronous wakeup protocol for sensor networks,” in *Proceedings - First International Conference on Broadband Networks, BroadNets 2004*, 2004, pp. 710–717.
- [52] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC protocols for wireless sensor networks: a survey,” *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, 2006.

- [53] K. Arisha, M. Youssef, and M. Younis, "Energy-aware TDMA-based MAC for sensor networks," *Syst. power Optim. Wirel. Multimed. Commun.*, pp. 21–40, 2002.
- [54] J. C. Haartsen, "Bluetooth radio system," *IEEE Pers. Commun.*, vol. 7, no. 1, pp. 28–36, 2000.
- [55] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," *Proc. first Int. Conf. Embed. networked Sens. Syst. SenSys 03*, vol. 12, no. 1, p. 181, 2003.
- [56] V. Rajendran, J. J. Garcia-Luna-Aceves, and K. Obraczka, "Energy-efficient, application-aware medium access for sensor networks," in *2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2005*, 2005, vol. 2005, pp. 623–630.
- [57] L. F. W. H. Van, P. J. M. Havinga, and L. F. W. Van Hoesel, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switches," *1st Int. Work. Networked Sens. Syst. INSS 2004*, pp. 205–208, 2004.
- [58] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," *Proc. 2nd Int. Conf. Embed. networked Sens. Syst. SenSys 04*, vol. 3, no. 4, p. 95, 2004.
- [59] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, 2004, p. 224.
- [60] I. Standard, *Part 15.4e: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, vol. 2012, no. April. 2012.
- [61] A. Ephremides and O. A. Mowafi, "Analysis of a Hybrid Access Scheme for Buffered Users-Probabilistic Time Division," *IEEE Trans. Softw. Eng.*, vol. SE-8, no. 1, pp. 52–61, 1982.
- [62] I. Rhee, A. Warriar, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, 2008.
- [63] I. H. Hou, Y. E. Tsai, T. F. Abdelzaher, and I. Gupta, "AdapCode: Adaptive network coding for code updates in wireless sensor networks," in *Proceedings - IEEE INFOCOM*, 2008, pp. 2189–2197.
- [64] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "IN-NETWORK AGGREGATION TECHNIQUES FOR WIRELESS SENSOR NETWORKS: A SURVEY," *IEEE Wirel. Commun.*, no. April, pp. 70–87, 2007.
- [65] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

- [66] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, 2005, vol. 2, pp. 8–13.
- [67] C. P. Antonopoulos and N. S. Voros, "Resource efficient data compression algorithms for demanding, WSN based biomedical applications," *J. Biomed. Inform.*, vol. 59, pp. 1–14, 2016.
- [68] Q. Huang, B. Tang, L. Deng, and J. Wang, "A divide-and-compress lossless compression scheme for bearing vibration signals in wireless sensor networks," vol. 67, pp. 51–60, 2016.
- [69] F. Marcelloni and M. Vecchio, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *Comput. J.*, vol. 52, no. 8, pp. 969–987, 2009.
- [70] M. Oltean, J. Picheral, E. Lahalle, H. Hamdan, and J. Griffaton, "Compression methods for mechanical vibration signals: Application to the plane engines," *Mech. Syst. Signal Process.*, vol. 41, no. 1–2, pp. 313–327, 2013.
- [71] N. Aloui, S. Bousselmi, and A. Cherif, "Speech Compression Based on Discrete Walsh Hadamard Transform," *Int. J. Inf. Eng. Electron. Bus.*, vol. 5, no. 3, pp. 59–65, 2013.
- [72] R. Stojanović, S. Knežević, D. Karadaglić, and G. Devedžić, "Optimization and implementation of the wavelet based algorithms for embedded biomedical signal processing," *Comput. Sci. Inf. Syst.*, vol. 10, no. 1, pp. 503–523, 2013.
- [73] N. Aloui, S. Bousselmi, and A. Cherif, "Optimized Speech Compression Algorithm Based on Wavelets Techniques and its Real Time Implementation on DSP," *Int. J. Inf. Technol. Comput. Sci.*, vol. 7, no. 3, pp. 33–41, 2015.
- [74] D. Chu, a Deshpande, J. M. Hellerstein, and W. Hong, "Approximate Data Collection in Sensor Networks using Probabilistic Models," *Data Eng. 2006. ICDE '06. Proc. 22nd Int. Conf.*, p. 48, 2006.
- [75] A. Jain, E. Y. Chang, and Y. F. Wang, "Adaptive stream resource management using Kalman Filters," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2004, pp. 11–22.
- [76] B. Kanagal and A. Deshpande, "Online filtering, smoothing and probabilistic modeling of streaming data," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, 2008, pp. 1160–1169.
- [77] D. Tulone and S. Madden, "PAQ: Time series forecasting for approximate query answering in sensor networks," in *European Workshop on Wireless Sensor Networks*, 2006, pp. 21–37.
- [78] D. Tulone and S. Madden, "An energy-efficient querying framework in sensor networks for detecting node similarities," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, 2006, pp. 191–300.

- [79] Y.-A. Le Borgne, S. Santini, and G. Bontempi, “Adaptive model selection for time series prediction in wireless sensor networks,” *Signal Processing*, vol. 87, no. 12, pp. 3010–3020, 2007.
- [80] S. Goel and T. Imielinski, “Prediction-based monitoring in sensor networks: taking lessons from MPEG,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 5, pp. 82–98, 2001.
- [81] S. Goel, A. Passarella, and T. Imielinski, “Using buddies to live longer in a boring world [sensor network protocol],” in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, 2006, p. 5–pp.
- [82] Q. Han, S. Mehrotra, and N. Venkatasubramanian, “Energy efficient data collection in distributed sensor environments,” in *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, 2004, pp. 590–597.
- [83] G. Simon *et al.*, “Sensor network-based countersniper system,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 1–12.
- [84] G. Werner-Allen *et al.*, “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, 2006.
- [85] V. Raghunathan, S. Ganeriwal, and M. Srivastava, “Emerging techniques for long lived wireless sensor networks,” *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 108–114, 2006.
- [86] C. Alippi and G. Anastasi, “Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications,” *...Adhoc Sens. ...*, pp. 0–5, 2007.
- [87] A. Jain and E. Y. Chang, “Adaptive sampling for sensor networks,” *Proceedings 1st Int. Work. Data Manag. Sens. networks conjunction with VLDB 2004 - DMSN '04*, no. Dmsn, p. 10, 2004.
- [88] R. Willett, A. Martin, and R. Nowak, “Backcasting: adaptive sampling for sensor networks,” in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, 2004, pp. 124–133.
- [89] I. F. Akyildiz, M. C. Vuran, and O. B. Akan, “On exploiting spatial and temporal correlation in wireless sensor networks,” *Proc. WiOpt*, vol. 4, pp. 71–80, 2004.
- [90] J. Zhou and D. De Roure, “Floodnet: Coupling adaptive sampling with energy aware routing in a flood warning system,” *J. Comput. Sci. Technol.*, vol. 22, no. 1, pp. 121–130, 2007.
- [91] B. Schott, M. Bajura, J. Czarnaski, J. Flidr, T. Tho, and L. Wang, “A modular power-aware microsensor with > 1000x dynamic power range,” in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005, p. 66.
- [92] A. Prati, R. Vezzani, L. Benini, E. Farella, and P. Zappi, “An integrated multi-modal

- sensor network for video surveillance,” in *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, 2005, pp. 95–102.
- [93] T. Kijewski-Correa, M. Haenggi, and P. Antsaklis, “Wireless sensor networks for structural health monitoring: a multi-scale approach,” in *ASCE Structures 2006 Congress*, 2006.
- [94] Y.-C. Tseng, Y.-C. Wang, K.-Y. Cheng, and Y.-Y. Hsieh, “iMouse: an integrated mobile surveillance and wireless sensor system,” *Computer (Long. Beach. Calif.)*, vol. 40, no. 6, 2007.
- [95] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks,” in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 588–599.
- [96] B. Gedik, L. Liu, and S. Y. Philip, “ASAP: An adaptive sampling approach to data collection in sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 12, pp. 1766–1783, 2007.
- [97] P. Padhy, R. K. Dash, K. Martinez, and N. R. Jennings, “A utility-based sensing and communication model for a glacial sensor network,” *Proc. fifth Int. Jt. Conf. Auton. agents multiagent Syst.*, pp. 1353–1360, 2006.
- [98] D. B. Steffelbauer and D. Fuchs-Hanusch, “Efficient Sensor Placement for Leak Localization Considering Uncertainties,” *Water Resour. Manag.*, vol. 30, no. 14, pp. 5517–5533, 2016.
- [99] H. V Fuchs and R. Riehle, “Ten years of experience with leak detection by acoustic signal analysis,” *Appl. Acoust.*, vol. 33, no. 1, pp. 1–19, 1991.
- [100] O. Hunaidi and W. T. Chu, “Acoustical characteristics of leak signals in plastic water distribution pipes,” *Appl. Acoust.*, vol. 58, pp. 235–254, 1999.
- [101] L. A. Rossman, “The EPANET programmer’s toolkit for analysis of water distribution systems,” in *WRPMD’99: Preparing for the 21st Century*, 1999, pp. 1–10.
- [102] J. M. Muggleton, M. J. Brennan, and R. J. Pinnington, “Wavenumber prediction of waves in buried pipes for water leak detection,” *J. Sound Vib.*, vol. 249, no. 5, pp. 939–954, 2002.
- [103] J. M. Muggleton, M. J. Brennan, and P. W. Linford, “Axisymmetric wave propagation in fluid-filled pipes: wavenumber measurements in in vacuo and buried pipes,” *J. Sound Vib.*, vol. 270, no. 1, pp. 171–190, 2004.
- [104] F. Almeida, M. Brennan, P. Joseph, S. Whitfield, S. Dray, and A. Paschoalini, “On the acoustic filtering of the pipe and sensor in a buried plastic water pipe and its effect on leak detection: An experimental investigation,” *Sensors*, vol. 14, no. 3, pp. 5595–5610, 2014.
- [105] Y. Gao, M. J. Brennan, and P. F. Joseph, “A comparison of time delay estimators for the detection of leak noise signals in plastic water distribution pipes,” *J. Sound*

Vib., vol. 292, no. 3–5, pp. 552–570, 2006.

- [106] Y. Munson, *Okiishi, and Huebsch*. 2010.
- [107] D. Eliades, “EPANET MATLAB Toolkit,” *Univ. Cyprus, Repub. Cyprus*, 2009.
- [108] M. J. Kendall and C. R. Siviour, “Rate dependence of poly (vinyl chloride), the effects of plasticizer and time–temperature superposition,” in *Proc. R. Soc. A*, 2014, vol. 470, no. 2167, p. 20140012.
- [109] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, “PIPENETa wireless sensor network for pipeline monitoring,” *Proc. 6th Int. Conf. Inf. Process. Sens. networks - IPSN '07*, p. 264, 2007.
- [110] M. Rizwan and I. D. Paul, “Leak Detection in Pipeline System Based on Flow Induced Vibration Methodology in Pipeline,” *Int. J. Sci. Res. ISSN (Online Index Copernicus Value Impact Factor)*, vol. 14, no. 4, pp. 2319–7064, 2013.
- [111] G. Owojaiye and Y. Sun, “Focal design issues affecting the deployment of wireless sensor networks for pipeline monitoring,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1237–1253, 2013.
- [112] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, “Energy and bandwidth-efficient Wireless Sensor Networks for monitoring high-frequency events,” *2013 IEEE Int. Conf. Sensing, Commun. Networking, SECON 2013*, pp. 194–202, 2013.
- [113] T. Kurp, R. X. Gao, and S. Sah, “An adaptive sampling scheme for improved energy utilization in wireless sensor networks,” *2010 IEEE Instrum. Meas. Technol. Conf. Proc.*, pp. 93–98, 2010.
- [114] H.-Y. Zhou, “Modeling of Node Energy Consumption for Wireless Sensor Networks,” *Wirel. Sens. Netw.*, vol. 3, no. 1, pp. 18–23, 2011.
- [115] B. Latré, P. De Mil, I. Moerman, B. Dhoedt, P. Demeester, and N. Van Dierdonck, “Throughput and delay analysis of unslotted IEEE 802.15. 4,” *J. networks*, vol. 1, no. 1, pp. 20–28, 2006.
- [116] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Trans. image Process.*, vol. 1, no. 2, pp. 205–220, 1992.
- [117] D. L. Donoho and J. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.
- [118] D. K. Masanobu Shinozuka Pai H. Chou, Sehwan Kim, Hong Rok Kim dan Lu Fei, “Non-invasive Acceleration-Based Methodology for Damage Detection and Assessment of Water Distribution System,” *Smart Struct. Syst.*, vol. 6, no. 10, pp. 545–559, 2010.
- [119] Z. Qu, H. Feng, Z. Zeng, J. Zhuge, and S. Jin, “A SVM-based pipeline leakage detection and pre-warning system,” *Measurement*, vol. 43, no. 4, pp. 513–519, 2010.
- [120] C. C. E. Leiserson, R. R. L. Rivest, C. Stein, and T. H. Cormen, *Introduction to Algorithms, Third Edition*, vol. 7. 2009.

- [121] D. E. Knuth, "Dynamic huffman coding," *J. algorithms*, vol. 6, no. 2, pp. 163–180, 1985.
- [122] J. G. Kolo, S. A. Shanmugam, D. W. G. Lim, L.-M. Ang, and K. P. Seng, "An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks," *J. Sensors*, vol. 2012, pp. 1–20, 2012.
- [123] M. Sabarimalai Sur and S. Dandapat, "Wavelet-based electrocardiogram signal compression methods and their performances: A prospective review," *Biomed. Signal Process. Control*, vol. 14, no. 1, pp. 73–107, 2014.
- [124] R. Benzid, F. Marir, A. Boussaad, M. Benyoucef, and D. Arar, "Fixed percentage of wavelet coefficients to be zeroed for ECG compression," *Electron. Lett.*, vol. 39, no. 11, pp. 830–831, 2003.
- [125] R. Benzid, F. Marir, and N.-E. Bouguechal, "Electrocardiogram compression method based on the adaptive wavelet coefficients quantization combined to a modified two-role encoder," *IEEE Signal Process. Lett.*, vol. 14, no. 6, pp. 373–376, 2007.
- [126] B. a Rajoub, A. Alshamali, and A. S. Al-Fahoum, "An efficient coding algorithm for the compression of ECG signals using the wavelet transform.," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 4, pp. 355–62, 2002.
- [127] J. Chen, F. Wang, Y. Zhang, and X. Shi, "ECG compression using uniform scalar dead-zone quantization and conditional entropy coding," *Med. Eng. Phys.*, vol. 30, no. 4, pp. 523–530, 2008.
- [128] K. Sayood, *Introduction to data compression*. Newnes, 2012.
- [129] T. Sheltami, M. Musaddiq, and E. Shakshuki, "Data compression techniques in Wireless Sensor Networks," *Futur. Gener. Comput. Syst.*, vol. 64, pp. 151–162, 2016.
- [130] Y. Zhang and J. Li, "Wavelet-Based Vibration Sensor Data Compression Technique for Civil Infrastructure Condition Monitoring," *J. Comput. Civ. Eng.*, vol. 20, no. 6, pp. 390–399, 2006.
- [131] W. J. Staszewski, "Wavelet based compression and feature selection for vibration analysis," *J. Sound Vib.*, vol. 211, no. 5, pp. 735–760, 1998.
- [132] R. F. Rice, "Some practical universal noiseless coding techniques, part 3, module PS114, K+," 1991.
- [133] M. Iqbal, M. Naeem, A. Anpalagan, A. Ahmed, and M. Azam, "Wireless sensor network optimization: multi-objective paradigm," *Sensors*, vol. 15, no. 7, pp. 17572–17620, 2015.
- [134] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," *Artif. Intell. Prentice-Hall, Egnlewood Cliffs*, vol. 25, p. 27, 1995.

Vitae

Name : NAJAM US SAQIB

Nationality : PAKISTAN

Date of Birth : 10-Jul-1989

Email : najm989@hotmail.com

Address : A-640, Sector 11-B
North Karachi, Karachi
Sindh - Pakistan

Academic Background : M.S. Systems Engineering
KFUPM, Saudi Arabia - May 2017
B.E. Electronics Engineering
NUST-PNEC, Pakistan - Jun 2012

Publications : Najam us Saqib, Muhammad Faizan Mysorewala, and
Lahouari Cheded. "A Multiscale Approach to Leak
Detection and Localization in Water Pipeline
Network." *Water Resources Management* (2017): 1-14.