TOWARDS DISCRETE-PULSE-BASED NETWORKING AND EVENT DETECTION ARCHITECTURES FOR RESOURCE-CONSTRAINED APPLICATIONS

By

Saptarshi Das

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering—Doctor of Philosophy

2019

ABSTRACT

TOWARDS DISCRETE-PULSE-BASED NETWORKING AND EVENT DETECTION ARCHITECTURES FOR RESOURCE-CONSTRAINED APPLICATIONS

By

Saptarshi Das

In this dissertation thesis, we develop a scalable and energy-efficient discrete-pulse-based networking architecture along with a Spiking-Neuron-based low-power detection framework for use in resource-constrained settings. Applications such as Structural Health Monitoring (SHM) using wireless sensor networks powered by ambient energy harvesting are particularly suited for such a framework. The key idea in pulse-based networking is to eschew unnecessary overhead as incurred in traditional packet-based networking and encode only the essential information using small number of discrete pulses and their positions with respect to a synchronized time frame structure. The baseline pulse networking does not scale well with increase in network size. In order to ameliorate this, we develop a scalable time frame structure for use in applications with large network size while preserving the energy advantages of pulse networking. In addition, we stress the importance of judicious use of erratic energy availability in ambient energy harvesting powered systems. To that effect, we build energy-awareness syntaxes within the pulse networking framework for better utilization of energy resources in such systems. We also demonstrate the feasibility of pulse networking over a through-substrate ultrasonic link layer and the advantages thereof in terms of utilizing existing infrastructure and removing the need for radio retrofits. We explore how the protocol performance varies for an airplane stabilizer monitoring application powered by ambient vibration energy harvesting in different energy availability scenarios. Beyond this, we also develop a Spiking-Neuron-based low-power event pattern detection architecture and illustrate how this can be incorporated within a pulse-networked SHM system. The Spiking

Neuron based architecture is evidenced to be simpler in terms of implementation but more efficient in terms of computation and energy usage, thus enabling in-situ detection even at intermediate nodes in the network and robust low-power event pattern detection immune to pulse drifts and errors.

Dedicated to my Maa and Baba.
Thanks for all your patience, love and inspiration.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all family and friends who stood by me during my years as a Ph.D. student. Every little gesture of patience and kindness from you all helped me get farther along on my journey and kept me motivated to work till I could reach my goal.

A lot of thanks go to my advisor Dr. Subir Biswas for supporting and nurturing me throughout my Ph.D. years. I am grateful for his guidance in shaping my research direction as well as preparing me to be a good researcher even beyond my Ph.D. tenure.

I would like to thank my committee members Dr. Mahapatra, Dr. Ren and Dr. Kulkarni as well for their patient support throughout my Ph.D. years, always being available for productive discussions, and helping me perfect my final thesis with their valuable inputs on my research work.

I also want to thank the National Science Foundation (NSF) for partly funding the work done as part of this thesis through the grant CNS 1405273.

Working at the NeEWS lab at MSU has been truly fulfilling during my Ph.D. years, not just because of the varied project experience but also because of the wonderful lab members that I have had the joy of working with over the years, including Debasmit, Dong, Faezeh, Yan, Feng, Rui, Henry, Brandon and Shahrukh. Even when the work was hard, it was a relief to have friends in the lab like you that I could count on and share the experience with.

My friends at MSU outside work have been a constant source of support throughout my years here and I consider them as close as my own family. I believe my Ph.D. experience would not have been half as fulfilling if not for their company through my years at MSU. Special mention

in this regard should be made of my roommates throughout the years – Chetan, Aritra and Preetam, with whom I have shared many wonderful moments during my time at MSU and who have always been there for me in moments happy or sad. I would also like to thank so many other members of my Spartan family from seniors in Debasmit Da, Arko Da, Shreya Di, Faezeh and Yan to peers like Oishi, Chetan, Tridip, Piku, Tias, Soumen, Saptarshi, Portia and juniors in Aritra, Preetam, Sayli, Sabyasachi, Kanchan, Tarang and Yashesh. I have learnt a lot from you all and lived a lot as well thanks to all your wonderful company as I worked toward my Ph.D. Thanks for always being there for me.

I would like to thank my family back in India for being a constant source of motivation even from the other side of the world. Special mention goes to my cousins Rivu, Shaon, Jhini, Sreeshty, Mamon and Mishtu and all my aunts and uncles who have been there for me and my parents as I spent years away from home in pursuit of my Ph.D.

I would also like to thank my elder brother Rajarshi, my sister-in-law Debolina Di and my lovely little niece Raya for their continued inspiration through my Ph.D. studies.

Last, but not the least, I would like to thank my Maa and Baba (parents) for their patience and constant motivation as I worked through my Ph.D. far away from them all these years. You have lived through many hard times but smiled through it all so that I could complete my studies well. I wanted to let you know that this Ph.D. was as much yours as mine and I hope I could make some of your dreams come true as well by completing this.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. Motivation

Applications in diverse domains such as environment monitoring [1]–[3], habitat monitoring [4]–[6], structural monitoring [7]–[10], target tracking [11]–[14], and industrial process control [15]–[17] rely on the collection of granular information across areas of concern using distributed sensing devices. The data from such devices can be collected at a central location for processing and drawing higher level inferences in the context of the application involved. Energy-efficient wireless sensor networks [18]–[21] are well suited for such applications because they allow low-cost deployment and data communication across the sensor nodes as well as to a central Base Station.

Packet-based networking [22]–[24] is the dominant mode of networking for such applications. Various routing strategies and energy-efficiency improvements [25]–[34] have been proposed in the literature to suit such packet-based wireless sensor networking in low-power environments. The latter is the case when the sensing devices are equipped with small batteries or are powered by erratic energy harvesting sources such as wind or ambient vibrations. However, depending on the application involved and data size as well as latency requirements, packet-based networking is not always the most energy-efficient approach. It is to be noted in this regard that most of the energy usage in a typical sensing device for a wireless sensor network is expended in the wireless (radio, ultrasound etc.) transmission/reception/idling costs and less so in actual sensing. Therefore, reducing the former is very important for resource-constrained applications. Also, packet-based networking generally involves a considerable amount of overhead information beyond the actual data payload being transported, which is constituted of components such as the

packet preamble and header. The preamble uses a chosen bit pattern to synchronize transmission timing between senders and receivers in a wireless networking context. This is especially important when long data sequences need to be transported and in an environment with many other wireless devices using the same transmission channel. The packet header contains other information needed to ensure reliable transmission (error detection / correction), addressability, quality of service etc. Thus, the preamble and header are needed to ensure data reliability, transmission synchronization, low latency etc., especially when the Protocol Data Unit (PDU) being transported is long. However, in application instances such as Structural Health Monitoring, the essential data is often just binary event information from distributed sensors and thus very small. In such scenarios, Packet-based networking approaches can often be overkill as overhead costs easily overwhelm actual data networking costs. It has been shown in prior work [35] that discrete-pulse-based networking approaches can be much more energy-efficient compared to packet-based networking approaches when the information content is small, latency requirements are relaxed, and network event rate is not exceedingly high. This can be achieved by using a discrete pulse's position in time within a synchronized time frame structure to encode next-hop information and origin location.

In this thesis, we will focus on applications which have the above-mentioned properties (i.e. low information content, less-stringent latency bounds) and explore several improvements that need to be incorporated to adapt the baseline discrete-pulse-based networking to various relevant application scenarios. The latter includes systems powered by scarce energy harvesting sources as well as large distributed networks where the baseline pulse networking might not easily scale. Specifically, we will develop scalable and energy-aware discrete-pulse-based networking architectures that can be applied in Structural Health Monitoring (SHM) applications. We will also

explore through-substrate networking using ultrasonic communication as a feasible and energy-efficient mode of communication for distributed sensors in an SHM context. It will be shown that discrete-pulse-based networking can be well-adapted to such a communication scenario in a variety of energy availability situations. In addition, we will also consider how to process the communicated information from discrete-pulse-based networking using a low-energy detection architecture to create a holistic energy-efficient communication and detection platform for resource-constrained wireless sensor networks. For the detection purposes, we will demonstrate the use of a Spiking Neuron based architecture for its simplicity in implementation and inherent energy efficiency, yet robust generalizability in detecting spatiotemporal spike patterns. We outline the various research challenges involved in developing such an architecture and how these are being addressed in ongoing work. We also include performance results showing the efficacy of the Spiking Neuron based implementation in realistic application scenarios.

1.2. Application Domain – Structural Health Monitoring

The core objectives in structural health monitoring (SHM) [36]–[38] are to collate information such as unusual stress, faults and cracks from many strategically positioned sensors over a target structure and infer the health of the structure using such information [39]. Energy-efficient wireless sensor networks [7], [8], [40], [41] are often deployed for multi-hop data collection from such sensors to an access-point or a sink, where the collective data processing functions can be placed.

A notable observation is that after a certain amount of local processing at a sensor, often the transportable information from the sensor to an access-point is merely an event. This can be a threshold crossing of local stress, detection of a pre-defined temporal stress pattern, or even a crack in the extreme case. Since the event information is binary (yes or no), a single pulse can be used

to communicate this, thus eliminating the need for packets and their associated overheads such as synchronization preambles. This would lead to significant benefits in energy-starved environments. The low energy usage can also improve reliability of operation in systems powered by energy harvesting, by keeping the consumption rate lower than the energy generation rate. The primary design questions here are how to: 1) transport event localization information using a single pulse, 2) route a pulse multi-hop without explicit node addressing, and 3) provide reliable event delivery even under energy constraints. These problems can be architecturally addressed in a discrete-pulse-based networking framework by integrating a pulse's (i.e., event's) area of origin within a MAC-routing protocol framework and incorporating several energy-awareness syntaxes within the same framework as will be discussed in later sections in this thesis.

Thus, Structural Health Monitoring can be a good application instance for showcasing the advantages of discrete-pulse-based networking and discussing the changes needed for such an architecture to adapt it to specific application needs such as extreme resource constraint or scalability issues. A low-energy detection architecture would also be very useful for interpreting the communicated event information in the form of discrete pulses, especially if it can easily be interfaced with the networked pulses. Spiking Neuron based architectures can be useful in this regard because of their inherent low-complexity design yet robust generalizability and will be developed as an event pattern detection solution in this thesis.

1.3. Scalability in Discrete-Pulse-Based Networking

The central foundation of the discrete-pulse-based architecture is a synchronized Frame Structure employed to control pulse transmission schedules such that the time of transmission can encode various information aspects that are useful to the application context. The size of such a frame is known to be related to the number of network cells [35], [42]. More precisely, the frame

size scales quadratically with increase in the number of network cells. It is also known that the frame size restricts the event delivery latency and event throughput of the network. Since, the core premise of discrete-pulse-based networking is to trade off latency (i.e. event delivery delay) and throughput for energy benefits, this works well for small to medium-sized networks if the limits for allowable delay and throughput are relaxed. In applications where such delay / throughput limits are more restrictive, but network cell count scaling needs to be supported in addition to energy efficiency, the frame structure does not adapt well. Network cell count might need to be increased to enhance maximum sensing localization resolution for the network or when the network area is increased. Hence, there is need for some redesign to adapt the baseline discrete-pulse-based networking approach for a more generic and scalable scenario as will be developed in this thesis.

1.4. Energy-Aware Discrete-Pulse-Based Networking

Various modern approaches employ sensing devices powered by energy harvested from the ambience e.g. solar power, vibration harvesting etc. Though harvesting sources have theoretically infinite potential, they can be exceedingly erratic in the short term. This can be shown to affect networking performance significantly in a baseline discrete-pulse-based networking approach. This is because, if networking is unaware of energy availability, transmission can often be wasteful when economy is warranted, leading to intermittent network failure due to power loss. Often, some amount of energy-awareness incorporated in the networking protocol can significantly improve network uptime while not sacrificing network throughput. We try to demonstrate this with our own improvements to the baseline discrete-pulse-based networking architecture. We also evaluate how such performance is affected for various energy harvesting

situations and show the utility of our approach across a wide variety of such scenarios in later sections of this thesis.

1.5. Ultrasonic Through-Substrate Communication

Many structures that need to be monitored e.g. airplane wings / stabilizers, bridge beams etc. are constituted of substrate materials such as metal or composites, which themselves can be used as communication media for signals such as ultrasound. This has motivated the idea of developing a through-substrate sensor network. Tiny sensors, when embedded or affixed on a bridge or an aircraft wing, can communicate with each other using ultrasonic pulses propagating through the structure's solid substrate. This can eliminate the need for out-of-substrate radio or wired links. A prototype modem as described in [8], which deals with the various challenges in such a design, had been developed in our laboratory to demonstrate the feasibility of this approach and we discuss various physical layer implications of such an approach in the current thesis.

Moreover, opportunities exist in harvesting energy from ambient vibrations in several structures such as airplane wings, stabilizers, and bridge beams. Such harvesting can provide energy for sensing and communication of collected sensor information. Self-powered sensors have already been demonstrated [43], [44] which can use the energy from the signal being sensed to power the sensing, computation and non-volatile storage operation. Work is under way on a collaborative project to design a piezoelectric-based transducer that can use vibrations inside a structure to power communications in addition to sensing and buffering needs. The notable fact here is that the same transducer (i.e., a substrate-embedded piezoelectric module) can be used for sensing (ultrasonic fault signatures), communication (ultrasonic link), as well as energy harvesting (ultrasonic vibration harvesting) to power all operations. Such convergence of functionality in a through-substrate approach leads to a cleaner design by removing the need for separate retro-fitted

components for sensing, communication and, energy generation. We will demonstrate the feasibility of such an approach through simulation results in an integrated harvesting and communication framework in the current thesis.

1.6. Spiking Neuron Based Event Pattern Detection

Structural Health Monitoring applications and others of the same family often require identification / classification of spatiotemporal event occurrence patterns [14], [15], [39], [45]–[49] using distributed sensor measurements in order to make higher level inferences based on the same. A key requirement in such applications is that the sensing architecture be flexible enough to be able to identify a variety of event occurrence patterns i.e. if new patterns need to be detected, the architecture should be able to adapt to the new scenarios. Another aspect is the need for generalizability over a range of similar inputs i.e. detection robustness to minor changes in the same event pattern. Such applications also assume some amount of energy-efficient / energy-aware operation because many modern distributed sensing architectures for such aim to create cheap and maintenance-free operation by relying on small sensing devices with limited energy storage but theoretically infinite energy generation capacity (harvesting from environmental sources) albeit at low / erratic rates (harvesting source unpredictability) as discussed in the previous sections. Spiking Neuron based approaches can be shown to have these properties and work with much less energy compared to comparable approaches and are well amenable to discrete-pulse-based networking approaches. In this thesis, we propose such an architecture melding discrete-pulse-based networking and spiking-neuron-based event pattern detection and discuss the research challenges associated with designing such a system. We also evaluate the developed architecture in the context of both synchronous and asynchronous discrete-pulse-based networking protocols

to show how the system can be a feasible and energy-efficient means of spatiotemporal detection especially combined with pulse networking.

1.7. Dissertation Objectives

The core objectives of this dissertation would be to design the following –

1) A scalable discrete-pulse-based communication protocol

2) An energy-aware discrete pulse-based networking protocol, and analyze how performance varies in different energy harvesting / availability scenarios

3) An ultrasonic through-substrate computing architecture, based on discrete-pulse-based communication and powered by vibration energy harvesting, for structural health monitoring applications, for example in airplane wing / stabilizer structures

4) A spiking-neuron-based architecture for energy-efficient spatiotemporal event pattern detection which can easily interface with a discrete-pulse-based networking architecture

1.8. Scope of Dissertation Thesis

In the current thesis, we cover all the objectives of the dissertation as listed in the last sub-section, including published results on the same. In a prior thesis proposal, we had presented then-completed work on the first 3 Dissertation Objectives and laid the foundations for the work on Spiking Neuron based pattern detection architecture (Dissertation Objective 4) and how future work would be used to evaluate this architecture.

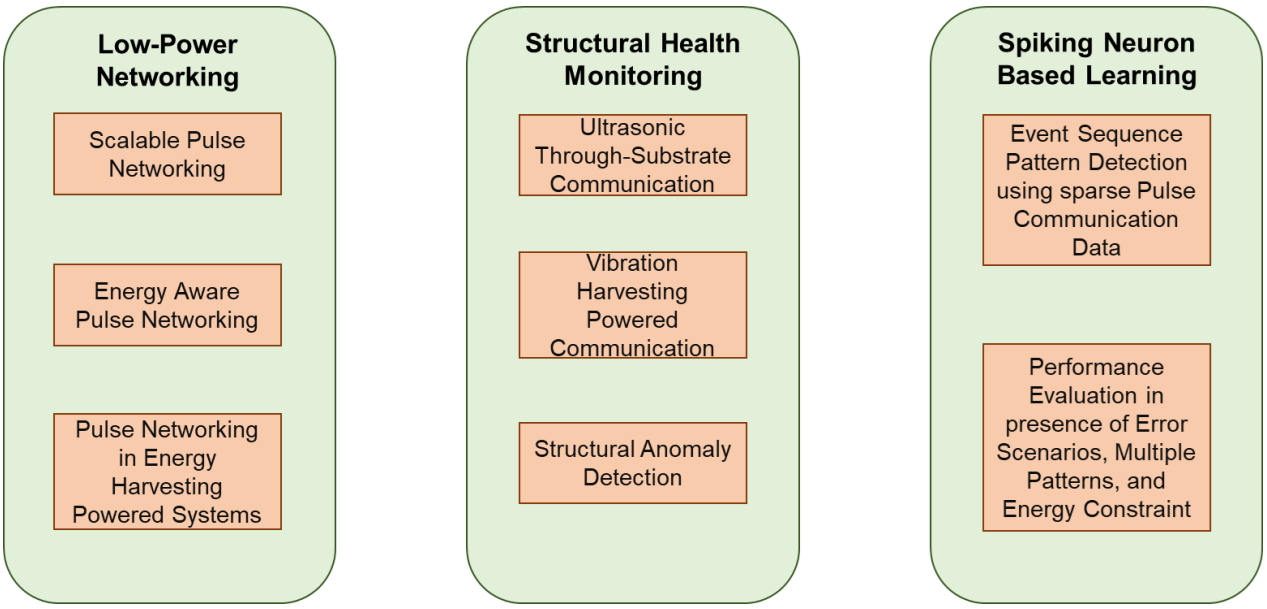


Figure 1.1. Thesis Outline

In the present thesis, we have extended the work from the proposal and completed the Spiking Neuron based detection architecture development and evaluation. In all, we have tried to develop a holistic solution for energy-efficient detection of structural anomalies using a through-substrate network of inexpensive sensors powered by ambient vibration harvesting. In Figure 1.1, we provide an outline of the main thesis objectives and clearly indicate what has been achieved grouping the contributions under various pertinent application areas. In the following chapters, we will start with a survey of the existing state-of-the-art in these areas and then follow up details on our specific contributions.

CHAPTER 2: RELATED WORK

2.1. Packet-Based Networking Alternatives

There are few reported approaches in the literature that address the energy and capacity overheads of packet-based network communication. This is mainly because networking for application niches such as Structural Health Monitoring, that require low volume information transport with relaxed latency requirements have not been explored very far. Research on networked systems with extreme energy constraints such as those powered by small piezo-electric energy harvesting devices is not very mature either. We aim to bridge the gap in research in this domain through the present thesis work.

Strategies like aggregation of short payloads [50] and binary sensing models [51] have been proposed for reducing the networking energy burden, but the inherent packet overhead limitations remain. In [52], it is shown that in single-hop networks, the worst-case performance of pulse-based communication is better than packet-based ones, albeit with a worse delay performance. Inspired by this result, the authors in [35], [53], [54] developed the concept of multi-hop pulse switching as an alternative to energy-inefficient packet-based communication. This concept is very well-applicable when the data to be transmitted is binary in nature and the tolerable delay is high. However, the multi-hop MAC-Routing protocol framework for event routing, as mentioned in [35], [42], has some limitations in the context of applications using energy-harvesting or extreme energy constraint. The latter is generally envisioned for various maintenance-free Structural Health Monitoring applications powered by ambient energy harvesting. The first shortcoming of the baseline pulse switching is that it implicitly assumes constant energy availability, which is not valid when the available energy is variable depending on the harvesting

conditions. This assumption can be detrimental in terms of energy management and can lead to unwanted power losses when the energy availability changes drastically. This is because the network energy usage is not matched with changing energy availability. Such erratic changes in energy generation are very probable particularly in harvesting-powered systems. In the current thesis, we will discuss energy-aware pulse routing syntaxes that have been developed to address this limitation. The second limitation is that the synchronized time frame structure proposed for the protocol in [35], [42] does not scale well for larger networks. We develop a new time-frame structure to ameliorate this. Yet another factor is that the architecture in [35], [42], is designed using Ultra Wide Band–Impulse Radio as the physical layer. In the present thesis, we will show how this architecture can be adapted specifically for an ultrasound-based through-substrate physical layer which presents various challenges of its own.

2.2. Energy Aware Networking

Various other approaches have been proposed in the literature [55]–[57] with regards to energy conservation, including data-driven approaches such as reducing data through in-network aggregation and compression [50] and interest driven data acquisition [58]. Another approach is to keep local energy costs low at the sensing nodes by adopting efficient sleep schedules and topology control (only a limited number of nodes are active, that is not in sleep), based on data and energy conservation requirements [55]. There have also been many approaches on energy-aware MAC and routing protocols in packet-based sensor networks operating under various forms of energy constraints including from harvested energy. A general direction is to adapt the sensing duty cycle based on a harvesting model [59], [60] or by tracking the battery energy level and its depletion pattern [57]. Other approaches [58] include using an energy-efficient networking approach such as directed diffusion while building energy awareness features on top to prevent

energy leakage when the network is energy-constrained. Energy-harvesting aware protocols like [61] have also been discussed in the literature, which consider the harvesting rate and current consumption characteristics combined into a node's current energy state and utilize that to design the optimal routing approach such that the system can survive on the harvested energy. While these approaches provide generalized energy-saving concepts, they are designed for traditional packet-based operation and thus cannot directly be applied to the pulse-based approach considered here.

In the current thesis, we discuss the development of energy-aware mechanisms using similar principles to the literature, but specifically for pulse-switching protocols. Like the literature, these would attempt to achieve energy neutrality, but for a pulse switching network in a vibration based harvesting environment, and thus deal with the unique challenges presented there. In the thesis, we also present a fresh look at the pulse networking architecture presented in [35], [42], [54] toward solving the inherent disadvantage of very high latency when the network cell count is scaled. A scalable pulse architecture, as will be discussed in later sections of this thesis, also provides better applicability in terms of larger information content routing, lower latency, fair access latency to the network sink from across the network without increasing the energy cost.

Deployment strategies and their effect on energy-efficiency and reliable data acquisition in Wireless Sensor Networks (WSNs) has also been covered extensively in the literature [62]–[64]. For example, non-uniform deployment of sensors has been proposed to deal with disproportionate energy consumption at the sensor nodes close to the sink thus increasing lifetime of network operation. Also, strategies like separation of relaying (forwarding) and sensing functionalities has been suggested to increase energy efficiency. It is to be noted that such strategies can be used in the Pulse Networking architecture as well because they mainly deal with network organization and set-up which is independent of the actual networking scheme in this case. Use of mobile nodes for

data acquisition from WSN sensor nodes has also been put forward in the literature [65]. Since, the autonomous energy-rich sinks can travel to the source nodes and collect information, there is no need for multi-hop transmission and the sensing node energy is conserved (due to absence of forwarding function). Authors in [66], [67] discuss various challenges in a mobile sink scenario which includes various strategies for synchronizing sink acquisition schedules with node sensing schedules. It is to be noted that though mobile sink-based approaches have been shown to be advantageous in terms of energy conservation of the sensing nodes, they involve network maintenance in terms of using a source-coordinated sink mobility schedule. In the SHM monitoring applications, like the one we have mentioned before, the intent is to keep the data acquisition maintenance-free and the multi-hop approach works well in such scenarios due to absence of any extraneous control (for sinks etc.).

2.3. Structural Health Monitoring

Structural Health Monitoring has been an area of growing interest in recent years. This is because of the vast array of infrastructure in need of maintenance and the fact that maintenance and repairs represent a staggering fraction of infrastructure costs. For instance, maintenance and repairs represent a quarter of airplane operating costs and the U.S. spends more than $200 billion dollars on maintenance of plant, equipment and facilities. The motivation in SHM is in reducing such costs by replacing scheduled maintenance with as-needed maintenance [68]–[70]. Other goals include anticipating structural lifetime and rate of degradation. Wireless sensor network-based approaches [7], [40], [71] provide a seamless way to deploy such monitoring solutions. Energy-harvesting powered wireless networks can be even more useful because the network of sensors can function almost maintenance-free just based on ambient energy harvesting [72]–[74]. Various works have been proposed in the literature in this regard [59], [60], [72], [75]–[78]. However,

almost all these approaches rely on traditional pulse-based networking approaches which have been shown to be energy-expensive in such low information content scenarios. The differentiation proposed in this thesis is mainly in terms of adapting the pulse networking architecture to this domain. We will show that by incorporating pulse networking along with energy-awareness syntaxes as well as a scalable framing architecture, we can get better and more resilient performance in the SHM application context. We will also propose further low-energy detection solutions which can be embedded within the network nodes themselves and how a holistic low-power networking and detection architecture can be developed based on the same.

2.4. Through-Substrate Ultrasonic Communication

There have been other contemporary approaches towards the use of ultrasonic communication for sensor networking. The authors in [79] have outlined the advantages of using ultrasonic communication for human intra-body applications as opposed to radio communication. The latter is inefficient around human bodies because of water being a dominant constituent. The authors have discussed various adaptations to the MAC-layer protocol such as rate adaptation and stochastic channel access techniques. However, the advantages of these adaptations hold true mostly in the intra-body context (where radio propagation has disadvantages compared to ultrasonic), whereas the advantages due to the Pulse-based networking approach can be utilized even when the underlying technology is based on radio such as UWB [54]. Our choice of ultrasound as the medium for communication is mainly motivated by the advantages in integrating the sensing, communication and energy harvesting aspects in structural health monitoring scenarios which we detail further in the upcoming chapters of this thesis.

2.5. Spiking Neuron Based Event Pattern Detection

The work in [80] was one of the first among many [81]–[84] to describe a biologically plausible supervised synaptic learning rule that Linear Integrate and Fire (LIF) neurons can use to efficiently learn and read out spike-timing based neural codes. This is useful because neural codes that embed information in the spatiotemporal structure of spike patterns have been known to be computationally a very efficient means of encoding information. Thus, spiking neurons once equipped with the Tempotron learning rule should be able to leverage this. The Tempotron work [80] mainly attempts to show that a gradient descent-inspired learning rule, based on an error which is represented as the voltage distance between expected membrane potential for spike / non-spike and the maximum actual sub threshold membrane voltage, can be used with a very high capacity to decode information encoded in spatiotemporal spike patterns. This work [80] is primarily concerned with demonstrating that Tempotron can successfully distinguish between patterns from two different classes within some bounds on the number of total patterns given the number of synaptic inputs to the neuron. In the current thesis, we develop a framework for applying the Tempotron learning rule to a spiking neuron for detecting selected event occurrence patterns vs others. We also lay out the scenario in terms of a specific application (event pattern monitoring) and the spike patterns involved therein due to use of spike-based networking protocols. The premise here is that event sequences patterns' detection can be valuable for many application-specific inferences like structural anomaly detection, tracking progression of a structural crack along the structural substrate and so on. We provide the architectural details on how to create a Spiking Neuron-based architecture for such applications and why it would be useful in terms of energy-efficiency and deployment ease. We also articulate the research challenges and then provide a detailed characterization of the effect of different system and learning parameters on the

system performance. These include choice of training patterns, choice of pulse-based protocols (synchronous and asynchronous) as well as the system's robustness to both spike jitter and a broad range of inter-event (spike) intervals, a presentation which has not been made in prior work.

Other works [81]–[83] have also covered the topic of spatiotemporal pattern recognition using Tempotron-like learning rules. For example, in [81] the authors aim to create a Spiking neuron-based learning framework that is able to distinguish among multiple classes of input patterns where information is embedded in the precise timing of spikes relative to each other to generate precisely timed output spikes. The authors suggest two specific learning rules and the main improvement over the Tempotron rule is the fact that the output spike jitter can be lower than the input spike jitter thus making the system more robust to noise. It is to be noted here that our current proposal of Spiking Neuron-based detection can also be reproduced in a similar setting replacing the Tempotron-based learning with the Chronotron rule as mentioned in [81]. However, we want to show the baseline operation of the application system in the context of our functional specifications and so consider only the Tempotron learning scenario. Future work will cover the implications of using different learning mechanisms.

Yet another approach to the spatiotemporal pattern classification problem using Spiking neurons is provided in [82]. This technique, called ReSuMe, applies a similar learning approach to Tempotron [80], but using a Widrow-Hoff rule to decide the weights' updates and an error based on the distance of actual output spike trains produced from the desired spike trains. It is to be noted here that approaches like ReSuMe can produce different spike trains for different classes of input spike train patterns such that the precise timing of the output spikes can be read out to make conclusions about the pattern class. This is more generalized than the Tempotron approach where the output pattern can be either a spike or no spike indicating only two classes. However, the latter

approach is suitable for the purposes of our application i.e. binary event classification and hence we chose to leverage the simpler Tempotron approach. In future work, we have plans to integrate other works like ReSuMe and report the efficacy of the system. Our main contribution in relation to the other established approaches [80]–[84] to precise spike time based pattern classification lies in the fact that we adapt it with energy-efficient discrete-pulse-based networking protocols and consider the detection scenarios for different classes of unknown negative patterns and noise scenarios that would be present in a practical application scenario.

2.6. Summary

In the following chapters of this thesis, we will develop scalability and energy-awareness within the inherent energy-efficient discrete-pulse-based networking paradigm. In addition, we will present an architecture for through-substrate ultrasonic communication for SHM applications using pulse networking and evaluate the system performance using realistic simulations for an airplane stabilizer structure. We will also develop a low-power Spiking Neuron based architecture for event pattern detection which we aim to incorporate in our pulse communication-enabled SHM architecture. We will mention the research challenges in this pursuit and how we tackle these. We will also include results from performance evaluation experiments for this low-cost detection architecture establishing its feasibility.

CHAPTER 3: SYNCHRONOUS PULSE NETWORKING

In this chapter, we will cover briefly the fundamentals of the Synchronous Pulse Networking approach as has been published in [35], [42], [54]. This is necessary because in subsequent chapters we will refer to various augmentations to the baseline protocol architecture discussed here in response to application needs like network scalability, energy-harvested operation, and through-substrate communication. The basic premise is that a discrete-pulse-based networking approach is better suited in terms of energy efficiency for applications with low information and latency demands. We will present here the protocol syntaxes and features which lend the discrete-pulse-based architecture its energy economy in target applications like Structural Health Monitoring.

3.1. Pulse Abstraction

In the pulse networking domain, the key mechanism is to use individual pulses or the absence of such to indicate binary event information, that is presence or absence of an event. In low-information-density applications requiring ultra-low energy operation, use of the pulse abstraction can significantly lower the energy overhead of transmissions compared to traditional packet-based mechanisms. Often, just the binary event information from distributed sensors when collated at a central Base Station is valuable for making high-level inferences about the system being monitored. However, beyond the binary event information, at least two more pieces of information are required for such systems, namely the event location information and the next-hop information to facilitate multi-hop routing. Event location is needed to give proper context to the event occurrence info gathered while multi-hop routing is generally preferred as it enables the use of small, inexpensive individual sensing devices. In order to preserve the energy advantages of pulse abstraction over packet-based networking, at least these two essential pieces of information

need to be encoded without incurring further transmission costs. In further sections, we will cover how this is achieved.

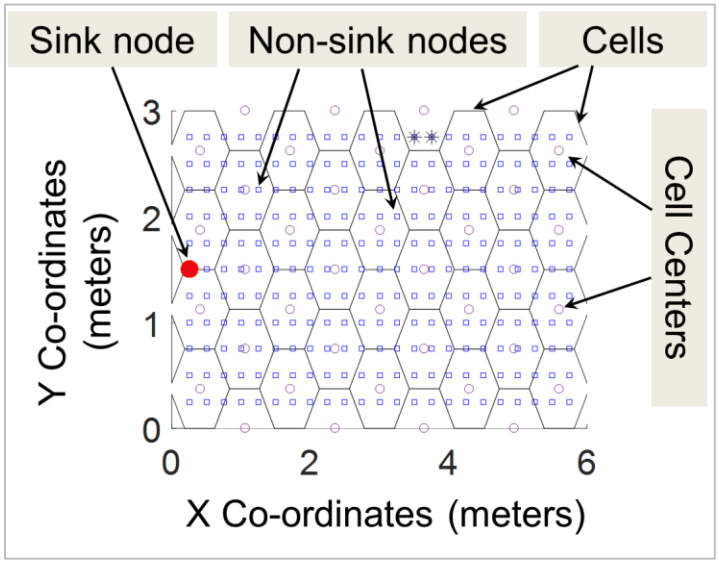


Figure 3.1. Cellular Network Model on a Rectangular Plate Structure

3.2. Network Model – Cellular Abstraction

Before delving into the pulse networking semantics, it is important to understand the general network model envisioned for such applications. The network of sensors for collecting, for instance structural event information, will consist of sensor nodes distributed uniformly across the structure being monitored and a strategically placed Base Station (sink) where information from sensors across the structure are collated. For simplicity, in Figure 3.1, we have shown an illustrative rectangular plate structure with sensor nodes embedded throughout the structural area. Data collection nodes are referred to as non-sink nodes while the data integration node is called the sink node (placed on the left edge of the plate). The sensor nodes are equipped necessary event sensing as well as forwarding functions (in terms of a pulse networking interface). Each node also has some pre-programmed localization information. Localization is accomplished with the

resolution of pre-defined sensor-cells like the hexagonal cells as shown in Figure 3.1. Each cell represents an event area, with a unique Cell-ID. Since spatial localization resolution is at the cell level, shrinking the sensor cell size can increase the resolution. This feature gives us the flexibility to tune our localization resolution based on application needs. Multiple sensor devices in each cell also enable data reporting redundancy which is valuable when the individual sensors might not be robust enough. Each sensor node belongs to one of these event areas (cells) and is pre-programmed with the Cell-ID of its own cell and those of its geographical neighbors. Although the cells in Figure 3.1 are shown to be hexagonal, there are no specific architectural requirements in terms of their symmetry, shape, and size. Generally, a hexagonal cell structure is chosen for maximal cell packing with minimum number of cells. Due to the cellular abstraction, the sensors are not individually addressed, and therefore no per-sensor addressing is necessary at the MAC or routing layers.

3.3. Cellular Pulse Protocol Frame Structure

A summary of the Baseline Pulse Switching Protocol is provided in this section. The objective is to highlight its major features, which make it an interesting choice for the application domain discussed earlier. The details of the protocol were originally presented in [35], [42], [54].

3.3.1. Pulse as a Protocol Data Unit

Upon detecting an event at a network node, a discrete pulse is sent to the sink using a multi-hop pulse routing process. Localization information about the received event is inferred by the sink from the time-of-arrival of the pulse with respect to a specified MAC-Routing frame structure as presented below.

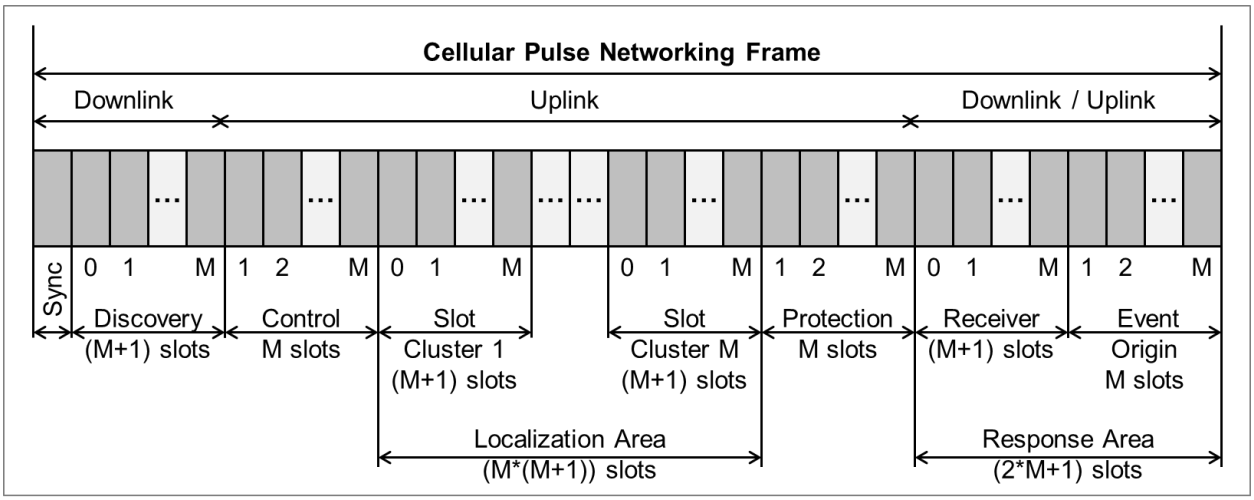


Figure 3.2. MAC-Routing Frame for Pulse Switching [85]

3.3.2. Joint MAC-Routing Frames

As reported in [35], [42], [54], a network-wide synchronized frame structure, as shown in Figure 3.2, is followed by each network node. The frame is controlled and enforced by the sink node. The frame is divided into many slots as shown in gray in Figure 3.2. Each slot in a frame is used for sending a single pulse. The width of a slot is chosen based on the minimum pulse separation delay for the physical link under consideration. For example, this can be on the order of nanoseconds (ns) [53] for UWB pulse radio systems while in the order of milliseconds (ms) [8] for ultrasonic pulse communication systems. This slot duration should also be large enough to accommodate any cumulative clock-drift during a frame, as well as the propagation delay for the physical layer communication medium used. It is to be noted that the propagation delay effects might lead to sync pulses reaching different nodes at slightly shifted time instants, but if the pulses arrive within the designated slot, the system operates correctly. As shown in Figure 3.2, each frame includes various downlink and uplink areas. Operational details for these are presented below.

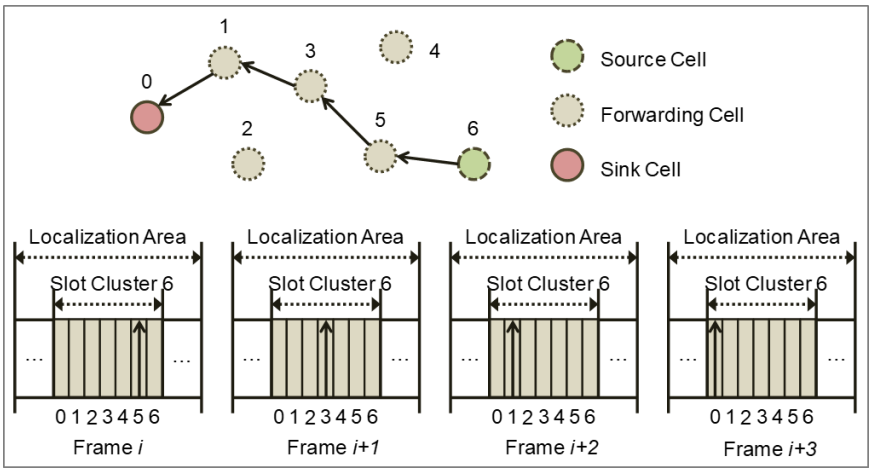


Figure 3.3. Pulse Forwarding in the Localization Area [8]

A. Frame Synchronization

Every frame starts with a Sync Area, which is allocated to the sink for network synchronization. During the Sync Area, the sink sends a pre-defined pulse pattern, which other nodes can detect and use to identify the start of a frame. The sink, having unlimited energy supply, can transmit the sync pulses with high enough power such that they reach all nodes in the network. Generally, this is a single pulse transmitted in the sync slot.

B. Pulse Forwarding

The uplink Localization Area of the frame is the key to pulse forwarding. In a network with M sensor-cells (excluding the sink cell), this area contains M slot-clusters, each cluster containing (M+1) individual slots. Each slot-cluster corresponds to a specific Cell-ID representing an event's cell of origin. In each slot-cluster, individual slots correspond to specific Cell-IDs, which represent the next-hop cell for a transmitted pulse within the slot cluster. Figure 3.3 shows an example scenario to demonstrate pulse forwarding using the Localization Area. It is important to note that the slot cluster for a pulse remains unchanged during the entire forwarding from the origin sensor

to the sink node, with the specific transmission slot dependent on the next hop as per the routing decisions. Hence, when the pulse arrives at the sink, its slot cluster indicates the cell of origin. Pulse forwarding decisions (i.e. the next hop) are made based on a sensor's routing table, which maintains a sorted list of next-hop cells based on the hop-counts of the resulting routes based on a discovery process discussed hereafter.

There is also a Control Area prior to the Localization Area of the frame, which is used for announcing impending transmissions within the Localization Area. All non-transmitting nodes can use this information to decide sleep and wake-up schedules for additional energy savings.

C. Route Discovery and Response Mechanism

These are optional components of the frame structure. The first, that is Route Discovery, is a continuous background process that creates and maintains the routing table in each sensor in terms of the next-hop Cell-IDs. The Response mechanism, on the other hand, enables a single-pulse acknowledgement for enhancing one-hop transmission reliability in presence of pulse errors. It is to be noted that the protocol works even without a Route Discovery scheme if there is a static routing table configured in each node before networking operation begins, but dynamic route discovery can help the network better adapt to changing conditions of routes e.g. in an energy dynamic environment. Response mechanism provides communication reliability but might be skipped in low Pulse Loss Rate environments to save energy on redundant pulse transmissions. Details of these two components are covered in [42], [54].

3.3.3. Protocol Features

The baseline Synchronous Pulse Switching protocol also provides other useful features such as Route Diversity (RD), Spatial Compression and Pulse Merging. RD is a protocol

parameter, which indicates the number of separate paths along which the pulse information should be propagated towards the sink providing redundancy and better reliability in case of cellular fault scenarios. Spatial Compression is a feature used to prevent identical transmissions from multiple nodes in the same cell. Yet another feature is Pulse Merging, which provides inherent in-network aggregation for events originated from same cells. It is to be noted that only nodes in the same cell can transmit simultaneously and, in such cases, there is no interference, rather the pulses are reinforced on superposition. Details of these are discussed in [42], [54].

3.4. Need for Scalability and Energy Awareness Improvements

It is to be noted that with increase in number of cells, the delivery delay for the Baseline Synchronous Cellular Pulse Networking scheme described above increases quadratically with the number of cells due to the Localization area size of the frame. Hence, the protocol is mostly targeted for networks with small number of cells for a reasonable localization resolution. When covering large areas, the protocol still works, though either with a reduction in spatial resolution (larger cells but lesser in number) or further relaxed delivery-delay specifications. However, when spatial resolution demand is high and further latency cannot be afforded, scalability suffers. This necessitates a fresh look at the fundamental time frame structure enabling pulse networking for possible changes to improve scalability.

The baseline Synchronous Pulse Protocol architecture also implicitly assumes that the energy availability in the network is constant. In erratic energy-harvesting-powered systems, such an assumption is invalid and will lead to mismanagement of energy resources resulting in nodes running out of power prematurely. Hence, there is a need to incorporate energy-aware mechanisms within the baseline architecture to better equip the systems for energy-harvesting-powered operation.

3.5. Summary

In this chapter, we covered the foundations of Cellular Synchronous Pulse Networking and how discrete pulses' positions with respect to a synchronized time structure can be used to convey event information. We detailed the networking protocol components to show how medium access is managed and multi-hop routing achieved. We also consider the limitations of using this framework as-is in a dynamic and constrained energy environment or in networks with large cell counts.

In the next chapter, we will develop a revised Synchronized Pulse Networking Frame Structure which can scale for larger networks without sacrificing the energy-efficiency benefits of Baseline Pulse Networking or realizing further latency concessions. In subsequent chapters of this thesis, we will also develop various energy-awareness syntaxes within the Baseline Pulse Networking framework which can provide more judicious management of dynamic energy resources and thus better performance in energy-harvesting-powered systems.

CHAPTER 4: DEVELOPMENT OF A SCALABLE PULSE FRAME STRUCTURE

Building on the baseline synchronous pulse networking detailed in Chapter 3, in this chapter, we will develop a Scalable Cellular Pulse Networking (SCPN) architecture. We will design a novel time Frame Structure which can enable graceful scaling of Pulse Networking latency with an increase in the number of network cells. As mentioned earlier, increase in number of network cells can be mandated by need for a higher resolution of detection/monitoring or increase in the network area being covered. The basic premise is still to use discrete pulses to indicate event occurrence, but the pulse's position in time will be with respect to a new frame structure which is smaller compared to the baseline version, allowing better latency performance without energy-efficiency concessions.

4.1. Scalable Cellular Pulse Networking (SCPN) Frame Structure

The scalable time frame structure as shown in Figure 4.1 is designed to be comprised of the following components:

A. *Discovery Sub Frame (DSF):* This is a downlink portion of the frame which has similar functions to the corresponding one in the baseline synchronous CPN frame structure as discussed in Chapter 3 of this document. This sub-frame is involved in discovering neighbor nodes during the initial stages of network set-up and thus aids in formation of routing tables to be used during multi-hop event transmission from source to sink. The Discovery sub-frame uses up to $(M+1)$ slots where M is the number of non-sink cells in the network.

B. *Next Hop Sub Frame (NHSF):* This is a part of the frame where a pulse in a designated slot indicates to the receiver that it should be awake during the Event Origin Sub Frame (discussed hereafter) part of the frame. The NHSF is used to notify the appropriate next hops about

information coming their way. In the NHSF, each cell *i* is allocated the corresponding slot *i*. Nodes with cell-ID *i* are only awake to receive in their own cell-slot i.e. slot *i* in the NHSF. Nodes are asleep in other slots unless they have pending information to forward. If a node has information to forward, it checks its neighbor table and selects the appropriate next hop cell-ID based on energy-aware routing strategies (detailed further in Chapter 5). If a node *i* decides to transmit to its neighboring cell *j*, it sends this transmission in the slot where *j* is receiving i.e. slot *j*. A pulse at slot *s* in the NHSF tells a receiver with cell-id *s* that it should be awake during the whole period of the Event-Origin Sub-Frame (EOSF) to receive event data coming its way. The NHSF uses up to *(M+1)* slots.

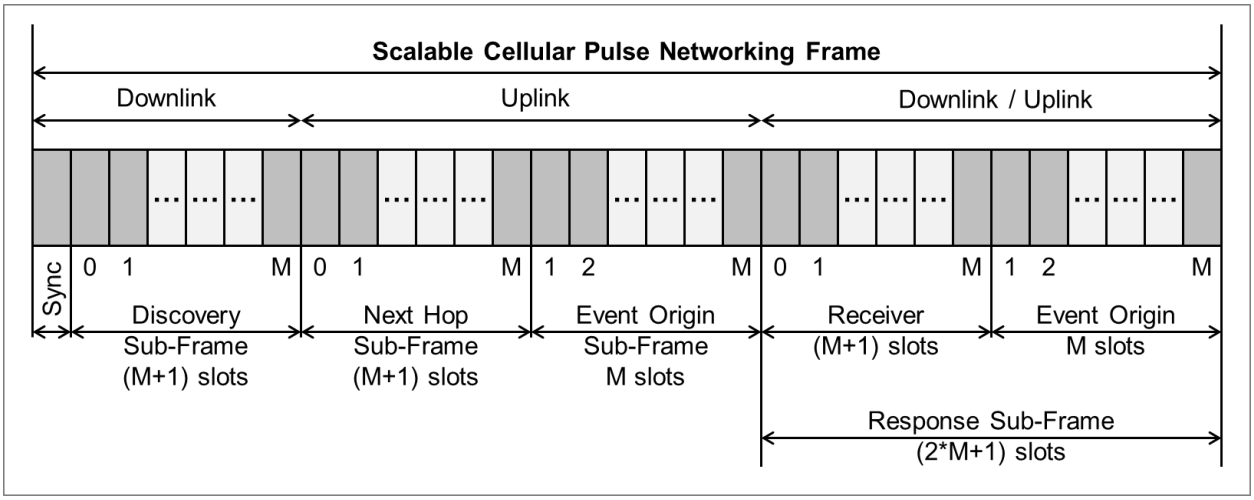


Figure 4.1. Joint MAC-Routing frame for Scalable Cellular Pulse Networking [85]

C. *Event Origin Sub Frame (EOSF):* The Event Origin Sub-Frame is used to indicate the event origin location for the information being forwarded. A pulse received at slot *s* in EOSF indicates an event originated at a node with cell-id *s*. It is to be noted here that the only nodes receiving in the EOSF will be the ones that have been notified about events coming their way during NHSF. So, all nodes indicated as next hops in NHSF will be awake and receiving in all slots of EOSF while all other non-transmitting nodes will be sleeping. All transmitting nodes will

be awake and transmit a pulse in the respective event origin cell-slots. The EOSF uses up to *M* slots where *M* is the number of non-sink cells in the network.

Response Sub Frame (RSF): The Response Sub-Frame is identical to the Response Area in the baseline Pulse Networking Scheme and an optional component to guarantee reliable detection of pulse errors / false positives. The RSF uses up to *(2*M + 1)* slots.

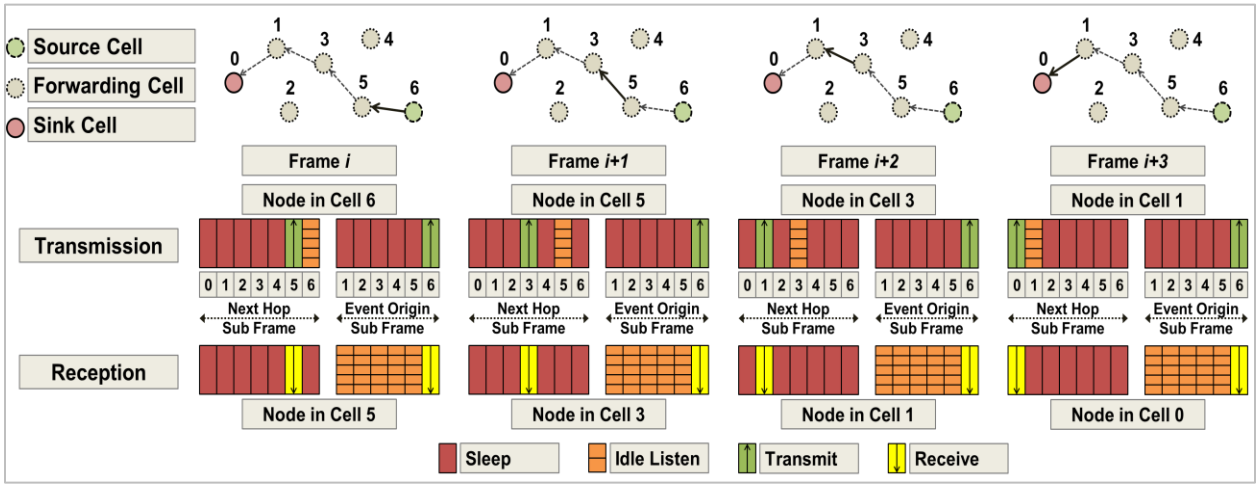


Figure 4.2. Demonstration of multi-hop event forwarding in SCPN

Figure 4.2 demonstrates a multi-hop forwarding operation using the SCPN frame structure. An event originating in a node with cell-ID 6 is forwarded through nodes in cells 5, 3, 1 up to the sink (cell-ID 0). In Frame 1, the node is cell-ID 6 transmits a pulse in the NHSF slot 5. This indicates to the node in cell-ID 5 that an event is to be sent to them. As a result, nodes in cell-ID 5 keep awake and listen during all slots in the EOSF of the same frame. Since the event is originated at cell-ID 6, the node in cell-ID 6 transmits a pulse in slot 6 of the EOSF which the node in cell-ID 5 receives. In the next frame, the node in cell-ID 5 forwards the event it received in the last frame by first sending a pulse in slot 3 of NHSF to notify its neighbors in cell-ID 3. In the EOSF of this frame, the node in cell-ID 5, transmits in slot 6 (indicating the origin cell of the event) which

is received by the node in cell-ID 3 which is awake in EOSF because of the NHSF transmission of node in cell-ID 5. Hence, the event originated in node with cell-ID 6 has now been propagated up to the node in cell-ID 3. Using a similar procedure, the node in cell-ID forwards the event up to 1 and then on to the sink (cell-ID 0) as shown in Figure 4.2.

It is to be noted that the NHSF and EOSF together combine the functionality of the Localization Area in the baseline Pulse Networking protocol. The important distinction is that while the Localization Area uses up to $O(M^2)$ pulse slots, the NHSF and EOSF in the new frame structure uses only $(M + 1 + M) = (2*M + 1)$ i.e. $O(M)$ pulse slots and is thus much better scalable with increase in the number of network cells. This saving in terms of the frame size can be utilized in multiple ways e.g. reducing the overall delivery delay or toward increasing the amount of information encoded in each frame while keeping the delivery delay at par with the baseline approach.

The trade-off in the current approach is the lack of distinction between event origin data heard from different nodes in the EOSF. During the EOSF, there is a possibility of nodes, which are awake, overhearing pulses which are not meant for them and thereafter forwarding these pulses to their neighbors. As a result, there might be extra energy used or leakage for communicating redundant information (because the same information will also be propagated through the node where it was originally intended). In the current study, we have not proposed any solution to this because as the simulations demonstrate the scalable frame structure gives better energy performance compared to the baseline Pulse protocol despite the overhearing leakage if any. Moreover, overhearing is not always a problem. It can only become a source of leakage when data originated from different sources are being propagated in the same frame across different routes and the nodes on these routes are neighbors such that one can overhear pulses meant for the other.

Such confluence of factors is very rarely the case. For example, in many cases the events being propagated across different routes are originated at the same node. In such cases, there will effectively be a merging of the pulse in the same slot and overhearing is not an issue. Also, if the routes in consideration are spread apart, there would be no overhearing. Another important thing to note is that the root cause of overhearing is the fact that event pulses meant for multiple destinations are being transmitted in the same sub frame. This problem can be ameliorated by staggering the transmissions from neighboring nodes if there are multiple nodes forwarding the same data.

4.2. Simulation Setting and Performance Results

4.2.1. Network, simulation and event model

An event-driven C++ simulator implementing MAC-Routing framing for both CPN and SCPN was developed. The baseline simulated network consists of TUPN sensor nodes evenly distributed on a rectangular plate structure with the sink node placed at the center left corner as indicated in Figure 4.3. The nodes are grouped into regular hexagonal cells with an average of 3 nodes per cell. The spacing between individual nodes is 0.25 m and the transmission range is kept at 0.75 m. Height of each baseline hexagonal cell is 0.5 m. We used different dimensions of the plate to vary the network size and hence the number of cells in the network to compare performance of SCPN vs CPN as the network cell count is scaled. The effect would be similar if the cell count were increased by reducing the cell size. The latter is also often the case when higher localization granularity is required. As shown in Figure 4.3, we used 6 different plate sizes of dimensions *4m X 3m, 6m X 3m, 8m X 3m, 10m X 3m, 12m X 3m* with the respective cell counts as 32, 41, 59, 68, 86 and 113 respectively. The plate width was kept similar, to have comparable

routing across the network despite the scaling when a common cell is chosen as source among all the networks.

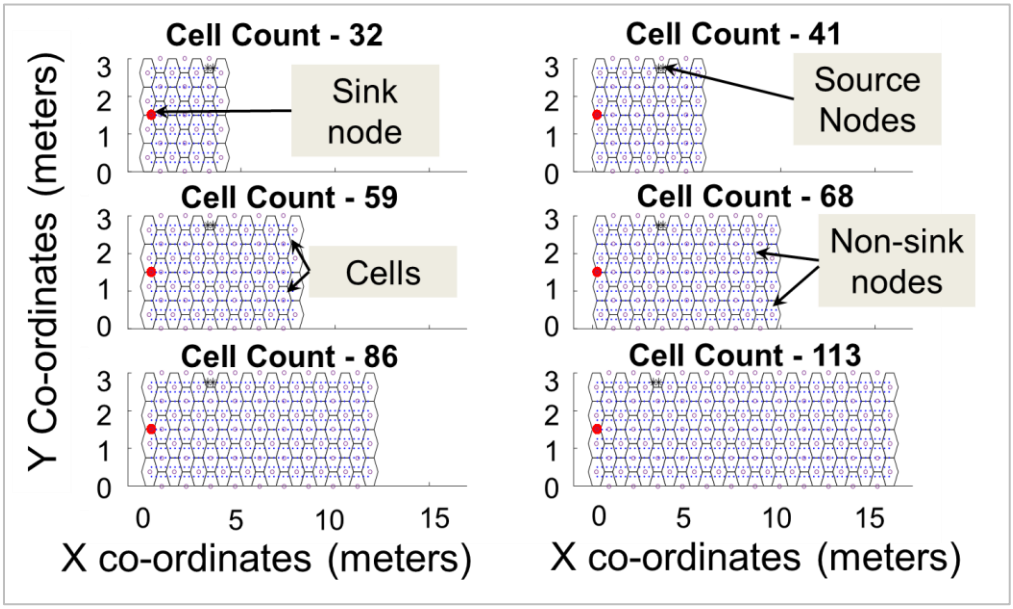


Figure 4.3. Simulated Network Topologies

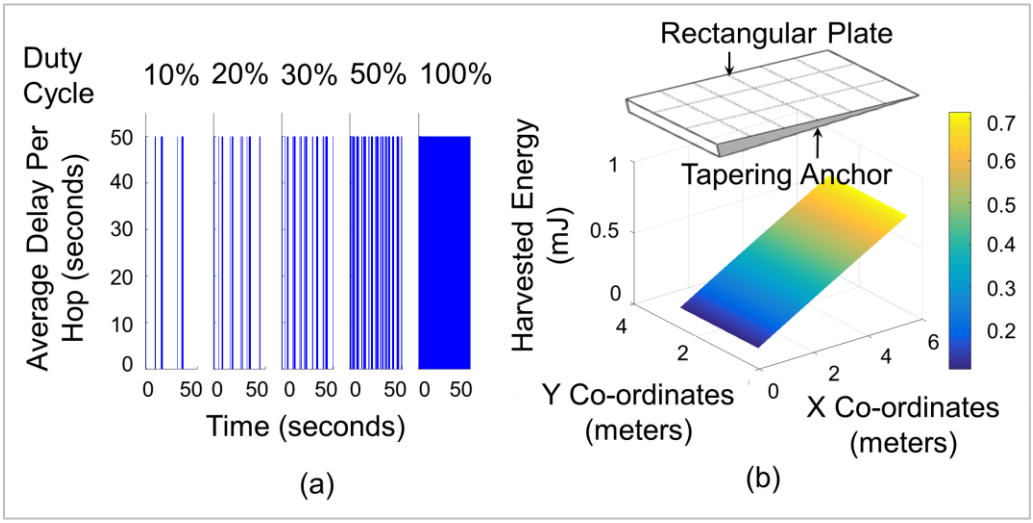


Figure 4.4. Spatiotemporal variation in harvested energy

An energy model is used where the energy for transmitting a single pulse is 100 μJ while the idle listening and sleep energy budgets are 1 mW and 1 μW respectively. Energy availability

at the nodes was designed to have both spatial and temporally variable components. Temporal variation was incorporated with the use of a duty cycle (DC) parameter which controls the proportion of time when harvesting power is available. Harvesting power availability is modeled as a stochastic process with energy availability (charging ON time) and non-availability (charging OFF time) intervals being exponentially distributed. The means of these distributions are chosen based on the DC selected. With this configuration, the harvesting power availability for different DCs is demonstrated in Figure 4.4 (a) with higher DC involving denser power availability leading to more harvested energy. Spatial distribution of energy is incorporated using a spatial distribution function which is linear in the current paper as shown in Figure 4.4 (b). Thus, the nodes near the left edge have less energy harvested compared to the nodes near the right edge assuming the anchor of the plate is at the left edge and tapering away toward the other edge as shown in the inset in Figure 4.4 (b). This is representative of the energy available using vibration harvesting on the plate which is more intense away from the anchor than near. It is to be noted that all nodes are charging synchronously in the simulated model because energy availability events are synchronous across the plate area. Events were generated at different source nodes across the rectangular plate and the corresponding average performance characteristics were noted. Events were assumed to be far apart from each other so that they can be considered as independent transmissions. For event transmission from any source node, 50 similar experiments were performed, and their average noted to take care of the stochasticity in the charging model introduced as part of the exponentially distributed Duty Cycle formulation.

4.2.2. Performance Results

A. Delivery Delay and Event throughput

To demonstrate the advantages of the SCPN design as compared to baseline CPN, we evaluated the network performance across different network sizes in terms of event delivery latency and throughput. In the initial experiments, we considered single event generation from cells across the plate to the sink and the highest level of energy availability i.e. no energy constraint. Further effects of energy constraint have been discussed in subsection C hereafter.

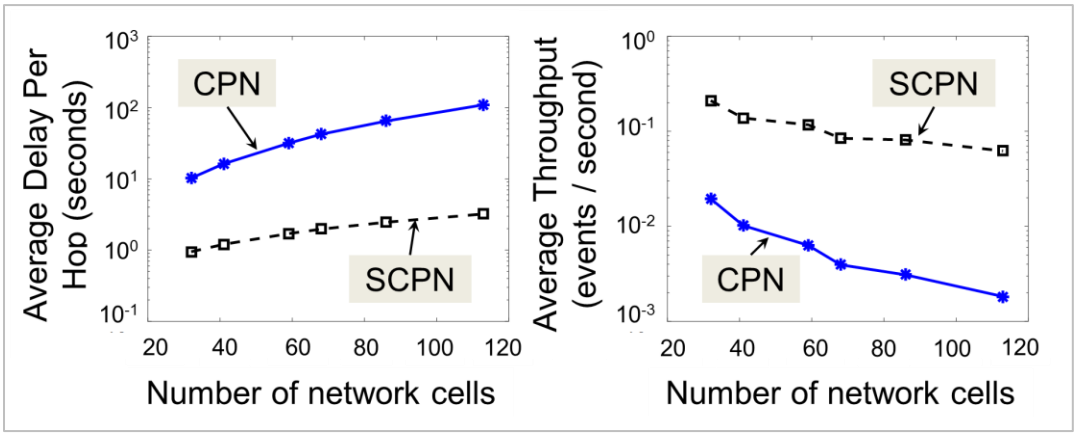


Figure 4.5. Network Cell Count vs Average Per Hop Delivery Delay / Event Throughput

As shown in Figure 4.5, network cell count has a more marked bearing on the delay / throughput parameters in the case of CPN as opposed to SCPN. It is to be noted that the y-axis of the delay / throughput plots is exponentially scaled which is intentional to demonstrate how quickly the average delay per hop and average event throughput scale in CPN when the cell count is increased. This is opposed to SCPN where the event delay increases much more gracefully i.e. slope of latency increase or throughput decrease is lower when the cell count is increased from 32 to 113. It is to be noted that due to the use of a linearized frame structure, SCPN offers per hop latencies in the order of 1-5 seconds while CPN for comparison has per hop delays which are on

the order of 10-100 seconds. Similarly, for the average throughout, SCPN offers much better values of 0.1-1 events/second for the used network topologies as opposed to CPN which offers only 0.1 or lower. Both results are a consequence of the fact that latency and throughput are functions of the frame size and CPN aims to gracefully scale the same. The improved performance ranges open up the feasibility of more time-sensitive applications as well as ones where a high throughput is required for pulse-based networking.

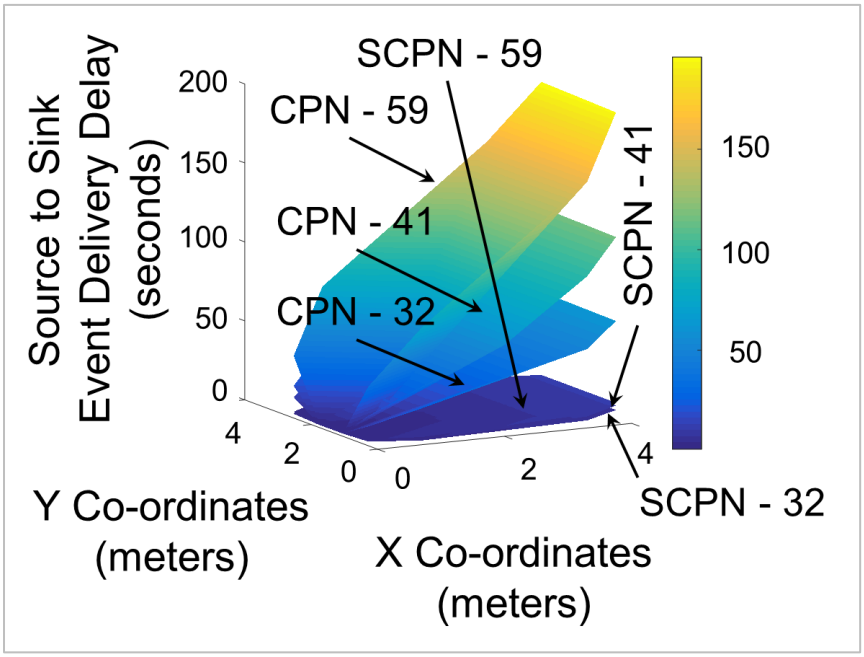


Figure 4.6. Heat maps for source-sink event delivery latency

B. Source to Sink Event Delivery Latency

For a better visualization of the latency advantages provided by SCPN, we have charted the source to sink delay (complete routing delay) for single events generated at different nodes across the plate in Figure 4.6. As seen from the heat maps provided in Figure 4.6, events that are located farther away from the sink, have much higher event delivery latency from source to sink compared to those located nearer to the sink. Such disparity becomes even higher when the

network cell count is increased as the network size increases. Thus, the distribution of event delivery delays is very non-uniform in the case of CPN and nodes farther away from the sink are implicitly disadvantaged. In the case of SCPN though, the source to sink delivery delays scale much more reasonably as the distance from the sink changes as well as due to network cell count changes. This can lead to a fairer event reporting schedule across the network.

C. Effect of energy constraint on event delivery latency

To evaluate the effect of energy constraint on the network performance, we considered two different network sizes - *4x3* with cell count 32 and *6x3* with cell count 41. As shown in Figure 4.7, when the energy constraint in the network increases i.e. the duty cycle decreases, the average delay per hop increases for SCPN and much faster than that of CPN.

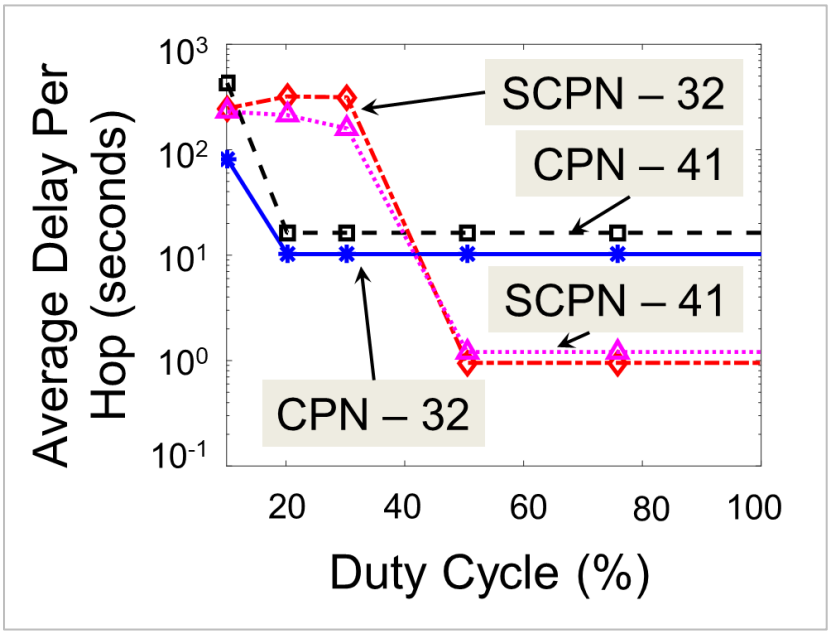


Figure 4.7. Effect of Energy Constraint on Per Hop Delivery Latency

At the highest energy constraint i.e. duty cycle 10%, the delay for SCPN and CPN are approximately similar. This nature of the latency is a consequence of the fact that with energy

constraint involved, the per hop latency not only a function of the frame structure, but also related to the delay brought about due to energy-aware event buffering. When the energy is low, irrespective of the protocol, the latency is large because it is primarily dictated by the time it takes for the nodes to acquire enough energy to transmit the pulses needed for event information transport. Hence, for either case of SCPN vs CPN, the latencies would be similar which masks the disadvantages of the longer frame structure of CPN compared to SCPN. In some low energy cases, as in DCs < 40 % in Figure 4.7, the delivery delay from SCPN can be higher than CPN. This is because, since more frames are completed for SCPN which involves more sync pulses as well as more attempts at node discovery, SCPN uses up more energy compared to CPN in the same time. This affects further transmission along the route and thus effective delivery delay. The observation from this graph is important because it indicates two things – one, that SCPN is roughly on par with CPN at high energy constraint, but is far more advantageous in network situations with low energy constraint; secondly, SCPN has lower increase in delivery delay with increase in network cell size irrespective of energy constraint.

D. Energy Consumption

It is to be noted that the energy consumption for both CPN and SCPN are roughly similar throughout the frame structure which is by design. In CPN, for a single hop transmission, the energy consumed by the network includes the transmission energy for the source / forwarding node and the reception energy for the receiving nodes i.e. idle listening. The number of pulses that need to be transmitted per event from a source / forwarding node for both CPN and SCPN are equal. For example, assuming a route diversity of 2, SCPN would have to transmit 4 pulses - 1 pulse for sync, 2 pulses for next hop and 1 pulse for event origin. Similarly, CPN would also involve 4 pulses – 1 pulse for sync, 1 for control, 2 for localization. Also, the number of slots that the

receiving nodes need to be idle listening for events is the same – for a network with M non-sink cells, CPN receivers must be idle listening in all the slots in control frame i.e. M slots and 2 slots in Localization Frame, i.e. a total of M + 2. Similarly, in SCPN, receivers would be idle listening for 2 slots in NHSF and M slots in EOSF i.e. M + 2. Hence, the total energy consumption due to transmission and idling is identical for both protocols. SCPN stands to make some extra energy consumption when overhearing is involved as explained earlier. However, with proper strategies in place such as merging and staggering of events, such extra consumption is essentially removed as seen in our simulations. Thus, using SCPN, we can obtain better network performance with similar levels of energy consumption when the energy constraint is not very high.

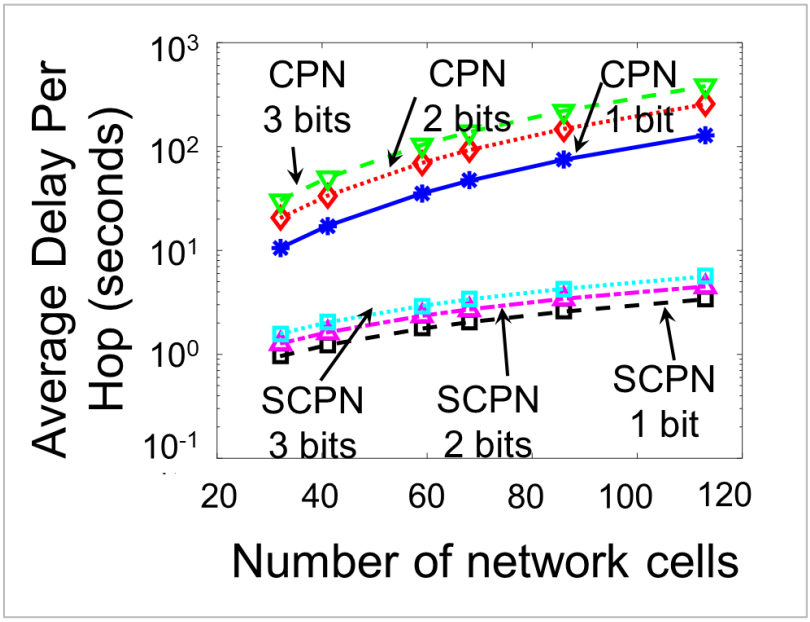


Figure 4.8. Effect of Information Content Size on Delivery Latency of Pulse Protocols

E. Higher granularity information transport

As mentioned previously, using the better scaling provided by SCPN we can include more information within the frame structure without sacrificing on the energy efficiency or latency. This

is demonstrated in Figure 4.8 above. In the baseline case, we use only 1 bit of event data i.e. the origin event area id. We can include further bits of information e.g. whether the available node energy is over / under a threshold etc. If such information were to be included in the CPN frame structure, the delay would increase significantly as demonstrated in Figure 4.8 for 1-bit vs 2-bit and 3-bit information. In SCPN though, such increase of information density does not need to come at a significantly higher latency price. In fact, even 2-3 bits of event information can be used in SCPN with a latency far less than the latency for 1-bit info using CPN. Such advantages are more pronounced when the network cell count is larger.

4.3. Summary

In this chapter of the thesis, we developed a novel time frame structure for discrete-pulse-based networking that scales well with increase in the number of network cells. We also evaluated the performance of the new scalable pulse networking architecture vs the baseline synchronous cellular pulse networking approach introduced in Chapter 3. We have shown that the SCPN protocol can provide similar energy-efficiency to the CPN while lowering the delivery delay. The protocol can also be reliably used in a wide range of harvesting situations. In the next chapter, we will discuss energy awareness features that should be included in the Baseline Synchronous Cellular Pulse Networking approach for more judicious energy management in extreme resource-constrained or dynamic energy availability scenarios. We will detail the energy aware protocol design and show the advantages of these new syntaxes through simulation experiments for a vibration-harvesting-powered network.

CHAPTER 5: ENERGY-AWARE PULSE NETWORKING

In earlier chapters of this thesis, we have mentioned that we envision the use of networks of sensors powered by ambient energy harvesting. For instance, in Structural Health Monitoring applications, such harvesting can be done from ambient structural vibrations using piezoelectric transducers [85]–[88]. The advantage in a harvesting-powered network is that the individual sensors can be less expensive and bulky (because large batteries need not be included) while being practically maintenance-free (as energy availability is theoretically limitless). However, such ambient harvesting sources are by nature very erratic and energy availability cycles might not be well-synchronized with energy use cycles. As a result, in absence of appropriate energy management, there might be leakage and eventual sensor power loss affecting effective network lifetime. In this chapter, we will develop an energy-aware pulse networking protocol which can deal with these shortcomings of the baseline synchronous pulse networking architecture. We will show that in systems powered by various levels of energy harvesting availability, the energy-aware version of the protocol performs significantly better compared to the baseline version. We will also show how the performance varies for different harvesting scenarios based on vibration harvesting on a simple rectangular plate structure.

5.1. Energy-aware Pulse Switching Protocol

This section introduces specific protocol syntaxes developed to deal with stochastic energy harvested, for instance, from the ambience (e.g., vibration) of a target structure. As mentioned in Chapter 3, the synchronous pulse networking architecture consists of the use of discrete pulses to indicate event information. The baseline architecture does not change transmission schedules based on energy availability. When an event is available, transmissions are attempted. In case of energy-constrained scenarios, for example, slow harvesting, the baseline protocol would attempt

transmissions even when enough energy is not available to transmit all necessary component pulses. Since the transmission is incomplete, this energy is essentially wasted. In the proposed energy-aware version of the protocol, the main idea is to manage the stochastically available energy in a prudent manner such that the wastage on incomplete pulse routing is minimized while the number of fully routed events is maximized.

5.1.2. Frame Adaptation

The same frame structure, as discussed in Chapter 3 without the *Route Discovery* and *Response Module*, is used for this energy-aware version. The route discovery part is omitted due to its energy intensiveness. A static routing operation with pre-defined routing table entries is chosen instead. The response area which is responsible for dealing with pulse losses is also removed. Given the low pulse loss rates (PLR) of the physical layer proposed i.e. ultrasound communication [8], it was decided that the benefits of the response mechanism may outweigh its energy overheads. Thus in the revised frame format, every frame starts with a *Sync Area* followed only by the *Control*, *Localization* and *Protection* areas.

5.1.3. Binary Event Buffer

Due to insufficient energy, a pulse sensor network can become partitioned i.e. a pulse may arrive at a node but cannot be forwarded due to the lack of energy required for transmission. To accommodate such scenarios, a binary event buffer is provided which can be used for storing such a pulse until enough transmission energy is available for a forwarding operation. As shown in Figure 5.1, the length of the buffer at each node is M, where M is the total number of sensor cells in the network. At node-j, the i-th buffer bit is set to 0 or 1, depending on the absence or presence of a pulse originated at sensor cell i, buffered at node-j. At the start of each frame, every node first

checks whether there are any buffered events, and tries to transmit them in the respective slots based on its energy availability. When a pulse is generated or received (for forwarding) at a node, it is stored in the binary event buffer in the bit corresponding to the cell of its origination. If the *i*-th bit of the buffer is already 1 in a node, and a new pulse originated at Cell-Id-*i* is received by the node, it simply ignores the new pulse without any changes in the buffer. This causes the old event to be merged with the new one at this point in the route. Such merging results in a loss of temporal resolution of event detection. For many structural health-monitoring applications, however, such a loss can be acceptable due to their low requirements on temporal granularity of event detection.

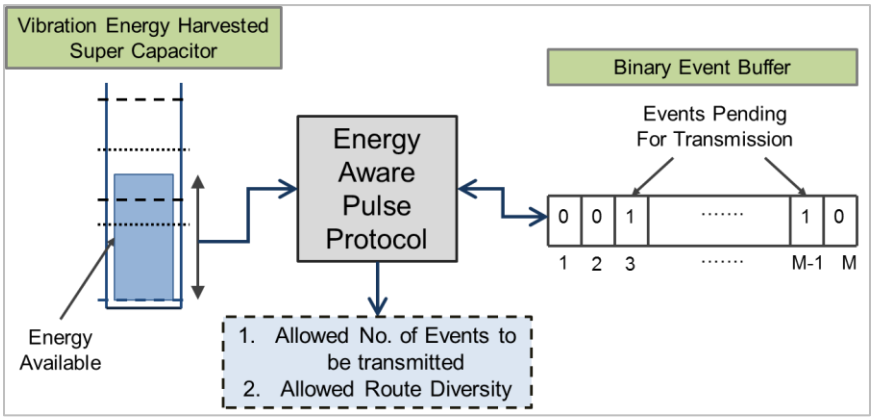


Figure 5.1. Summary of Energy-Aware Pulse Switching

5.1.4. Energy-aware Forwarding Syntaxes

Each networked sensing unit contains an energy storage super capacitor that is charged at a rate based on the harvesting source. For structural health monitoring applications using Through-Substrate Ultrasonic Pulse Networking (TUPN) units as discussed in Chapter 1, this harvesting source can be the ambient vibrations in the structure where the TUPN is embedded. All baseline forwarding decisions, as explained in Chapter 3, are modulated by the amount of energy available in the super capacitor and the estimated energy expenditure for sending the pending events that are

buffered in the binary event buffer. For example, the event buffer in Figure 5.1 indicates buffered/pending events from cell IDs 3 and M-1 that need to be forwarded. The protocol estimates the energy needed to send those event pulses with different route diversities (as defined in chapter 3) and makes selective forwarding decisions based on that estimation and the available energy in the super capacitor.

While checking for available energy, the protocol first ascertains whether there is enough energy to transmit all pending events with the highest allowed RD, and if not then with the next lower RD and so on. If energy is not sufficient to transmit even with lowest RD (i.e., 1), then the node tries to transmit one event less than the number of pending events, with RDs starting from the highest and so on. If not even one event can be transmitted with the lowest RD, the node suspends transmissions in the current frame. Total energy consumption E_c is estimated as:

$$E_c = (n_{ctrl} + n_{loc}) * e_{pulse} + p_{idling} * F \quad \ldots (5.1)$$

where n_{ctrl} and n_{loc} are the number of Control and Localization area pulses respectively, e_{pulse} is the energy consumption for a single pulse generation, p_{idling} is the idling power, and F the frame duration. The idling energy consumption for a full frame is considered as a conservative estimation to remove the chance of nodes running out of energy in the middle of a frame. It is to be noted that n_{loc} in the above equation is a function of the Route Diversity (*RD*), and hence estimates the energy expenses for different route diversity values. Depending on the amount of energy in the super-capacitor and the estimated amount needed, a node may be able to forward only a subset of all the buffered events. To ensure fairness, the events from the least recently used bits in the event buffer are selected for transmission first.

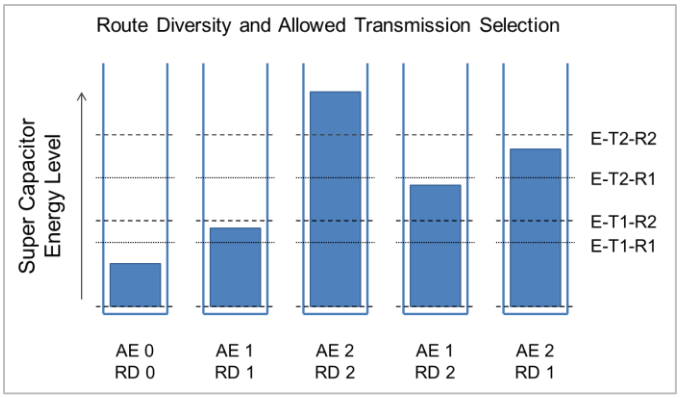


Figure 5.2. Routing Decision based on Energy Levels

Figure 5.2 shows different energy availability situations in the super-capacitor and how they influence the routing decisions as per the protocol described above. A level *E-Ti-Rj* indicates the estimated energy needed for transmission of *i* events with a route diversity *j*. Each column in the figure shows an energy availability scenario (i.e. the y-axis levels) and a corresponding forwarding decision, marked as a *{Allowed Events (AE), Route Diversity (RD)}* pair. After the allowed events are subsequently transmitted, they are removed from the event buffer.

5.2. Performance Results

An event-driven C++ simulator implementing the MAC-Routing framing was developed. Both the non-energy aware (NEA) and energy aware (EA) versions of the protocol were simulated. Note that the baseline energy advantages of pulse switching over traditional packet-based event reporting have already been established in [4, 5]. In this chapter, we mainly validate and characterize the effectiveness of the energy-awareness of the architecture in a harvesting scenario. We first consider a scenario where the energy generation is synchronous across the network i.e. all nodes receive similar energy harvesting. In the next scenario, we cover the efficacy of energy-aware pulse networking in scenarios where the energy generation is non-uniform across the network.

5.2.1 Results with Synchronous Energy Generation

A. Model for Vibrational Harvested Energy – Synchronous

The objective is to develop pulse network protocols that can operate using sparse energy, harvested from system ambience such as aircraft wing vibration. A piezoelectric harvesting mode using synchronous vibration is assumed. This implies that all the TUPN units mounted on a structure experience similar levels of ambient vibrations and resulting charging of their respective super capacitors. Discharging of the capacitors, however, will depend on the generated events and their resulting spatial-temporal patterns in different parts of the structure. An on-off charging model is used for evaluation of the protocol performance. In this model, only the 'on' duration corresponds to the availability of structure vibration sufficiently intense for harvesting.

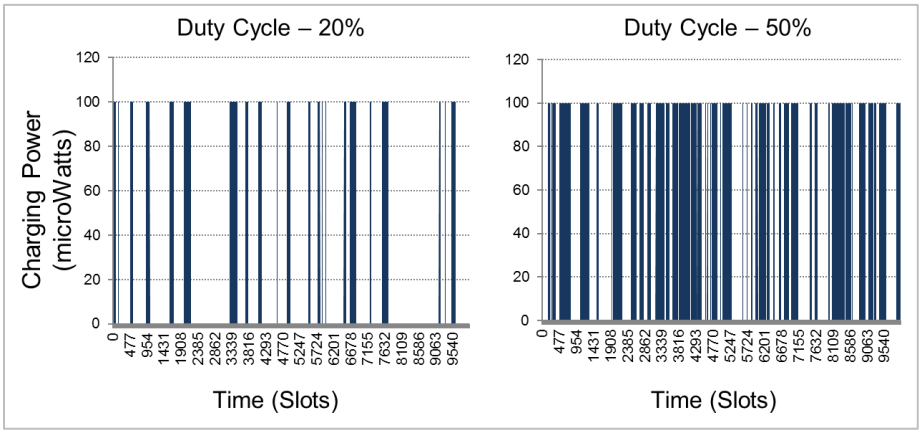


Figure 5.3. Capacitor charging for different duty cycles

To simulate memory-less vibration patterns, exponentially distributed random processes are used for generating both the 'on' and 'of' periods of the vibration profile. The level of harvesting is parameterized by the average duty cycle of the on-off pattern. A higher duty cycle would correspond to an environment with a higher vibration level indicated by longer 'on' durations. The amount of energy harvested during the 'on' period depends on the specific

piezoelectric harvesting mechanism and its efficiency [89]. For the results in the next section 4.2., we use a specific harvesting rate of 100 μW during the 'on' periods. The corresponding capacitor charging patterns for two different vibration duty cycles are shown in Figure 5.3. The presented protocols were evaluated with different harvesting/charging rates representing different harvesting technologies and their efficiency.

B. Network Structure and Event Generation

The simulated network consists of 210 evenly distributed TUPN sensor nodes with a sink node placed at the lower left corner of a rectangular terrain that measures *6m x 3m*. The nodes are grouped into 84 regular hexagonal cells with an average of 3 nodes per cell. The spacing between individual nodes is around 0.3 m and the transmission range is kept at around 0.75 m. Height of each hexagonal cell is 0.5 m with cell side length of around 0.288 m.

Events are generated at different network locations. For each event, a pulse is sent to the sink repeatedly with a pre-specified inter-pulse interval, such that even if a specific pulse is lost on its way due to insufficient energy availability or unreliable ultrasonic links, successive transmissions will increase the probability of successfully informing the sink about the event. Experiments were conducted with a variety of such *pulse repetition intervals* in order to understand their implications on event delivery latency under different energy constraints.

C. Network Node Energy Traces

Simulation experiments were performed using three duty cycles (DC) i.e. 20%, 30%, 40%, and three different pulse repetition intervals (PRI) i.e. 20, 10, and 5 frames for both NEA and EA pulse switching. The energy budget for a single pulse transmission was chosen as 10 mJ, and constant idling power consumption as 1μW, which includes the power consumption for reception

of pulses. A harvesting energy rate of 100 μW was used during the 'on' period of the fully synchronous structure vibration as discussed in Section 4.1.5. All the graphs in Figure 5.4 demonstrate the remaining energy in the super capacitor of a node on the event delivery route. The first notable observation is that for the same vibration DCs (i.e. degree of harvesting energy constraint) and PRIs, the NEA protocol often leads the super-capacitor to zero-energy condition, whereas the EA version manages to avoid it.

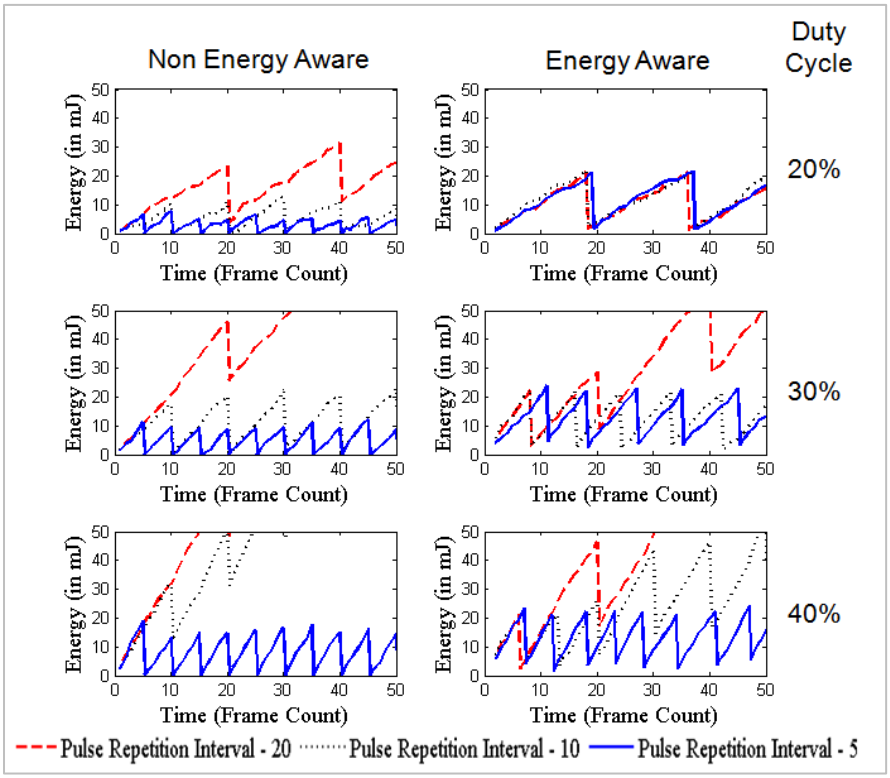


Figure 5.4. Energy trace at a node on event propagation route

It should also be noted that irrespective of the duty cycle, the consumption cycles in the NEA protocol operation are more sensitive to the pulse repetition intervals. For example, when the event generation interval is 5 frames, there is visible energy consumption after every 5 frames in the NEA protocol at all DCs, indicating pulse transmissions. However, at the lower DCs, the observed drop in energy is not enough for transmission of all the pulses, which requires at least 20

mJ for the lowest RD. This indicates that only a subset of all the needed pulses for an event are being sent. As a result, certain amount of energy is wasted even though the event is not successfully transmitted. This can affect future event transmissions in an adverse manner. This is especially true in the low DC scenarios where there is never enough energy to transmit all the pulses for an event at the lower PRIs (i.e., 5 and 10 frames).

For the higher PRIs, since the node has more time to accumulate energy between successive event transmissions, incomplete transmissions are less frequent. In contrast, for the EA protocol, irrespective of the PRI, consumption happens only when sufficient amount of energy (i.e., 20 mJ) has been accumulated for sensing all the pulses needed for an event transmission. This avoids wastage caused by partial transmissions and therefore improves the overall energy economy. In all scenarios, as the duty cycle increases, the operation of the EA protocol becomes similar to the NEA case because of higher energy availability.

D. Event Reporting Performance

Figure 5.5 depicts the event reporting delay, which is the primary performance index of the system. Such delay is defined as the latency between the transmission of first pulse for an event from the source node and the first time that particular pulse or a subsequently repeated pulse for that event arrives at the sink node. This indicates the effective delay in reporting a structural failure event. Such delay is caused due to: 1) end-to-end routing delay, 2) pulse losses because of channel errors, 3) deferred/stalled pulse routing due to event buffering caused by energy unavailability in the energy-aware routing. Due to the pulse repetition feature of the event generation model, and the event buffering in the energy-aware routing protocol, an event is guaranteed to be reported with infinite delay.

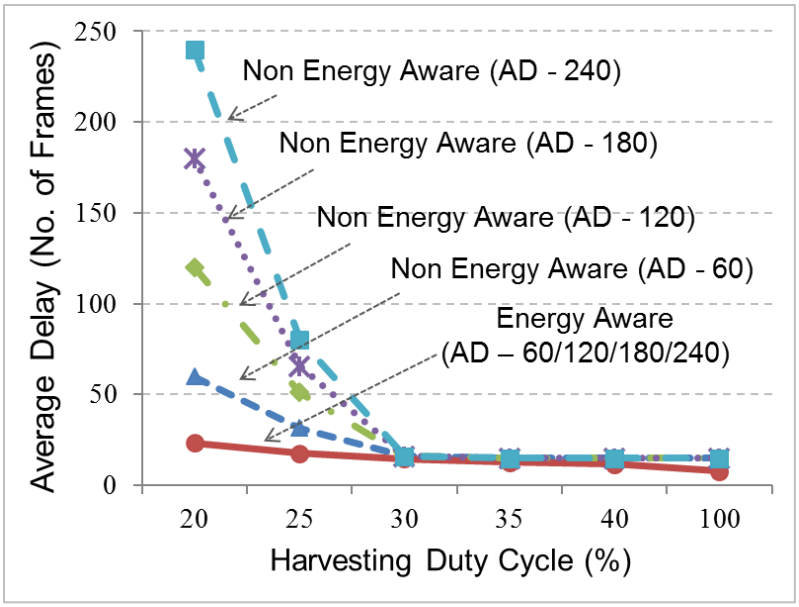


Figure 5.5. Event Reporting Delay for Different Duty Cycles

To accommodate practical settings, however, a notion of Allowed Delay (AD) has been developed while defining the concept of successful reporting. An event is regarded as successfully reported if it reaches from source to sink within a certain Allowed Delay. Figure 5.5. reports results for four different ADs, namely, 60, 120, 180 and 240 frames. For the required frame duration of 76 sec corresponding to our network, in Figure 5.5 those delays correspond to approximately 1, 2, 3, and 4 hours respectively. Those AD numbers are deemed reasonable for typical structural health monitoring applications. Observe in Figure 5.5, for all vibration DCs and ADs, the EA version of the protocol reports events quicker than the NEA version. Also, as expected, with higher AD limits, the average reporting delay for the NEA protocol increases, as more events can be considered as successfully delivered in the longer intervals. The EA protocol, however, appears to be insensitive to the AD. This is because of its ability to deliver all events with reporting delays that are well below the lowest AD even at the lowest vibration DC. We do not consider DCs below 20% because in such cases events are never delivered within the highest AD for the NEA version. Note that in

all the scenarios, for high DCs (i.e., 30% and above) both EA and NEA protocols behave similarly. The reporting delay stabilizes for both protocols since they are able to deliver all events successfully within the lowest AD due to high energy-availability. It is notable that improvements in delay demonstrated by the EA protocol are more pronounced under severe energy constraints, i.e. the low DC scenarios. Improvements are also higher with higher ADs. This can be explained using the next set of results showing the event delivery ratio in Figure 5.6.

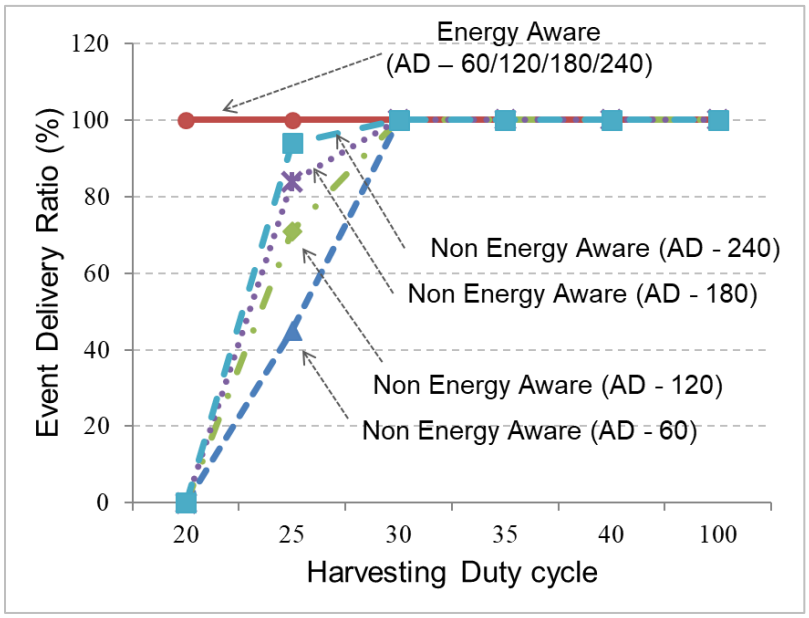


Figure 5.6. Event Delivery Ratio at Different Duty Cycles

Event delivery ratio is defined as the fraction of generated events that are successfully reported to the sink for a given AD. As can be seen in Figure 5.6, for the NEA protocol, the delivery ratio improves with more structure vibration up to around DC 30%. Delivery ratio also improves with higher AD. Beyond that point, the system enjoys enough harvested energy, which results in near-100% event delivery. The EA protocol, however, is significantly less sensitive to both vibration DC and AD because of its judicious energy usage and event buffering as presented in Section V. In summary, the proposed EA protocol syntaxes show significant improvements over

its NEA counterpart in low structural vibration situations. Such improvements are consistent for both event reporting delay and delivery success.

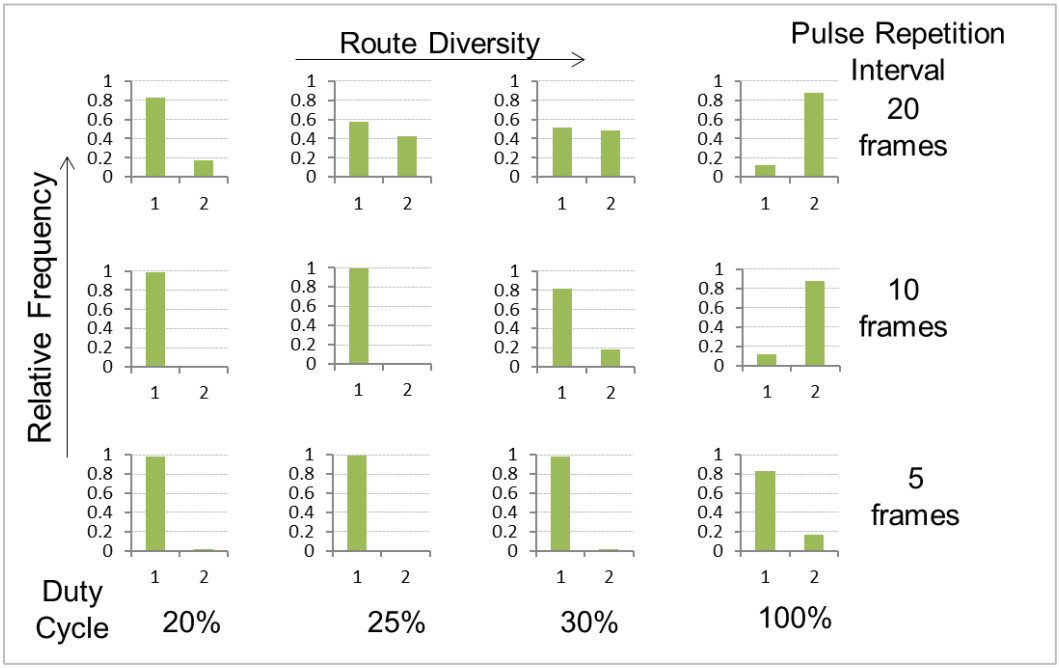


Figure 5.7. Route diversity for different energy availability

E. Impacts of Adaptive Route Diversity

A key component of the EA protocol is its ability to choose appropriate route diversity (RD) based on the instantaneous energy availability. Figure 5.7 reports the occurrence distribution of RDs under different vibration DCs and PRIs. It can be observed that for a given PRI, higher diversity routes occur more frequently as energy becomes plentiful (i.e. higher vibration DCs). For a given energy availability (i.e. DC), routes are more diverse for larger PRIs and less for smaller ones. This is because with more frequent pulse repetitions (small PRI) at the source, there is a bigger drain on the available energy, which prevents the protocol from choosing energy-expensive higher route diversities. These results validate the protocol's ability to choose pulse RD in a manner that is adaptive to the instantaneous available energy. For the NEA version, such diversity

is statically chosen and can cause energy wastage leading to higher reporting delays and lower delivery success, depicted in Figs. 5.5 and 5.6.

F. Event Buffering Characteristics

In Section 4.1.3, we presented how event buffering can be leveraged in the absence of sufficient pulse forwarding energy in the proposed protocol for reducing event losses and the resulting reporting delay reductions.

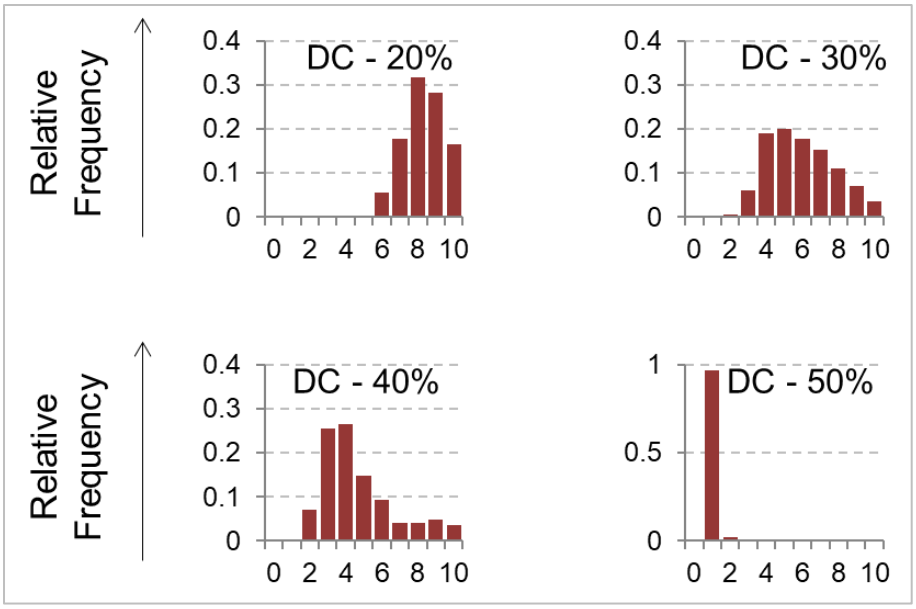


Figure 5.8. Event buffering/storage delay (in frames) distributions

Figure 5.8 reports an example distribution of the per-hop buffering delay (i.e., averaged over all the hops on an end-to-end route) experienced by an event on its way to the sink node. As can be seen from the distribution graphs, the distribution's peak moves towards the lower end of the buffering delay as the vibration DC increases. This implies that when energy is scarce (i.e. low DC), the protocol employs longer storage times to allow sufficient accumulation of harvested energy before transmitting all the needed pulses for complete event forwarding. As the energy

becomes more abundant, the need for storage becomes less important and storage times approach the lower limit of one frame per hop. These observations validate the event-buffering process and its intended impacts.

5.2.2. Results with Synchronous Energy Generation

A. Model for Vibrational Harvested Energy – Asynchronous

The main objective of following evaluation was to evaluate the performance of Energy-Aware Pulse Switching protocol in dynamic energy harvesting scenarios without any limitations like synchronous energy harvesting in all nodes of the network. To that effect, a few representative vibration harvesting models were constructed for the evaluation experiments.

In order to simulate an asynchronous vibration energy harvesting model, we used an exponentially distributed ON OFF charging (energy accumulation) model modulated by a spatial distribution function. The exponentially distributed ON OFF model helps to simulate the temporal variation in vibration (and hence, charging) while the spatial distribution function incorporates the spatial variation in harvested energy profiles.

In order to parameterize the temporal variation in harvesting efficiency, we define a parameter called Duty Cycle as follows –

Duty Cycle = Average of ON time distribution / (Average of ON time distribution + Average of OFF time distribution),

where the ON time and OFF times of the model are both exponentially distributed to emulate memory less processes mirroring the fact that structural vibrations available are essentially stochastic in nature and not correlated. Duty Cycle, therefore, gives us a measure of the harvesting

efficiency of the system on a temporal level. The spatial distribution function is multiplied with the duty cycled energy to obtain a spatiotemporally variable vibration profile.

We have used four different spatial distribution functions (as detailed in Equations 5.2 – 5.5) to evaluate the efficacy of Energy-Aware Pulse Protocol operation when powered by a variety of vibration profiles. These vibration / harvested energy profiles correspond to the four anchor configurations for the rectangular plate as shown in Figure 5.9 (a)-(d).

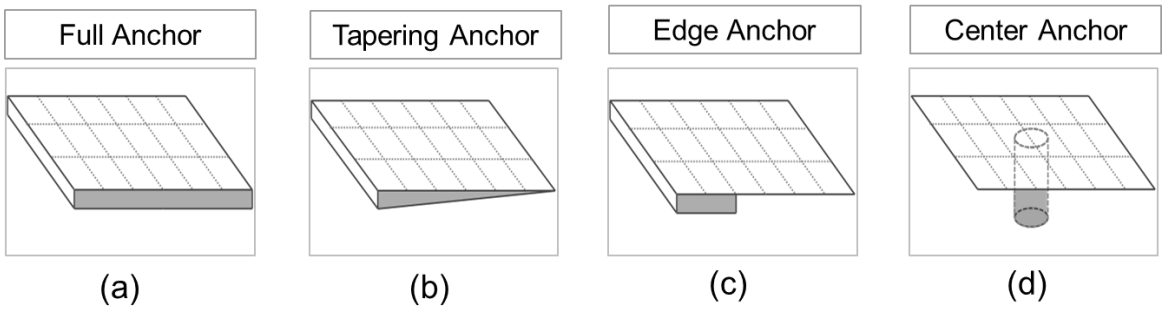


Figure 5.9. Rectangular plate designs with various anchor configurations

In each equation below, the energy harvested during a single ON time of the vibration profile is evaluated. This energy is accumulated over the course of time as more ON times are encountered, leading to energy harvesting in a particular node. In relation to the equations below, the definition of component parameters has been provided in Table 5.1.

Table 5.1. Asynchronous Harvesting Model Parameters

Parameter Name	**Parameter Representation**	**Parameter Definition**
Charge Interval	t_{charge}	ON time interval (modulated by the duty cycle)
Charging Power	P_{charge}	Harvesting power during the ON time (assumed to be constant)
-	m, c, k	parameters determining the spatial variation

Table 5.1. (cont'd)

-	x	X coordinate of the node whose energy is being calculated
-	y	Y coordinate of the node whose energy is being calculated
-	x_i	X coordinate of the node i
-	y_i	Y coordinate of the node i
Energy Charged	e_{charge}	Energy accumulated in one Charge Interval

$$e_{charge} = (mx^0 + c) * t_{charge} * P_{charge} \quad \ldots (5.2)$$

$$e_{charge} = (mx^1 + c) * t_{charge} * P_{charge} \quad \ldots (5.3)$$

$$e_{charge} = (mx^2 + c) * t_{charge} * P_{charge} \quad \ldots (5.4)$$

$$e_{charge} = [(x - \text{mean}(xi))^2 + (y - \text{mean}(yi))^2] * t_{charge} * P_{charge} \quad \ldots (5.5)$$

The first configuration (Fig 5.9 (a)) assumes that the anchor is continuously attached along the whole length and breadth of the rectangular plate. As a result, there is no spatial variation involved as indicated by the constant value of the Harvesting Power Multiplier in Fig 5.10 (a). This is because all the area of the plate experiences a consistent level of vibration due to similar anchoring configuration throughout. This effect is modeled in the Equation (5.2).

In Figure 5.9 (b), an anchor configuration is demonstrated which tapers down from the left edge of the wing to the right edge. The anchor width is unchanged along the y-axis for all points which are located at a fixed distance from the left edge. Such a variation is modeled using the Equation (5.3) with the corresponding harvesting multiplier variation shown in Figure 5.10 (b).

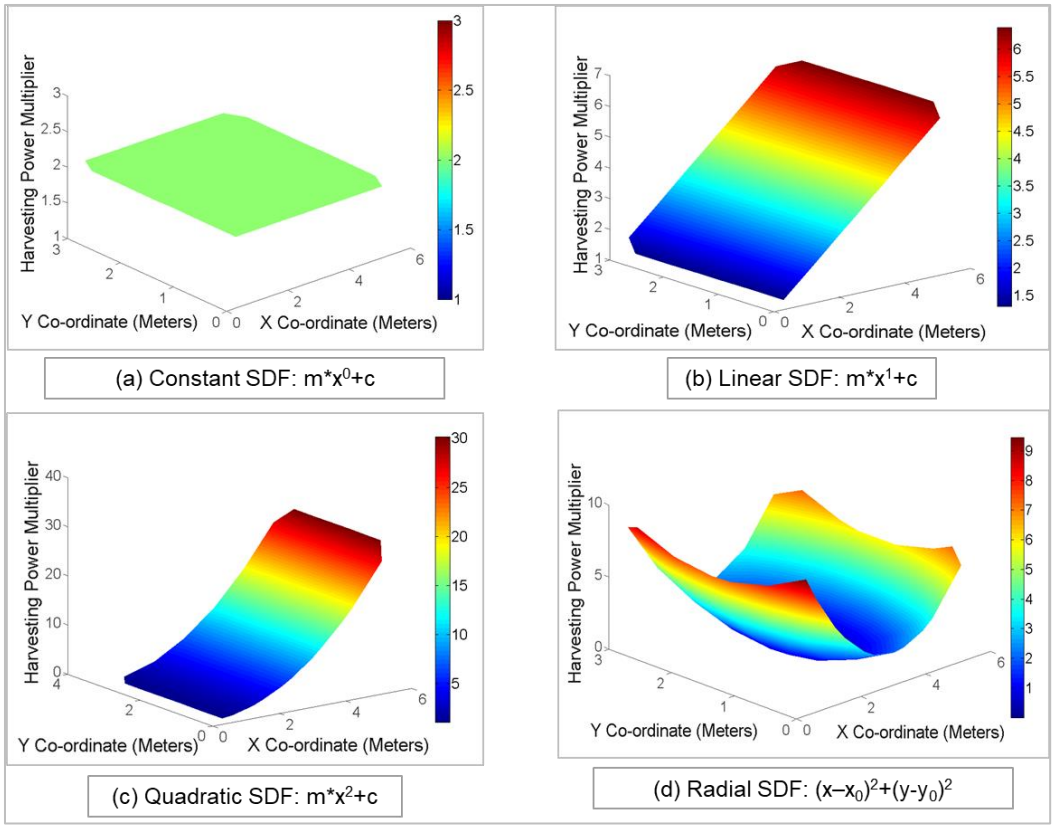


Figure 5.10. Spatial variation of node charging profiles at a fixed duty cycle

The third scenario illustrates a case where the anchor is present only near the left edge of the plate as shown in Figure 5.9 (c). As a result, rate of increase of vibration levels is higher when we move farther from the left edge. The corresponding harvesting power multiplier is demonstrated in Figure 5.10 (c) where the slope increases in quadratic manner in accordance with Equation (5.4).

Figure 5.9 (d) demonstrates a slightly different anchoring scheme where the plate is affixed at the center instead of along the edge. This is represented by the charging equation in (5.5) where the energy accumulated depends on the position away from the center (in this case calculated using the mean of all the node positions since they are uniformly distributed along the two axes). The corresponding harvesting multiplier variation is shown in Figure 5.10 (d). As mentioned above,

the spatial variation in the equations (5.2) - (5.5) is demonstrated in the following temperature maps where the harvesting power multiplier based on the equations is plotted versus node location. This represents the available harvesting power distribution across the rectangular plate at a constant duty cycle or alternatively at a certain instant in time.

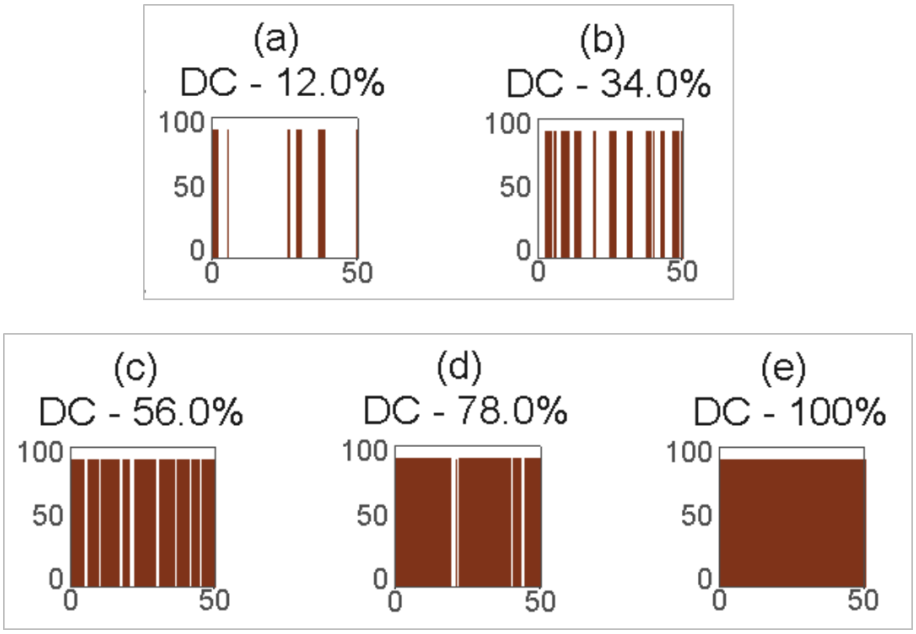


Figure 5.11. Temporal variation in harvested energy availability using a duty cycled approach

The temporal variation (change in duty cycle) for a specific spatial variation is demonstrated in Figure 5.11. As we move from Figure 5.11 (a)-(e), the duty cycle increases and as a result we can see that the temporal availability of harvesting power available is higher i.e. harvesting occurs at a higher rate leading to faster accumulation of energy.

As we will show in the following sections, the spatial distribution function and the temporal harvesting efficiency combine to create a spatiotemporal variation in the harvesting profile which is then used to evaluate the operation of the Energy-Aware Pulse Switching Protocol when powered by the same.

B. Network structure, simulation parameters and event generation model

The simulated network consists of 1057 TUPN sensor nodes evenly distributed on a rectangular plate structure with a sink node placed at the center left corner as shown in Figure 5.12. The nodes are grouped into 84 regular hexagonal cells (like Figure 5.12) with an average of 3 nodes per cell. The spacing between individual nodes is around 0.3 m and the transmission range is kept at 0.75 m. Height of each hexagonal cell is 0.5 m with cell side length approximately 0.288 m. An energy model is used where the energy for transmitting a single pulse is 10 mJ while the reception energy budget is 10 μW (10uJ/sec) including the energy required for pulse detection and idle listening.

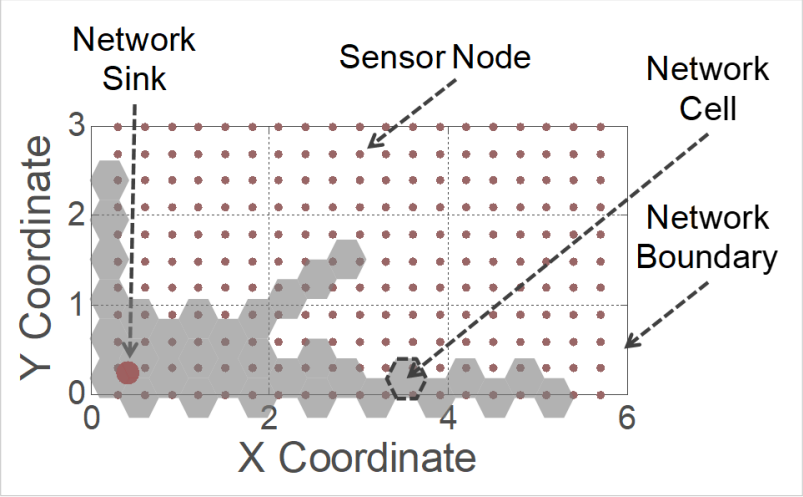


Figure 5.12. Cellular network model on a rectangular plate structure

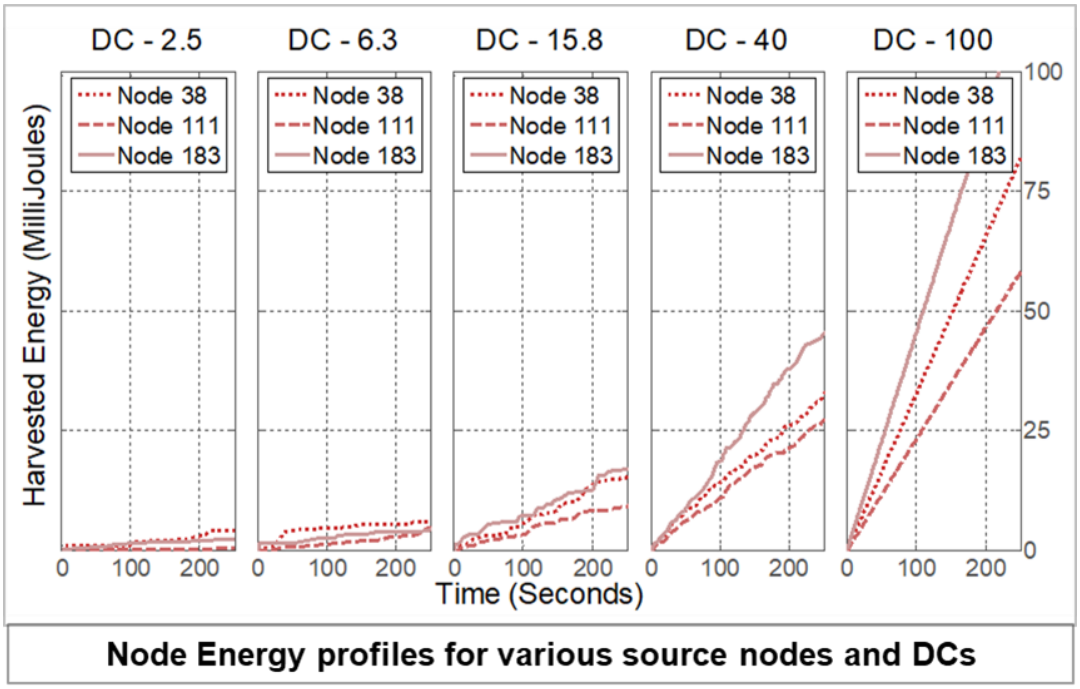
Events were generated at different source nodes across the rectangular plate and the corresponding performance characteristics were noted. Events were assumed to be far apart from each other to be considered as independent transmissions. For event transmission from a particular source node, 100 similar experiments were performed, and their average noted to take care of the

stochasticity in the charging model introduced as part of the exponentially distributed components Duty Cycle formulation.

C. Network Node Harvested Energy Traces

In order to demonstrate the spatiotemporal variation of harvested energy across the simulated rectangular plate, the energy traces for three representative nodes on the structure were obtained at varying levels of energy availability (signified by the Duty Cycle of ON-OFF charging). For example, the energy traces shown in Figure 5.13, were obtained from simulations on the rectangular plate anchored at the center. It is evident from the graphs that the energy harvested increases monotonically with increase in the temporal frequency of vibrations represented by the Duty Cycle (DC). The spatial variation in energy harvesting is also easily discernible with nodes near the anchor (e.g. Node 111) enjoying much less in terms of harvested energy compared to nodes away from the anchor (e.g. Node 183).

It is to be noted that the former node always has lower levels of vibration being closer to the anchor location and thus consistently lower levels of harvested energy compared to the latter node irrespective of the duty cycle of vibrations. The positions of the selected nodes on the rectangular plate have also been indicated to demonstrate the dependence of node location on the harvesting rate. The figures thus demonstrate the spatial variation in energy profile (among the selected nodes at each duty cycle) as well as the temporal variation (for each node across the range of duty cycles observed). The range of duty cycles observed is divided into exponential intervals (0.025, 0.063, 0.158 etc.) such that the temporal variation for a large range of duty cycles can be demonstrated.



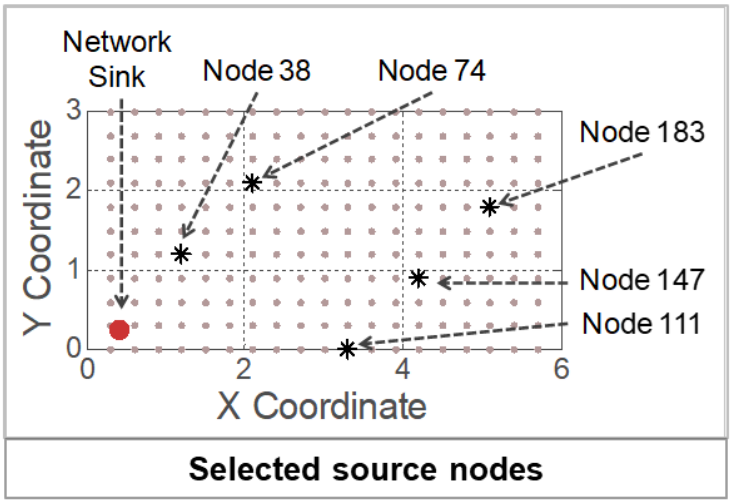


Figure 5.13. Energy traces for selected nodes on the rectangular plate

D. Networking Performance Characteristics

In this section, we evaluate the networking characteristics on using Energy-Aware Pulse Switching on the spatiotemporally variable harvesting profiles discussed in Section 3.

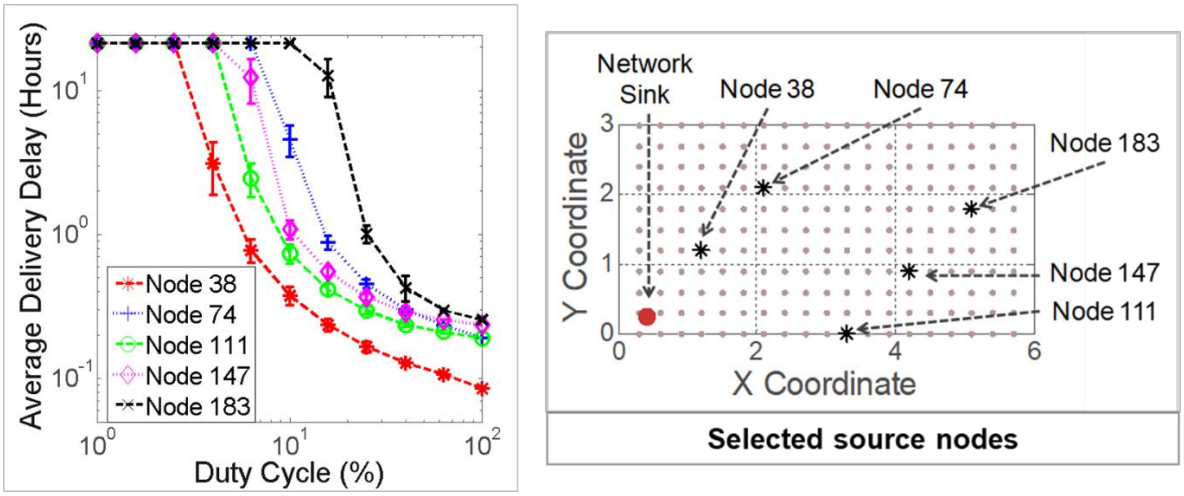


Figure 5.14. Delivery Delay characteristics for selected source nodes on the rectangular plate

In Fig 5.14, the average delivery delay goes down exponentially as the energy availability in the system is enhanced. Fig 5.14 shows the average delivery delay for 5 representative nodes on the surface of the plate with anchor configuration at the center. It can also be seen that the variability in the delivery delay increases with decrease in duty cycle. This is because with decrease in duty cycle, energy availability is more erratic, and the temporal variation becomes more pronounced in this case. As a result, even for the same source node for various node deliveries, event delivery delay can differ substantially.

In Figs. 5.15 and 5.16, we demonstrate the average harvested energy (over 100 seconds) and corresponding network delivery delay characteristics for events originated at source nodes across the rectangular plate using a temperature map.

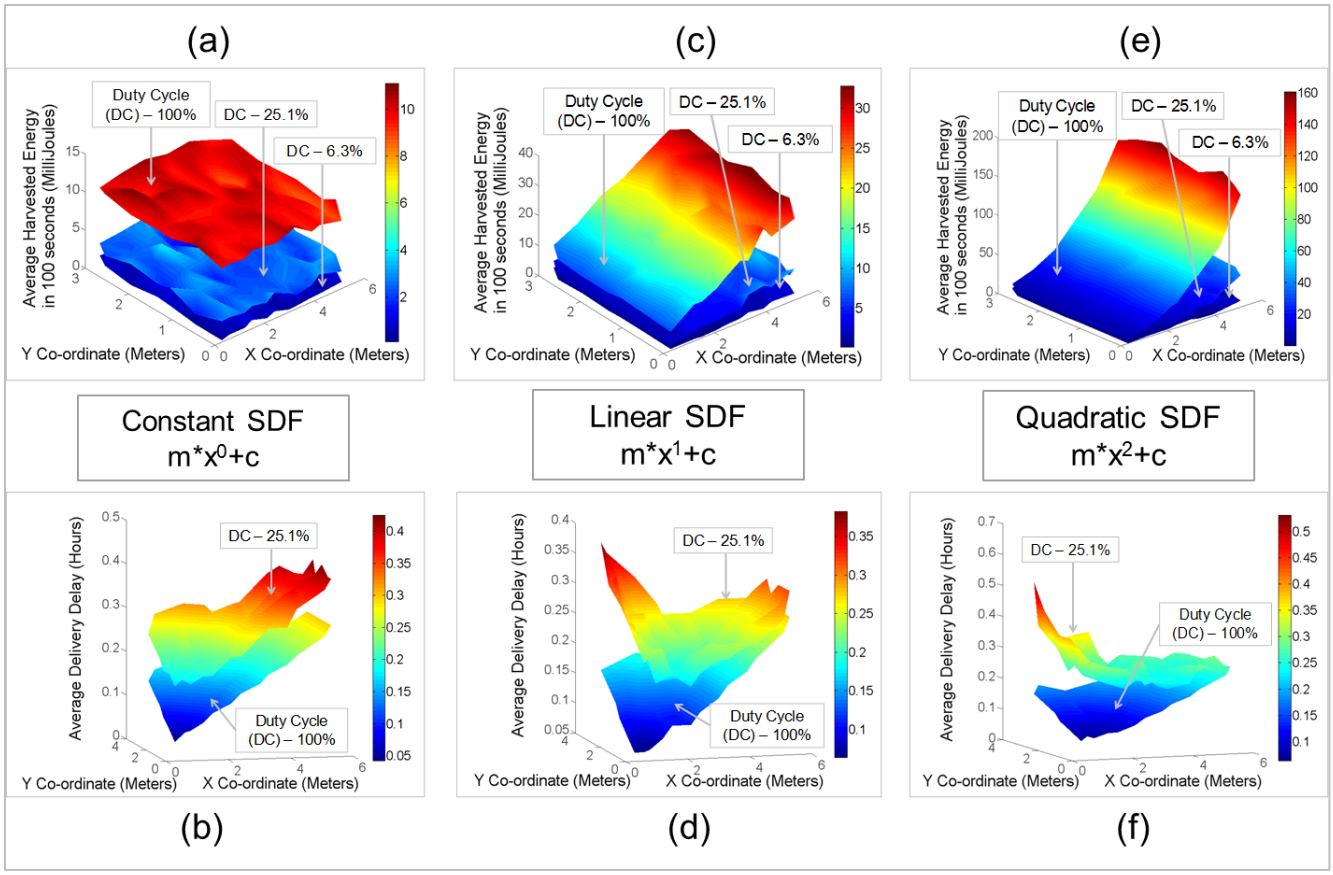


Figure 5.15. Average Harvested Energy and Event Reporting Delay distributions across rectangular plate with different edge anchor configurations

As the color moves from blue to red (i.e. height of the surface increases), the value of the appropriate parameter being showcased increases. Figure 5.15 demonstrates the energy generation and delay characteristics for the anchor configurations along edge of the rectangular plate (Figure 5.9 (a)-(c)) while Figure 5.16 demonstrates the same for the center anchor configuration (Fig 5.9 (d)). We also show the variation for separate duty cycles.

As seen in Figure 5.15 (a), the harvested energy generation is consistent over the whole area of the rectangular plate (based on spatial distribution function shown in Figure 5.10 (a)). As a result, the delivery delay characteristics are mainly modulated by the distance of the source nodes

from the sink. This is seen in Figure 5.15 (b), where the nodes near the right edge of the rectangular plate have much higher delays compared to those near the left edge because the latter are situated closer to the sink. In Figure 5.15 (c), we demonstrate a spatial variation in harvested energy which varies approximately in a linear fashion along the x coordinate of the wing (corresponding to the spatial distribution described in Figure 5.10 (b)). As a result, the delay characteristics are also changed from the previous scenario. As can be seen in Figure 5.15 (d), at a lower duty cycle, when the energy is at a premium, the location of the nodes near the right edge of the plate allows for an energy benefit which even trumps the disadvantage of being further away from the sink in distance. As a result, the delivery delay for events sourced at nodes near the right edge of the plate turns out to be lower than even those on the left edge which are closer to the sink. The latter nodes despite being closely situated to the sink, receive very low levels of energy being near the anchor of the plate which affects their event reporting delay. This effect is less pronounced when the energy availability is less constrained, and the delay characteristics resemble the characteristics in the constant spatial variation scenario. A similar trend is seen in Figure 5.15 (f), because the corresponding energy availability curve in Figure 5.15 (e), is even steeper in the x-coordinate direction which ensures that nodes on the right edge of the plate have delays even less than the previous case for the higher duty cycle scenario.

In Figure 5.16 (a), we show the average harvested energy for the anchor configuration of Figure 5.9 (d), corresponding to spatial distribution function as in Figure 5.10 (d). We consider the event delivery delay characteristics across the node for three different Duty Cycles 6.3%, 25.1% and 100% as shown in Figure 5.16 (b), (c) and (d) respectively.

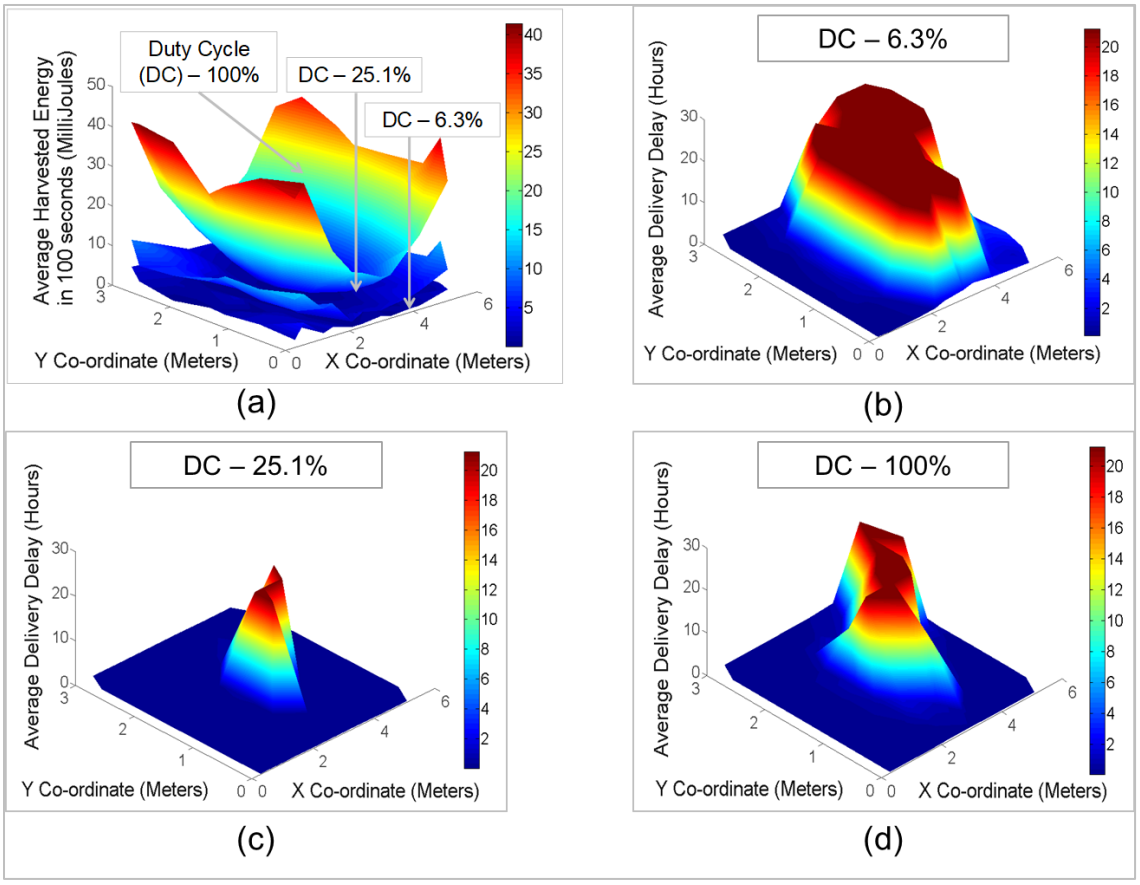


Figure 5.16. Distribution of Average Harvested Energy and Event Reporting Delay across rectangular plate with central anchor configuration

It is noted that nodes near the edges of the plate have lower delays consistently compared to those near the central regions. With the increase in duty cycle, the overall energy availability increases and as a result, even the central regions of the plate deliver events with less delay. The dark red regions correspond to higher delays which are notably seen to get thinner with increasing energy availability. The dark red regions extend from the center towards the right edge of the wing in the lower duty cycle scenarios (Figure 5.16 (b), (c)) because at these harvesting levels the distance of the nodes on the right edge of the plate from the sink also plays a role is modulating the delivery delay values. Overall, it is noted from the delivery delay characteristics that the network performance is a function of the energy availability and the Energy-Aware Pulse Protocol

works based on the energy availability to modulate the transmission and lead to this behavior as we will further explore in the following sections. The increased delay allows for enough energy generation for event propagation instead of event losses and that is the benefit of the energy awareness in the Energy-Aware Pulse Protocol.

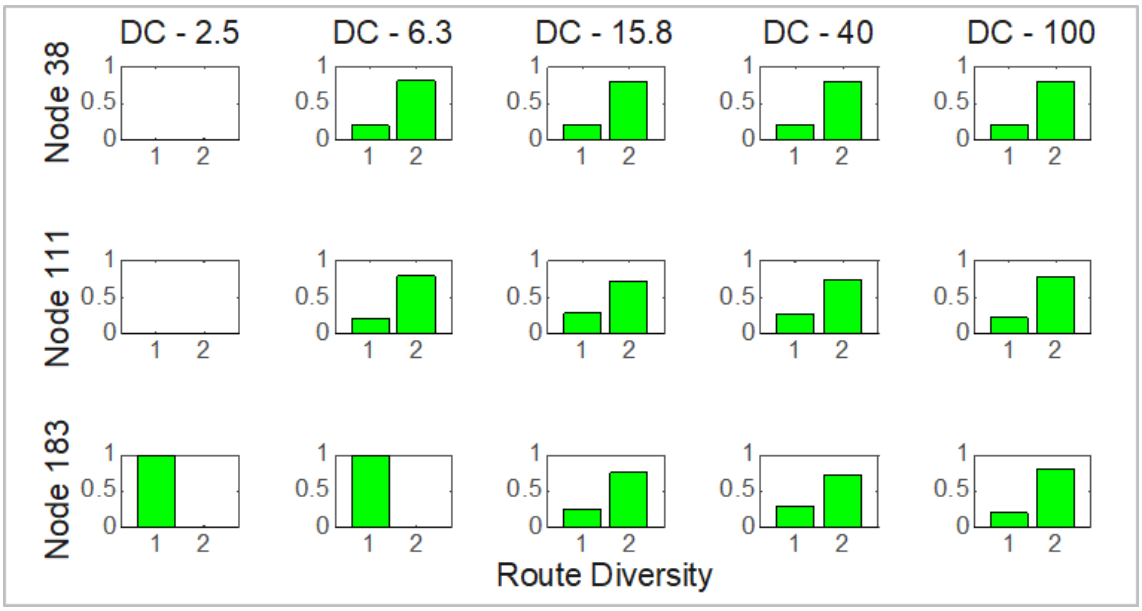


Figure 5.17. Per hop transmission route diversity distributions for selected source nodes

E. Route Diversity Distributions

The Energy-aware Pulse Switching protocol helps to modulate the energy usage of individual nodes depending on their available energy levels based on energy harvesting. When energy is in abundance, the protocol assures that the nodes utilize maximum number of routes to ensure higher reliability which is represented by use of a higher Route Diversity. When the energy is scarce, nodes conserve their power and stagger the communication by storing generated events in the binary buffer until energy becomes available. Even when energy is available, if it is just enough for transmission, the nodes select a lower route diversity sacrificing more reliable transmissions (with higher route diversity) keeping in mind the energy limitations. This is

demonstrated in the graphs of Figure 5.17 where the selected route diversity distributions have been shown for various levels of energy harvesting. When the duty cycle is high representing a faster rate of energy harvesting, the nodes have a higher percentage of transmissions with greater route diversity compared to lower duty cycle cases where the energy availability is limited.

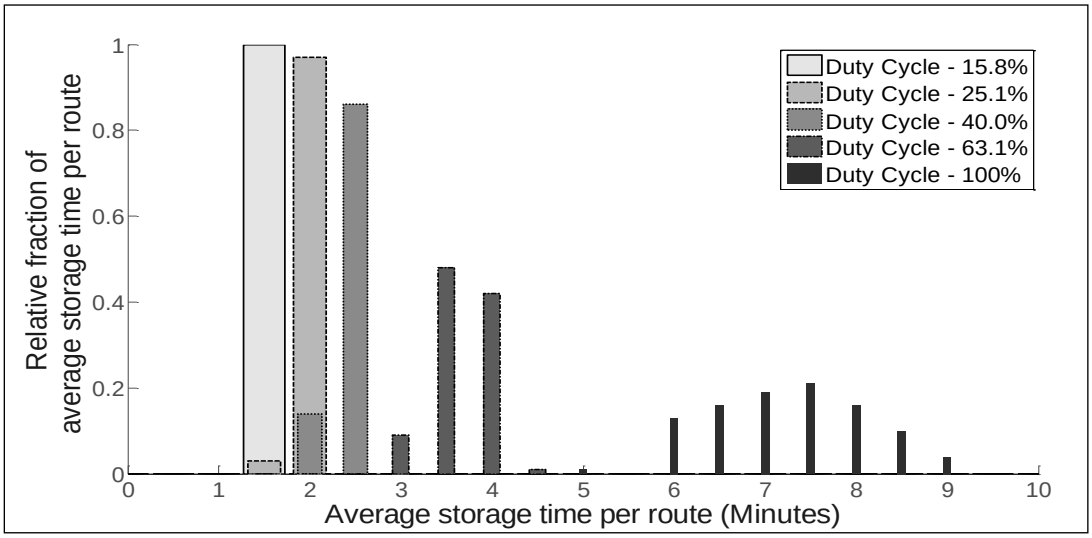


Figure 5.18. Buffer storage time distributions vs harvesting efficiency (Duty Cycle)

F. Buffer Storage Time Distributions

In Section 4.1.3, we have discussed how a Binary Event Buffer has been incorporated per node within the Energy-Aware Pulse Routing scheme in order to reduce event losses due to energy scarcity. The latter can effectively translate into reduction in overall event reporting delay because the reporting of the event does not need to rely on subsequent repeat transmissions.

In Figure 5.18, the distributions of the per-hop event buffering delay (averaged over all hops on an end-to-end route) have been demonstrated. As is evident from the distribution graphs, the peak moves towards the lower end of the storage times when the vibration DC increases. Thus, when the energy availability is scarce (lower DC), the percentage of larger buffer storage times is

higher than when energy is gratuitous (higher DC). This is effectively a consequence of the energy-awareness in the pulse protocol coming into play.

With energy at a premium, the protocol employs longer storage times to accrue enough energy for transmission of pulses needed for event forwarding. As energy becomes more abundant, the need for buffering is reduced and the per hop storage times gradually approach the lower limit of 1 frame delay which in this case corresponds to a time of 1.5 minutes as shown in Figure 5.18. Thus, the observations validate the event buffering process and its proposed impacts.

5.3. Summary

In this chapter, we have developed an energy-aware pulse networking protocol and show the advantages of using the energy-aware version over the baseline version of Pulse Networking in a vibration energy harvesting setting. We also provide evaluation of the energy-aware pulse protocol performance in a variety of harvesting scenarios which consider spatiotemporal variation of the energy harvesting availability. We conclude that the energy-aware protocol provides better performance over the baseline version as demonstrated in a wide variety of harvesting scenarios.

In the next chapter, we will develop a Structural Health Monitoring framework utilizing Energy-Aware Pulse Networking in a piezo-electric transducer-based through substrate communication scenario. We will also evaluate the performance of the developed protocol framework using realistic vibration traces from an airplane wing structure.

CHAPTER 6: THROUGH-SUBSTRATE ULTRASONIC PULSE NETWORKING

In this chapter, we develop a through-substrate ultrasonic communication framework utilizing the energy-efficient discrete-pulse-based networking architecture detailed in the previous chapters. We also evaluate the efficacy of this networking architecture in a realistic Structural Health Monitoring (SHM) setting, namely event monitoring on an aircraft stabilizer structure. For the evaluation process, first, a realistic acceleration profile across an airplane stabilizer is developed using dynamic response simulation based on Finite Element Modeling. Thereafter, a simulated model of harvested energy is obtained from the spatiotemporal vibration profiles on the stabilizer. Finally, Energy-Aware Pulse Networking (as detailed in Chapter 5) simulation is performed on an array of nodes distributed over the stabilizer. In this process, we also consider pulse networking time frame adaptations relevant for ultrasonic through-substrate operation. Using this evaluation framework, we demonstrate the performance advantages of the Energy-Aware Pulse Protocol in such an SHM scenario.

6.1. System Architecture

The high-level system architecture envisioned for the representative SHM application of airplane stabilizer monitoring is shown in Figure 6.1. The broad idea is to use a through-substrate sensor network as shown in the figure, where a collection of Through-Substrate Ultrasonic Pulse Networking (TUPN) units are mounted / embedded on the structure being monitored. The example in Figure 6.1 (a) and 6.1 (b) demonstrates how an event transportation network can be formed on an airplane stabilizer through the stabilizer substrate itself. Each TUPN is comprised of a piezo-electric transducer and a low-power computing module involved in sensing and networking control as shown in Figure 6.1 (e). In Figure 1(e), the computing module is a prototype, while in Figure

6.1 (f), we show a more recent work-in-progress integrated circuit (IC) chip with a smaller footprint both in size and in energy usage.

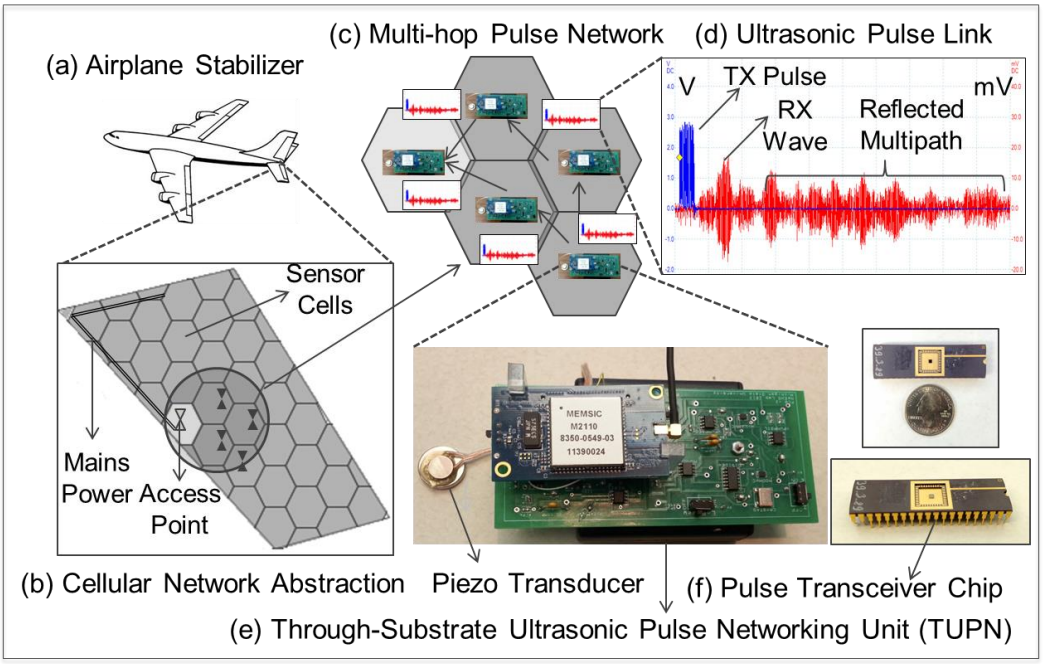


Figure 6.1. Event Monitoring using a Through-Substrate Sensor Network

The advantage of using a piezo-electric transducer in this application is three-fold. First, the transducer can be used for sensing purposes because it reacts to vibrations and can be used to identify vibration patterns that indicate structural anomalies. Secondly, the piezo transducer can be used to harvest energy from the ambient structural vibrations which can be used to power the TUPN modules. Finally, the piezoelectric modules can be used for communication through the very substrate that they are embedded on, thereby removing the need for any retrofitted components for communication such as radio transceivers. As shown in Figure 6.1 (c), neighboring TUPNs can form short ultrasonic communication links through the substrate (e.g., aluminum alloy or composite). A TUPN detected event (i.e., strain, fatigue etc.) results in an ultrasonic pulse, which is transported multi-hop to a data logger or sink node in such a manner that the latter can

indicate: 1) the very occurrence of the corresponding event, and 2) its location of origin with a pre-defined resolution. Resolution is based on a cellular abstraction (Figure 6.1 (b)) in which the TUPNs are addressed not individually but based on the IDs of the cells in which they reside. Even with such limited information, several application level conclusions can be derived at the sink by correlating multiple event pulses [39].

6.2. Prototype Ultrasonic Transceiver and Link Characterization

A prototype TUPN, as shown in Figure 6.1 (e), was developed and used for characterizing pulse-based ultrasonic data links (Figure 6.1 (d)) through metal substrates. Each TUPN can both transmit and receive using an ultrasonic piezoelectric attached. The Pulse Loss Rate (PLR) and False Positive Pulse Rate (FPPR) for communication using the above-mentioned transceiver through a 2024 Aluminum alloy plate (substrate) is listed in Table 6.1. We consider the ultrasonic link performance over a variety of distances and two different source voltages. It is to be noted that the choice of substrate material (Al 2024) was motivated by the fact that this is the most prevalent material used in aircraft stabilizer construction [90]. The two source voltage levels (i.e. 3V, 6V) were chosen keeping in mind the different voltage requirements for transmission in two network roles – source (forwarding) and sink (synchronization) as will be discussed in Chapter 5 for the pulse networking protocol.

As shown in Table 6.1, for the distances and voltage levels considered in the experiments, the ultrasonic signal propagation was found to be reliable with PLR/FPPR ranges in the order of 10^{-6}. It is to be noted that the architecture is specifically designed for short links and an overall small network size. Hence, the choice of distance less than 1m for 3V source voltage experiments (pulse forwarding links) and up to 4m for 6V source voltage experiments (frame synchronization

pulse links). At such distances, the ultrasonic SNR levels and signal shape are good enough for efficient and robust signal reconstruction at the receiver.

Pulse Link Length (meters)	Pulse Voltage (80 µs pulses)	Pulse Loss Rate (PLR)	False Positive Pulse Rate (FPPR)
0.5	3	$2.16X10^{-6}$	$4.02X10^{-6}$
0.75	3	$2.89X10^{-6}$	$5.77X10^{-6}$
1.00	3	$3.02X10^{-6}$	$7.21X10^{-6}$
4.00	6	$1.62X10^{-6}$	$3.4X10^{-6}$

Table 6.1. PLR/FPPR for ultrasonic communication over Al 2024 alloy plate using prototype TUPN module

6.3. Application and Network Model

6.3.1. Application Model

TUPN units, as discussed above, are distributed across the structure to be monitored and form an event transportation network. We use an airplane horizontal stabilizer as the target structure in this work. Each TUPN is equipped with piezoelectric sensors for structural event sensing, and piezoelectric transceivers for generating ultrasonic pulses when events are detected in the locality. The TUPNs are powered by energy harvested from structural vibrations using a piezoelectric transducer. The TUPN network is used for transporting local event information to a centralized Access Point / Base Station (Figure 6.1 (b)) where pulses received from multiple such units across the structure are collated to make inferences about the overall health of the structure, which is the stabilizer in our case.

6.3.2. Network Model

A cellular network abstraction is used for organizing the TUPNs distributed across the stabilizer. Localization is accomplished with the resolution of pre-defined regular hexagonal sensor-cells as shown in Figure 6.2. Each cell represents an event area with a unique Cell-ID. Since spatial localization resolution is at the cell level, shrinking the sensor cell size can increase the resolution. Each sensor node (i.e., a TUPN) belongs to one of these event areas (cells) and is pre-programmed with the Cell-ID of its own cell and those of its geographical neighbors. Although the cells in Figure 6.2 are shown to be hexagonal, there are no specific architectural requirements in terms of their symmetry, shape, and size. Due to the cellular abstraction, the sensors are not individually addressed, and therefore no per-sensor addressing is necessary at the MAC or routing layers.

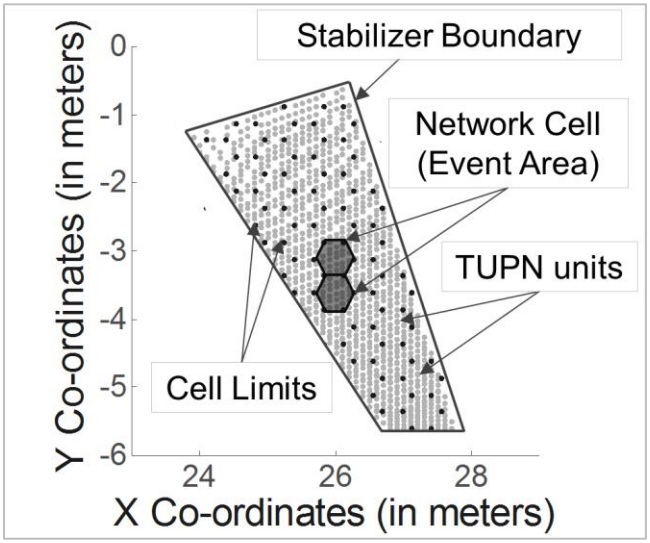


Figure 6.2. Network Model on Airplane Stabilizer Structure

6.3.3. Performance Needs

Event reporting delay from sensor to base station is considered the primary performance index for this architecture. Delay in this architecture depends on the route lengths and energy

availability in different parts of the target structure. As explained later (Section IX), pulse losses on the structure also translate into additional delay.

In traditional networks, the delay typically ranges from few milliseconds to 100s of milliseconds. However, in a through-substrate network that runs on vibration-harvested energy, the expected delay can be much larger - in the range of few minutes to 100s of minutes. Such large delay can be acceptable for niche applications such as aircraft structure monitoring on a per-flight basis. For example, non-critical sensed events (e.g., anomalous stress pattern in certain parts of the structure such as an aircraft wing or stabilizer) during a flight can be reported to an access point while the aircraft is in flight. After landing, the collected data can be downloaded from the access point, thus avoiding the need for a ground-based structure inspection after each flight. For such an application, the reporting delay needs only to be smaller than typical flight duration, which can be up to tens to hundreds of minutes.

6.4. Structural Vibration Model

This section summarizes dynamic response simulation of an aircraft stabilizer for generating spatiotemporal vibration (i.e., acceleration) data. This data is then used for modeling energy harvesting for the target sensor network. The simulations were performed using a finite element method and the model was based on the geometry of the Boeing 737 horizontal stabilizer. Extracted results consisted of acceleration time histories at the TUPN node locations along the stabilizer surface.

Structurally, a typical aircraft stabilizer consists of an internal framework of stringers (for bending resistance), and diaphragms (for shear resistance and load distribution) enclosed by a continuous surface, or skin (for torsional resistance and aerodynamic lift). The lift pressures

generated from the wind flow are primarily controlled by the airfoil shape and the skin; while the structural behavior (i.e., dynamic response) is primarily controlled by the internal framework.

The finite element simulations were conducted with the program Abaqus [91], [92]. The geometry for the model was simplified to represent the main structural components for the stabilizer element and obtain a realistic dynamic response from the simulation. The airfoil geometry of the stabilizer was ignored.

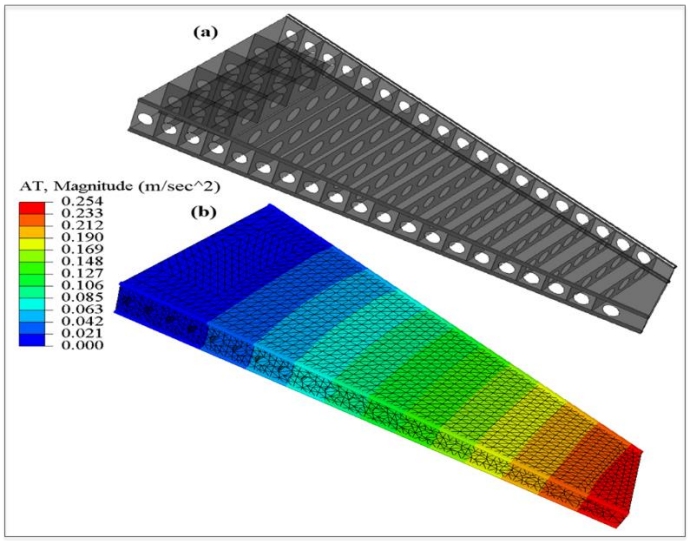


Figure 6.3. (a) 3D model of stabilizer, (b) Accln. of stabilizer based on Finite Element analysis

A view of the model is shown in Figure 6.3 (a). All parts were modeled with continuum-type shell elements with uniform thickness. Additional stiffness from the stringers was modeled by placing spars (beam elements) on the top edges of the leading and back stiffeners. Thickness of the shell elements including the box structure and the stiffeners was 5 mm. This thickness value was determined to obtain realistic dynamic properties due to additional mass from non-structural components in the stabilizer. The material properties assigned to the model were those of Aluminum 2024 [93] with an elastic modulus of 73.1 GPa, a Poisson ratio of 0.33 and density of

2780 kg/m^3 [93]. Figure 6.3 (b) shows the instantaneous acceleration available at different parts of the stabilizer structure at a particular time, in this case the 7600th second after start of simulation.

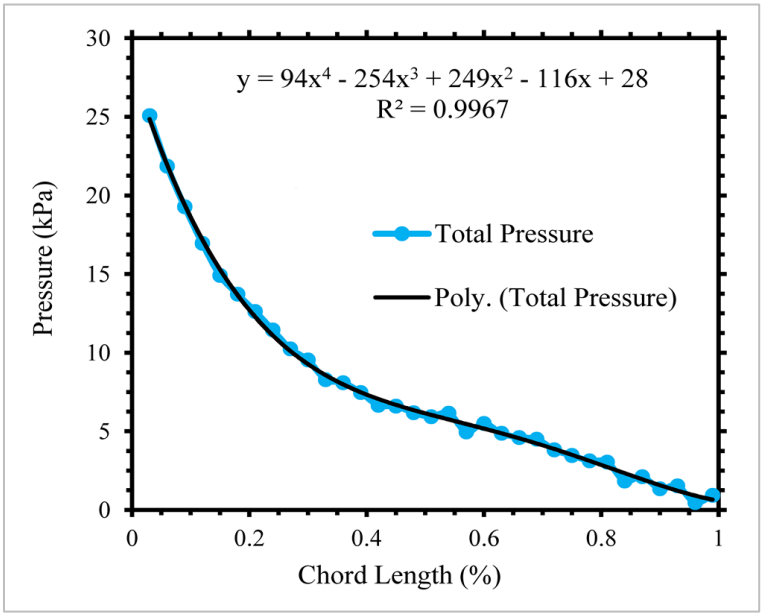


Figure 6.4. Total pressure profile and simplified triangular pressure profile vs. chord length (aircraft speed 800 km/h)

Frequency analyses were conducted to obtain vibration properties and a dynamic analysis was performed to determine spatiotemporal acceleration profiles under simulated cruising conditions for an aircraft. The aero-dynamic loading pressure on the stabilizer was estimated using FoilSim III [94] for a speed of 400 km/s and extrapolated to 800 km/s. The lift pressure distribution across the stabilizer chord length is illustrated in Figure 6.4. The loading pressure was then applied to the model as an incremental ramp with a noise perturbation of 10% over a time domain of 7600 seconds. It should be noted that the simulated demand neglected the rigid-body flight dynamics of the plane and thus the simulation captures only the relative response of the stabilizer element. A surface contour plot of the average acceleration at the bottom surface nodes is shown in Figure

6.5. As expected, nodes at the tip experience greater acceleration on average than the nodes at the mid span of the stabilizer.

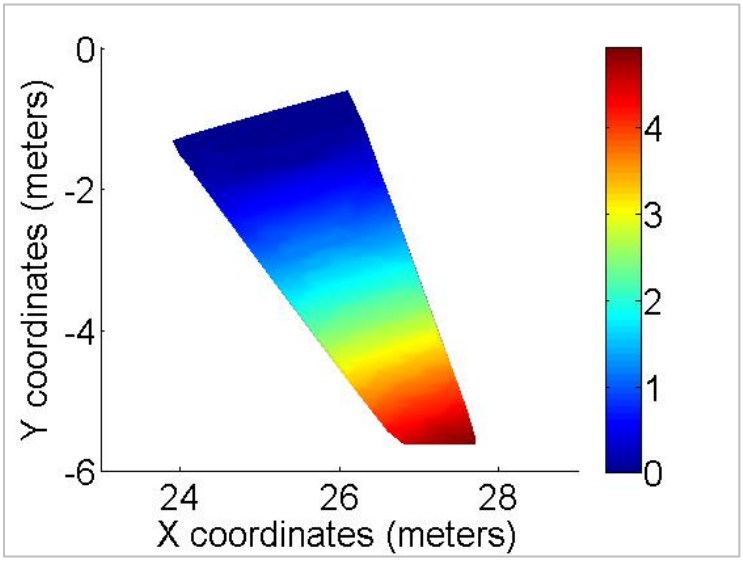


Figure 6.5. Average Acceleration based on node coordinates (at aircraft speed of 800 Km/h)

6.5. Energy Harvesting Model

An energy harvesting simulation model was developed for translating the above spatiotemporal acceleration data into energy generation across the target stabilizer structure. A bridge rectifier based piezoelectric energy harvester circuit model as shown in Figure 6.6 is used. The piezoelectric module on the harvester is used to transform the ambient vibration in the structure to electric power, which is subsequently rectified and stored in an attached super-capacitor for driving the TUPN module operation.

In Figure 6.6, the left part represents the mechanical equivalent circuit for a single degree of freedom piezoelectric energy generator. The resistor R_m represents the damping constant, K_p represents the spring constant, and M represents the equivalent mass – all for the piezoelectric device. The transformer with gain ratio $\theta:1$ is used for modeling the transfer of mechanical energy

to electrical energy. The parameter θ represents the electromechanical coupling constant, which is an inherent physical property of the piezoelectric device material. The parameter x represents the displacement of the piezo generator device (i.e., from its mean position) because of the vibration applied on the structure to which the device is attached. For a device mass of M, the generated force can then be written as Mx'', where x'' is the generated acceleration.

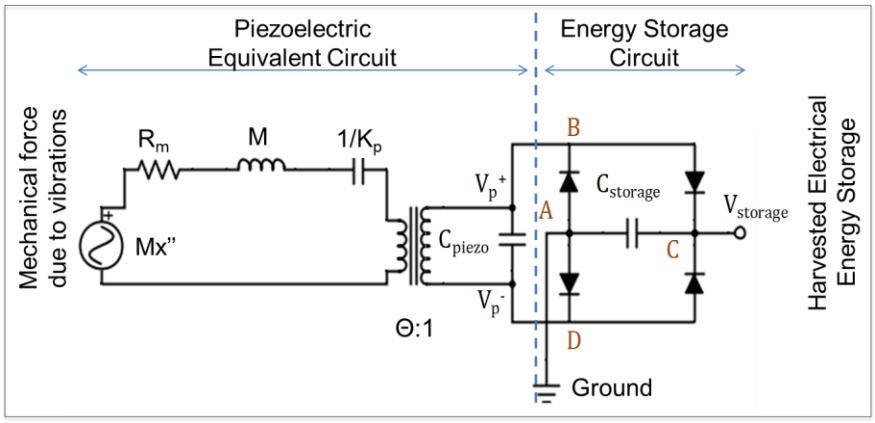


Figure 6.6. Piezoelectric Harvester Circuit Model

The right part of Figure 6.6 represents the equivalent circuit for generated electrical energy regulation and storage. A bridge rectifier is used to rectify and store the generated energy in a storage super-capacitor with capacitance $C_{storage}$. Another capacitor with capacitance C_{piezo} represents the internal capacitance of the piezoelectric device itself. The TUPN sensor nodes are to be connected across the storage capacitor as the load.

$$\frac{dv_0}{dt} = \frac{dx}{dt} = v_1 \quad \ldots (6.1)$$

$$\frac{dv_1}{dt} = -\frac{K_p}{M} * v_0 - \frac{R_m}{M} * v_1 - \frac{\theta}{M} * v_2 + \left[a_1 + \frac{(a_2 - a_1) * (t - t_1)}{(t2 - t1)}\right] \quad \ldots (6.2)$$

$$\frac{dv_2}{dt} = \frac{1}{C_{piezo}} * \left[\theta * v_1 + I_S * \left[e^{\left(\frac{-v_2 - v_3}{2 * nV_T}\right)} - e^{\left(\frac{v_2 - v_3}{2 * nV_T}\right)}\right]\right] \quad \ldots (6.3)$$

$$\frac{dv_3}{dt} = \frac{1}{C_{storage}} * I_S * \left[e^{\left(\frac{-v_2 - v_3}{2*nV_T}\right)} + e^{\left(\frac{v_2 - v_3}{2*nV_T}\right)} - 2 \right] \quad \ldots (6.4)$$

The equations 6.1 – 6.4 as shown above model the circuit in Figure 6.6. The differential equations have been formulated in terms of the four variables v_0, v_1, v_2, v_3. These variables capture the displacement (i.e., x), its first derivative ($\frac{dx}{dt}$), the voltage across the piezo capacitor (i.e. ($V_p^+ - V_p^-$)), and the voltage across the storage super-capacitor (i.e. $V_{storage}$) respectively. The parameters a_1 and a_2 represent the accelerations on the piezo module at consecutive discrete simulation time instants t_1 and t_2 respectively. The quantity $\left(a_1 + \frac{(a_2 - a_1) * (t - t_1)}{t_2 - t_1}\right)$ expresses the interpolated acceleration at any time instant t. The equations have been obtained using the equations of motion [43] of the piezoelectric element and the closed loop circuit equations. By solving these differential equations simultaneously for a specific input acceleration profile (i.e., a_1 and a_2), it is possible to generate the corresponding harvested energy profile $\left(\frac{1}{2} * C_{storage} * v_3^2\right)$. The efficiency of harvesting is sensitive to many parameters including the electromechanical coupling co-efficient (i.e., θ), which is a function of the piezoelectric material, as well as its dimensions. The amount of energy that can be stored in the super capacitor and the rate of energy accumulation depend on the capacitance of the storage capacitor. These parameters and their effects on the network performance during event reporting across the stabilizer structure will be presented in Section 6.7.

6.6. Integrated Evaluation Architecture

An end-to-end software simulation architecture was developed for evaluating network performance in the presence of models for structural vibration and the resulting harvested energy. As shown in the right column of Figure 6.7, a set of sensor nodes (i.e., the TUPNs) are distributed

over a target structure of airplane stabilizer to form a cellular pulse network. Each node executes the energy-aware pulse protocol as outlined in Chapter 5. The left column shows how spatiotemporal acceleration profiles (Section 6.4) are used for generating harvested energy using the model presented in Section 6.5. Based on the specific placement of a TUPN node, each sensor node can make use of a certain time-varying energy profile that is applicable for its specific location on the stabilizer structure. Using their specific time-varying energy input, as they execute the energy-aware pulse protocol, the reserve energy in the storage capacitor (see circuit in Figure 6.6) cycles with time. An example of such cycling for a specific TUPN sensor node is shown in the bottom middle part of Figure 6.7.

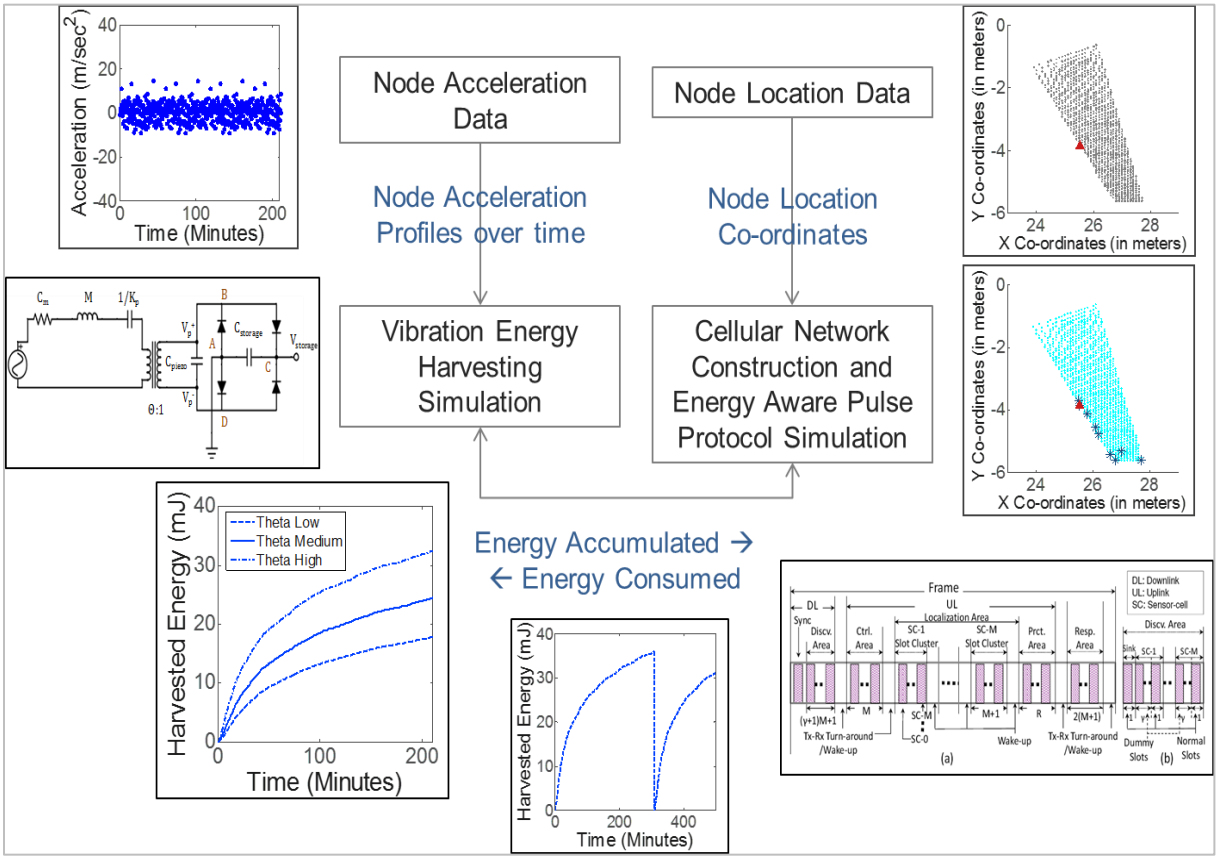


Figure 6.7. Architecture of integrated evaluation software

End-to-end event transportation performance is determined by the structure vibration profile, event generation profile, and the network performance. The integrated evaluation platform integrates all these components within a single software module that was written in C++.

6.7. Performance Results

6.7.1. Network, Energy and Event Generation Model

The simulated network consists of 1057 TUPN sensor nodes evenly distributed on the described airplane stabilizer structure with a sink node placed at the center left corner as shown in Figure 6.8. The nodes are grouped into 56 regular hexagonal cells (as in Figure 6.2) with an average of 3 nodes per cell. The spacing between individual nodes is around 0.3m and the transmission range is kept at 0.75 m. Height of each hexagonal cell is 0.5m with cell side length approximately 0.288 m. The entire simulated structure is approximately 6m x 3m x 0.005m.

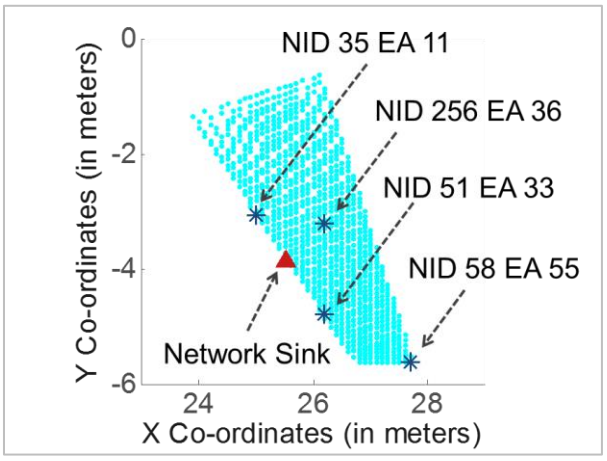


Figure 6.8. Pulse network mapping on a target aircraft stabilizer structure

Energy needed for transmitting a single pulse is chosen to be 1 mJ. This transmission budget has been chosen in line with the estimates for ultrasonic message transmission budgets as reported in [8]. The power consumption for reception as well as idling operation is chosen to be 1

μW see datasheets for ATTiny25/45/85 at [95]–[97]). Higher values for these budgets (as compared to the references) are chosen to demonstrate the worst-case event reporting delays delivered by the proposed architecture.

Event generation locations are uniformly scattered across the stabilizer. Single events are generated at different source locations across the stabilizer and their reporting characteristics are studied. Spatial and temporal energy generation is controlled using the harvested model described in the previous sections.

6.7.2. Network Node Energy Traces

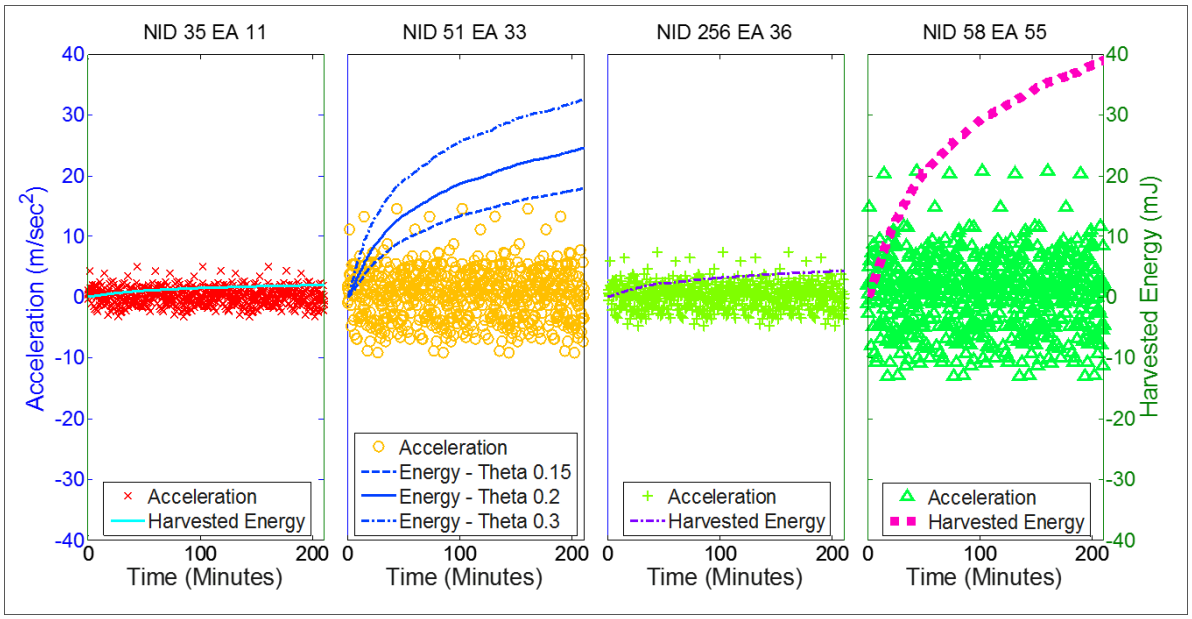


Figure 6.9. Acceleration and harvested energy profile at chosen TUPNs

Figure 6.9 shows the applied acceleration, generated from structure vibration, at four different nodes whose locations on the stabilizer are shown in Figure 6.8. A title {NID *i*, EA *j*} represents the corresponding node-id and event area (cell ID) that the node belongs to. In addition, the figure shows how the harvested energy builds up in each node's storage capacitor in the

absence of any expenditure due to network activities. In particular, for one sample source node NID 51, the figure also demonstrates the variation in energy generation rate as a function of the harvesting efficiency which is indicated by the parameter Θ (Section 6.5).

It should be observed the node situated near the outer tip of the stabilizer structure (i.e., NID 58 EA 55) experiences the highest level of vibration, and hence the fastest harvesting out of all other selected nodes. The node closest to the anchor of the stabilizer (i.e., NID 35 EA 11) experiences the least amount of vibration, and hence the lowest rate of energy build-up. The energy harvesting rates of the other two selected nodes are somewhere in between. The energy build-up for node {NID 51 EA 33} is shown for three different electromechanical coupling constants (i.e., θ) to demonstrate its effects on the harvesting efficiency and its resulting storage capacitor charging rate. The harvesting rate is important since the performance of the event transportation network protocol does depend on such rates across the structure as will be discussed in the following subsections.

6.7.3. Event Reporting Performance

The performance of multi-hop event reporting from a source sensor node (i.e., TUPN) to the sink node is characterized in terms of the reporting delay. Such delay is characterized with varying: 1) structure vibration intensity, 2) per-node storage capacitor size, and 3) electromechanical coupling efficiency of the energy harvesting piezoelectric material.

Vibration intensity is controlled by a scalar acceleration multiplier, which is used for scaling the raw acceleration values generated by the finite element model. The maximum node accelerations are varied in the range 10 m/sec^2 to 20 m/sec^2 (i.e., approximately from g to 2g, where g is the acceleration due to gravity). Such a range for the acceleration values has been chosen

based on data from prior studies with representative unmanned aerial vehicles [98] which indicate that the accelerations range between $\pm 1g$ and $\pm 1.5g$ [98]. Three acceleration multiplier values have been used to create such variation viz. 10, 15 and 20. These correspond to maximum node acceleration values of 10.4652 m/sec^2, 15.6978 m/sec^2 and 20.9304 m/sec^2 respectively. The corresponding average node acceleration values across the stabilizer are 1.0124 m/sec^2, 1.5186 m/sec^2 and 2.0248 m/sec^2 respectively.

Experiments are conducted with different storage capacitor sizes that range from 1 μF to 10 μF to characterize the effects of available storage. The electromechanical coupling co-efficient (i.e. parameter θ in Eqns. I – IV), a physical property of the piezoelectric generator material and its dimensions, is changed in the approximate range of 1.96*10^{-5} Coulomb/meter to 1.96*10^{-4} Coulomb/meter. This is used for varying the available energy-harvesting rate. This maximum value for the co-efficient θ (i.e. 1.96 x 10^{-4}) was calculated using Equation 6.5 [99]

$$\theta = \frac{d_{31}}{\frac{1}{E_{11}}+\frac{1}{E_{12}}} * b * t * \pi * \frac{1}{2*l} \quad \ldots (6.5)$$

Here we assumed a PZT ceramic type piezo material with the following characteristic parameter values [99]

d_{31} = 320x10^{-12} Coulomb/Newton

E_{11} = 5.0x10^{10} Newton/meter^2

E_{12} = 6.2x10^{10} Newton/meter^2

It is to be noted that d_{31}, E_{11} and E_{12} are inherent properties of the piezo material) [99].

The dimensions of transducer were chosen as follows –

Length (l) - 31.7 mm

Breadth (b) - 16 mm

Thickness (t) - 0.0028 mm

Figure 6.10 shows the delivery delay experienced by events (e.g., stress anomaly in the structure) detected at four different event areas of the aircraft stabilizer structure. The exact locations of origin of those events and the sink node are shown in Figure 6.8. The four event areas are subject to different vibration levels (at the same instant and across time) and correspondingly different levels of energy harvesting. Such variations in energy availability in these areas, and along the route to the sink, cause different event reporting delay for the pulse protocol.

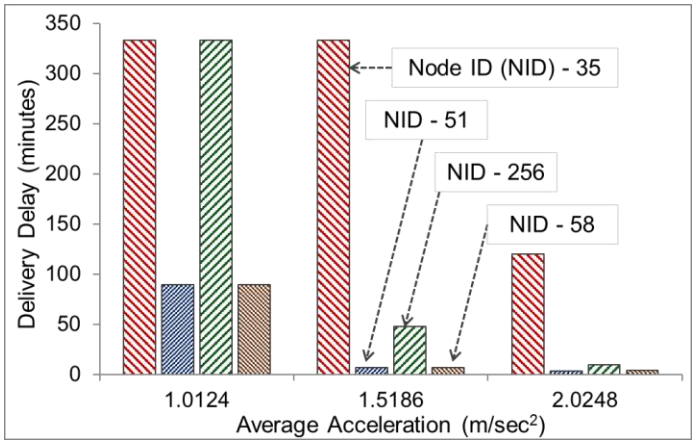


Figure 6.10. Event reporting delay with different vibration intensities

Delays for those four areas are evaluated with three different vibration intensities, which are indicated by the corresponding average acceleration values in Figure 6.10. As expected with higher vibration intensities, the events from all four areas are reported to the sink with lower delays, mainly due to higher energy availability and lower event buffering at the intermediate nodes on

the route. It should also be observed that the relative patterns for the delay experienced by the events from all four areas are similar across all vibration levels.

Figure 6.10 shows that with intense structure vibration, the reporting delay on the target stabilizer structure can be as low as a few minutes. With very low vibration, however, the delay can be as large as hundreds of minutes. High event reporting delays (i.e., minutes) can be acceptable for non-critical monitoring applications as outlined in the Performance Needs subsection in Section 6.3.3.

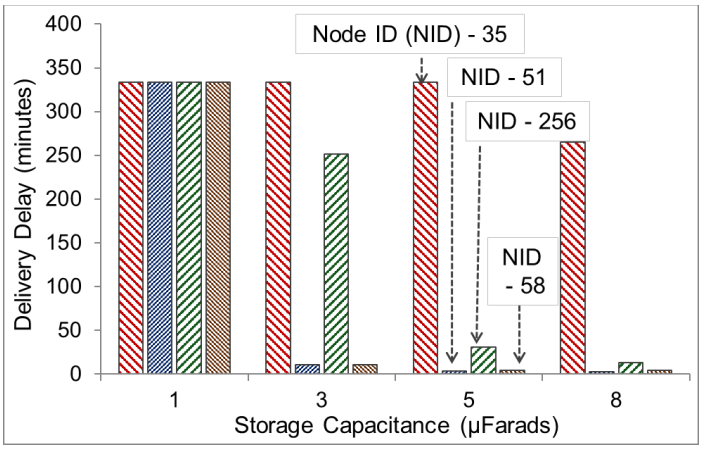


Figure 6.11. Event reporting delay with different energy storage capacity

Figure 6.11 depicts the dependence of event delivery delay on the available storage for harvested energy at individual nodes. As the super capacitor capacitance ($C_{storage}$) (as shown in Figure 6.6) is increased, the net energy availability per node increases in a given time because for the same electrical voltage available, the higher capacitance can store a larger amount of electrical charge and thus energy. Thus, the event-buffering phenomenon (as introduced in the protocol in Chapter 5) occurs less frequently. This helps in reducing the overall end-to-end event reporting delay. This effect is evident in the delay for all four sample sensor nodes shown in Figure 6.11.

The rate of energy harvesting can be also increased by using a piezoelectric material with higher Electromechanical coupling coefficient (i.e., θ in Eqns. 6.1 - 6.4). Figure 6.12 depicts the delay improvement as a function of that parameter. As observed in Figs. 6.10 and 6.11, the relative delay patterns for the sensor nodes 35, 51, 256, and 58 remains the same across all the θ values. In each case, a higher value of Θ ensures faster harvesting and thus lower reporting delays on average.

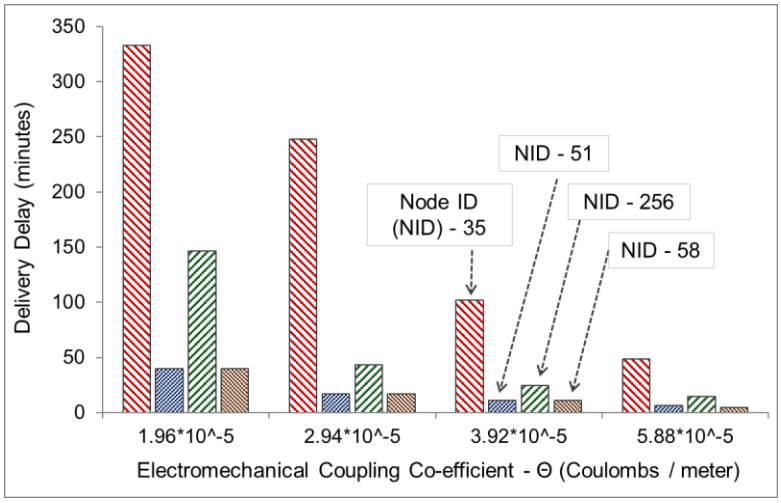


Figure 6.12. Reporting delay for different electromechanical coupling

It is interesting to note that in all the three Figs. 6.10, 6.11 and 6.12, the nodes situated near the tip of the stabilizer and the edge e.g. 58, 55 enjoy less delivery delay compared to nodes situated near the heart of the stabilizer farther from the tip e.g. 256, 35. Such spatial variation in performance is explored in detail in the following results.

Spatial distribution of energy availability and event reporting delay are shown in Figs. 6.13 and 6.14 respectively. The average harvested energy over a 100 second period at different parts of the stabilizer structure is shown in Figure 6.13. The energy values are shown for different electromechanical coupling coefficient (i.e., θ). It can be observed that for all θ, the available

energy is higher at locations farther from the anchor of the stabilizer. As expected, with higher θ, more energy is harvested, causing a more visible gradient from the anchor to the tip.

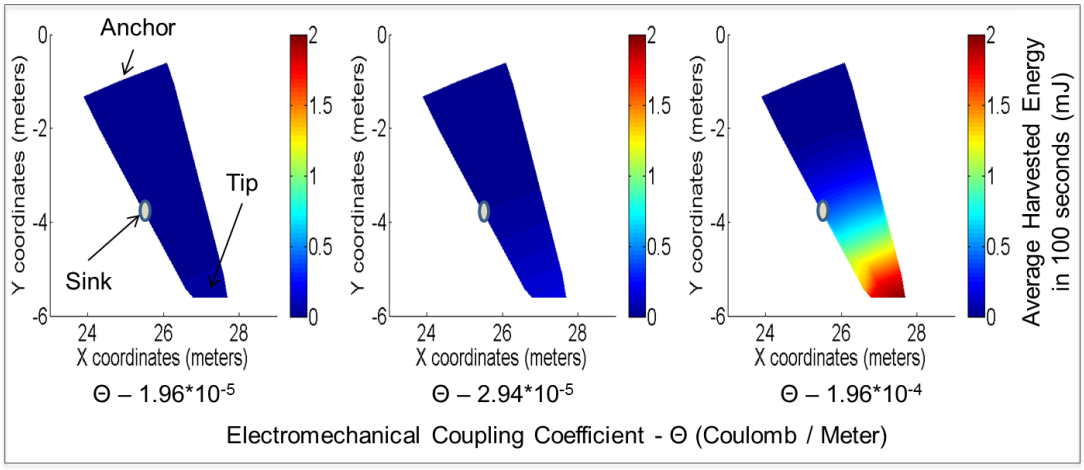


Figure 6.13. Spatial distribution of harvested energy

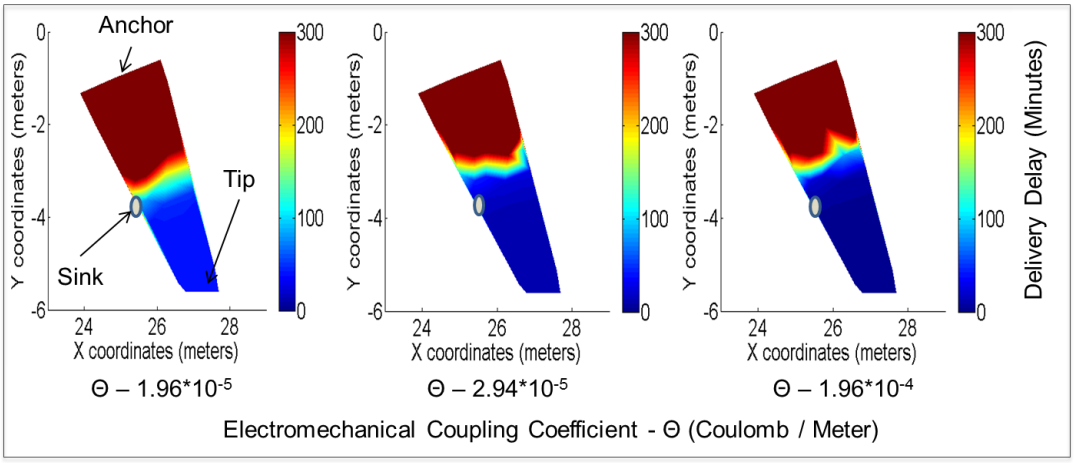


Figure 6.14. Spatial distribution of event reporting delay

Figure 6.14 depicts event reporting delay in the presence of the energy profile shown in Figure 6.13. The delay value in the temperature map at a given location represents the delay experienced by an event that is generated in that specific location on the stabilizer. The dark brown color near the anchor represents high delivery delay due to very low energy availability in that region. The blue color near the stabilizer tip indicates lower delay, owing to higher energy

availability in those regions as shown in Figure 6.13. It can be observed that with higher electro-mechanical coupling constant θ, the overall delay reduces across the whole structure due to higher harvesting efficiency.

A maximum allowable event reporting delay of 300 minutes was set for all the experiments corresponding to the results in Figure 6.14. This maximum allowable delay explains the sharp transition in the delay values near the middle part of the structure. Depending on the specific coupling constant θ, the delays are higher than this maximum allowable value up to a distance from the anchor of the stabilizer. Beyond that distance, the delay starts decreasing, thus causing the multicolor bands somewhere in the middle of the structure. Beyond that band area towards the tip, where the harvested energy is plentiful, the delay becomes very small as indicated by the blue color. It is to be noticed that the band moved towards the anchor with increasing energy availability caused by higher θ values.

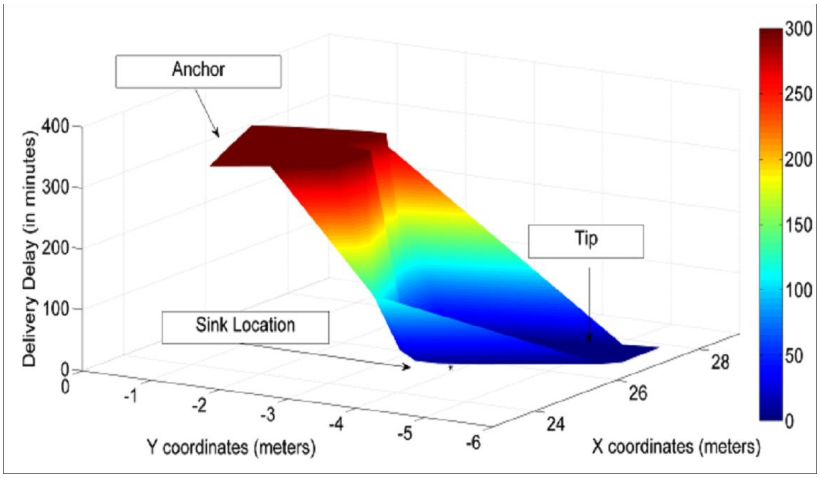


Figure 6.15. A lateral perspective of the delivery delay temperature map ($\theta = 1.96*10^{-4}$)

To better understand the delay bands observed in the top view in Figure 6.14, we present a lateral view of the spatial delay profile in Figure 6.15, when θ is set to $1.96*10^{-4}$. It can be seen in Figure 6.15 that the delivery delay falls quickly as the event source is moved toward the tip of the

stabilizer. This sudden fall in the delay causes the multi-color delay band to be quite narrow in the top-view temperature maps in Figure 6.14.

6.7.4. Impacts of Adaptive Route Diversity

As an energy-aware routing syntax, *adaptive route diversity* is explored for delay reduction, especially when extra energy is available. Figure 6.16 depicts the effects of available energy on the effective route diversity for node (NID 58 EA 55), which is near the tip of the stabilizer. The figure shows that for the largest coupling constant (i.e., $1.96*10^{-4}$), when the harvested energy along the stabilizer is the highest, a large fraction of the transmissions uses a route diversity of 2. This is an attempt to reduce the event delivery delay by exploiting alternate routes, even though they consume higher energy which is abundant in this scenario.

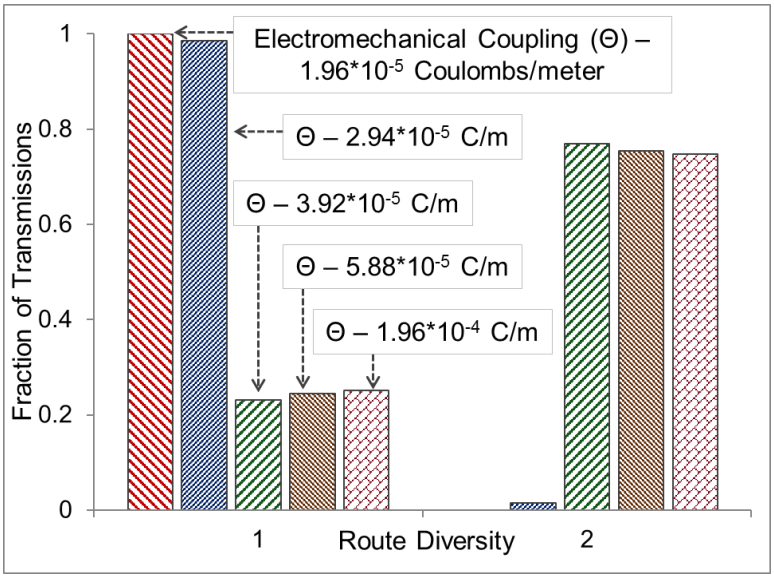


Figure 6.16. Transmission route diversity with varying coupling constant

The trend is reversed as smaller coupling constant values are chosen. For example, when θ is $1.96*10^{-5}$, almost all the transmissions refrain from using multiple diversity (i.e., they use route

diversity of one). Note that for all experiments corresponding to Figure 6.16, a maximum allowable route diversity of 2 was chosen.

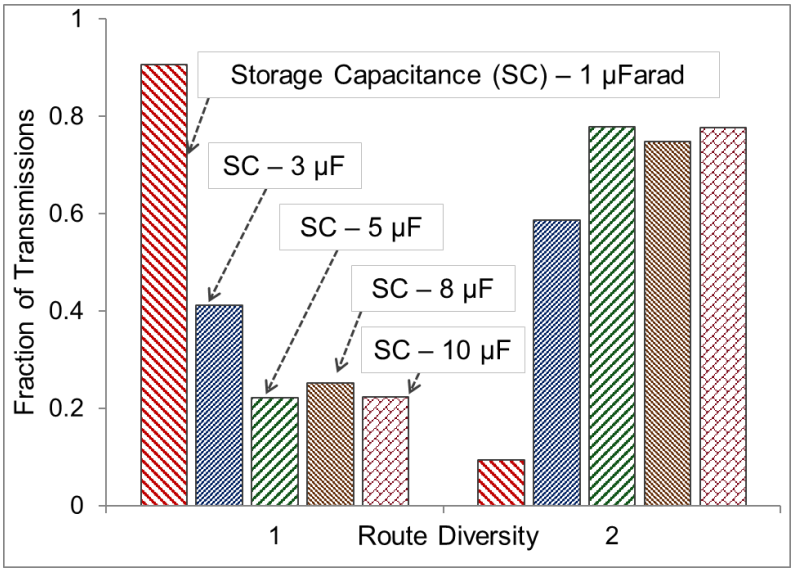


Figure 6.17. Route diversity with varying storage capacitance

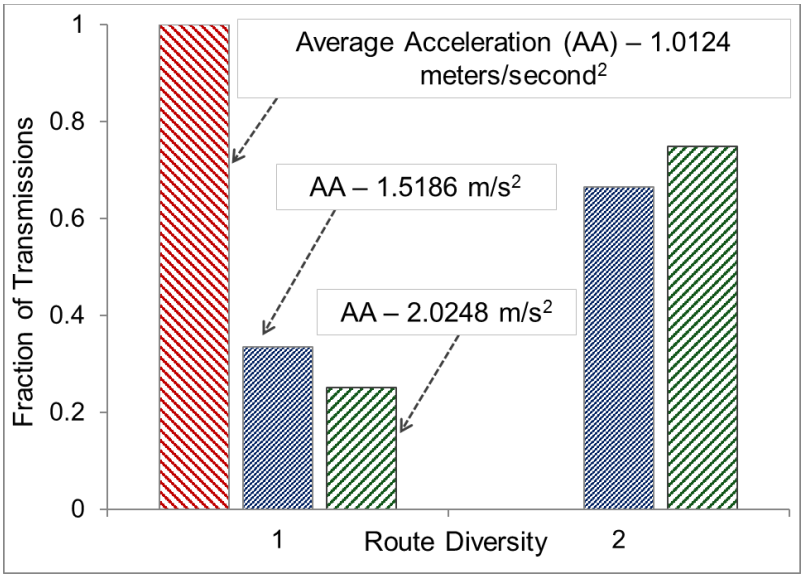


Figure 6.18. Route diversity for different average acceleration levels

The impacts of other energy-availability factors, namely, super capacitor storage capacitance and structure vibration intensity (i.e., the maximum acceleration) on route diversity

are shown in Figs. 6.17 and 6.18 respectively. The same trends (i.e., as observed in Figure 6.16) of high effective route diversities for high-energy situations can be observed in both Figs. 6.17 and 6.18. High storage capacitance and large acceleration values create such high-energy situations.

6.7.5. Impacts of Error

In the absence of pulse loss, the energy-aware routing architecture ensures guaranteed event reporting even when the delay is increased due to energy unavailability along the route. However, when pulses are lost due to noise and detection errors, such guarantees cannot be made. To address that, a repetitive pulse generation model is adopted in which upon event detection, the source node starts generating pulses with a fixed periodicity (e.g., two frames) for indefinite period. This way, even when a pulse is lost, one of the subsequent pulses from the same event eventually gets delivered to the sink, causing an effective event delivery. The tradeoff of this pulse generation model is that the temporal resolution of event reporting is lost. Event reporting delay under this model depends on the pulse generation periodicity and the pulse loss rate (PLR).

Event reporting delays from two different sensors to the sink on the aircraft stabilizer structure is shown in Figure 6.19. In both scenarios, the acceleration multiplier was set to 20 (i.e. an average acceleration of 2.0248 m/sec^2), and the energy storage capacity at the super-capacitor was set to 10 μF. Delay for two different values of the electromechanical coupling co-efficient (i.e., θ) are shown in the figure.

The first observation is that with increasing rates of pulse loss, the delay increases or stays constant. This is because after a pulse is lost, the event can only be reported when a later pulse corresponding to the same event makes its way to the sink node. In some cases, such as for node 256 (where vibration intensities are low), even with an increase in PLR, the delivery delay does

not increase because the event buffering effect due to energy constraint overshadows the effect of delay due to pulse losses.

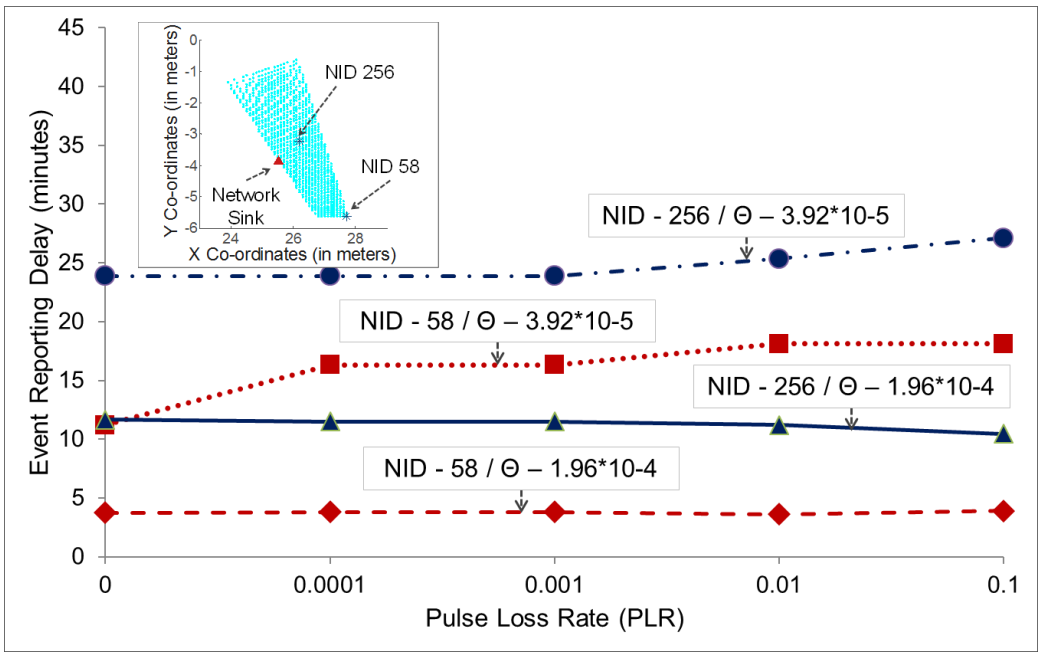


Figure 6.19. Impacts of pulse loss on event reporting delay

The second observation is that for a given θ, the delay for events from the sensor NID 58 i.e., on tip of the stabilizer is lower than that for sensor NID 256, which is near the center of the stabilizer. This difference is mainly due to the highest energy availability near the tip, as demonstrated in Figure 6.13. Finally, for a given sensor, higher θ caused lower delay due to higher available harvested energy. Overall, the simulation results demonstrate that pulse losses affect the event reporting delay, which increases with an increase in PLR, but such increase can be minimal when there is already a significant amount of buffering delay in the system due to energy constraints.

6.8. Summary

In this chapter of the thesis, we developed and evaluated a discrete-pulse-communication-based sensor network architecture that uses through-substrate ultrasonic links and is powered by energy harvested from ambient structural vibrations. An airplane stabilizer is used as the target structure for both monitoring and harvesting purposes. Using an integrated simulator, it was first shown that in the presence of vibration energy harvesting, reliable network performance in terms of event reporting delay could be accomplished by employing the proposed energy-aware protocol syntaxes. Simulation results are then used for analyzing the network performance sensitivity to key system parameters, namely, structural vibration intensity, energy harvesting efficiency of the used piezoelectric material, the energy storage capacity at the pulse switching sensor nodes, and pulse losses caused by ambient vibration noise present in the structure.

In the following chapter, we develop a Spiking-Neuron-based learning architecture for event sequence pattern detection purposes particularly suited to pulse communication systems. We will also demonstrate how this can be combined with the low-power event sensing and networking architecture developed in Chapters 4, 5, 6 to create a holistic low-power framework for event monitoring and detection for Structural Health Monitoring applications. We will support this design using evaluation results showing how the architecture can be effective for low-power detection.

CHAPTER 7: DISTRIBUTED COGNITION USING NETWORKED PULSES AND SPIKING NEURONS

The need for identification of spatiotemporal patterns in distributed event occurrence sequences turns out to be a recurring theme in various popular application scenarios such as object tracking [14], intruder detection [45], structural health monitoring [39], and environment monitoring [46]. Wireless sensor networks are well-suited to such applications for seamlessly capturing the event occurrence information and sending to a centralized base station to facilitate pattern detection and inference. In-situ detection at local cluster-heads in the network, when possible, would be more energy-efficient and facilitate faster response, instead of transporting the data all the way to the Base Station. However, sensing nodes are generally energy-constrained and thus in-situ detection is less practical. In this chapter, we demonstrate that using a single-layer Spiking Neuron architecture, we can efficiently detect the occurrence of such pre-defined event occurrence sequence patterns and with much less energy compared to traditional neural network-based approaches. This opens the possibility of in-place detection of such pattern sequences leading to less networking costs and quicker response. The proposed architecture can also easily be interfaced with various energy-efficient pulse networking architectures like those described in Chapters 4, 5, and 6 of this dissertation. It can thus be part of a complete discrete pulse-based sensing, networking and detection architecture aimed at optimal resource utilization. The latter can be very useful when employed in applications with small and inexpensive sensing devices or those powered by scarce energy harvesting and/or equipped with small batteries. Despite the simplicity of the architecture and inherent energy-efficiency, we demonstrate that high detection accuracy can be achieved, and the approach is well-generalizable over a reasonable range of event interval variation. The detection results are robust enough to handle practical amounts of pulse drift errors, and detection performance can be extended to cover pulse loss and false positive error scenarios

as well using appropriate modifications to the training process. In the current chapter, we provide details of the architecture, implementation and learning parameters as well as training methodology and the corresponding rationale for their choice to make the case for use of Spiking Neurons in energy-efficient distributed cognition applications.

7.1. Introduction

Applications in diverse domains such as habitat monitoring [46], target tracking [14], industrial process control [15], structural health monitoring [39] etc. are known to require the identification / classification of spatiotemporal event occurrence patterns using distributed sensor measurements in order to make higher level inferences based on the same. Wireless sensor networks (WSNs) [47] often provide a seamless way to implement such solutions by facilitating low-cost distributed sensing and communication among sensing nodes. Application instances could range from moving object identification [14], intrusion detection [45] to environmental change [6, 7], structural anomaly detection [39] etc.

A key requirement in such applications is that the sensing architecture be flexible enough to be able to identify a variety of event occurrence patterns i.e. if new patterns need to be detected, the architecture should be able to adapt to the new scenarios. Another aspect is the need for generalizability over a range of similar inputs i.e. detection robustness to minor changes in the same event pattern. Applications in the field usually also assume some amount of energy-efficient / energy-aware operation because many modern distributed sensing architectures for such, aim to create cheap and maintenance-free operation by relying on small sensing devices with limited energy storage but theoretically infinite energy generation capacity (harvesting from environmental sources) albeit at low / erratic rates (harvesting source unpredictability).

Neural network-based learning approaches have been shown to offer a good degree of accuracy when identifying / classifying spatiotemporal patterns in general and with a good degree of generalization robustness which can guarantee suitable operation in the field where conditions are highly dynamic even though underlying patterns might remain the same. However, perceptron-based neural networks [100] (also referred to as second-generation neural networks) which are the workhorse of such applications, generally consume a considerable amount of energy in terms of computation because good recognition is associated with the use of extensive number of neurons. Learning systems like the actual human brain, however, are known to be much more efficient in terms of energy consumption for such computational tasks. This has motivated the development of the third generation of neural networks also referred to as Spiking Neural Networks which try to recreate the mammalian brain neuron functionality more faithfully than perceptron-based approaches. Spiking Neurons have been demonstrated to offer significantly higher computational capacity per neuron [80] and can thus, handle larger computational tasks at a lower energy budget due to lesser number of neurons being employed. This is mainly because Spiking Neurons can recognize inputs in the form of spike trains instead of mere values as in the perceptron-based networks. Encoding inputs in spike trains opens a much larger set of possibilities in terms of temporal coding, rank coding etc. which the spiking neurons can identify if suitably trained, all of which would be much more energy expensive to build / support using a perceptron-based architecture. In terms of hardware implementation as well, Spiking Neuron approaches can be more energy-optimized because they can be implemented using simple highly energy-optimized analog circuitry, eschewing the need for complex digital circuitry and the associated energy burdens for perceptron-based approaches.

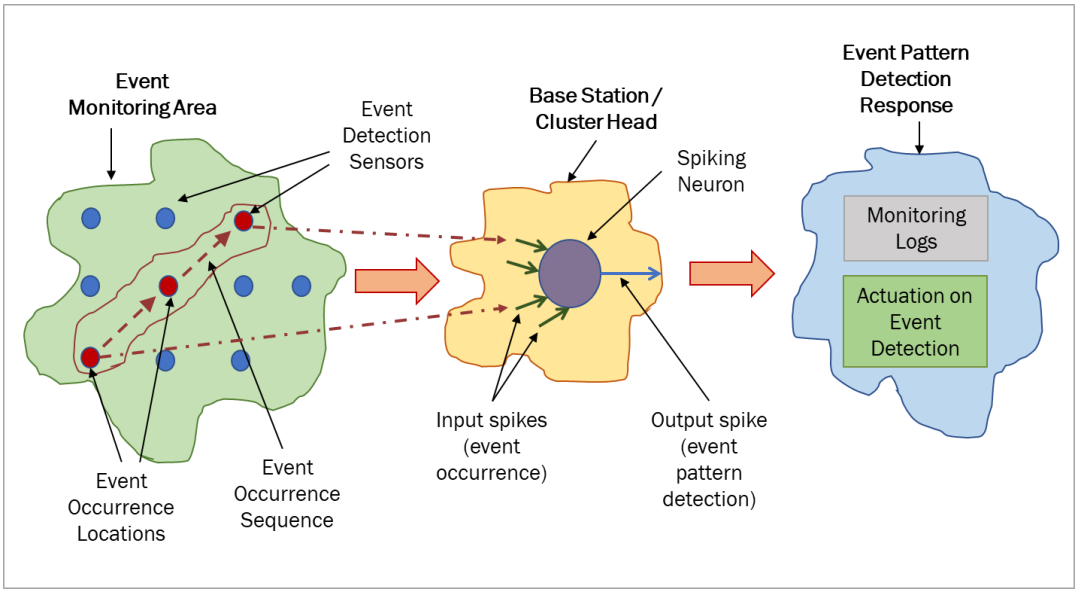


Figure 7.1. Application Overview

In the current work, we demonstrate the use of a Spiking Neuron based architecture and associated learning / training mechanism, based on a learning model called the Tempotron rule [80], to design an energy-efficient spatiotemporal pattern classification architecture for distributed networked sensing applications. We envision a scenario as demonstrated in Figure 1, where a distributed set of event occurrences across a sensor network are available to a central node such as a Base station or cluster-head. The latter has a pre-trained Spiking Neuron which can look at the sequence of event occurrences and detect a known pattern to indicate cognition of an environmental situation that needs to be acted upon or logged for future analysis. As discussed already, Spiking Neurons support extreme energy-efficient operation as well as good detection robustness. Further, because Spiking neurons accept spike trains as inputs, we can adapt them for use with various discrete pulse-based networking approaches as described in [101] and [102], which further helps energy-efficient operation. It has been shown in prior work [35] that discrete pulse-based communication approaches can be significantly more energy-efficient compared to traditional packet-based data communication approaches when the data content is small and

delivery delay constraints are relaxed. This is quite the case for the class of applications we mentioned earlier and, in particular, structural health monitoring, because the only information to be transferred are the time of event occurrence and its location (i.e. data content is small) and the event occurrence intervals are in most cases far larger than the data communication delays e.g. a person walking between sensors placed about 10 meters from each other can trigger events in intervals of about 4 – 8 seconds (assuming normal walking speeds) while communication of such event information from source to sink can generally be achieved in much less time than that, often in the order of milliseconds, thus allowing event information delivery latencies much larger than usual in communication networks.

The primary objectives of the work presented in the current chapter are as follows -

1) To develop an ultra-low-energy architectural solution for detection of spatial-temporal progression of events in WSNs,

2) To demonstrate and evaluate the architecture's ability / constraints in detecting a variety of spatiotemporal patterns of event occurrence across a chosen terrain (as in target trajectory detection, environmental change detection, structural anomaly detection applications etc.) using event occurrence data transmitted from sensing nodes dispersed across the terrain. Such event data will be available only as discrete low-energy asynchronous spikes with event source id information encoded,

3) Develop reasonable learning mechanisms and training approaches such that the architecture can recognize a large variety of patterns as well as ensure robustness to reasonable pattern jitter.

The key contributions are as follows –

a) Mapping spiking neuron-based pattern detection (Tempotron learning rule) to a pulse-based wireless networking context,

b) Use of a Spiking Neuron to learn unique temporal characteristics of expected pattern(s)

c) Use of pulse networking schemes (Pulse Position Coded Pulses / Pulse Time Encoded Networking) for low-energy transmission of event occurrence information from source nodes to a central sink (single-hop / multi-hop) within constraints necessary for proper spiking neuron-based detection.

In the following section, we first discuss the general application and network models considered in this work before going into the details of the learning and networking architecture and demonstrating performance / evaluation results.

7.2. System Architecture

Before we delve into the specifics of the Spiking Neuron Based learning approach and the discrete-pulse based coding architecture, it is reasonable to discuss in some detail about the nature of applications where this architecture would be best-suited. We envision applications where the need is to infer some higher-level conclusions based on a distributed set of sensor measurements across a terrain. For example, we can imagine a geographical area of chosen size (referred to as *Sensing Area*) with sensors deployed uniformly throughout at regular intervals as shown in the green shaded area to the left of Figure 7.2. Such sensors might be of diverse nature e.g. shadow detection [103], sound detection [104], temperature detection [105], structural anomaly detection [106] etc. The important notion here is that the nature of sensing is not as important as the fact that an event can be detected e.g. threshold crossing of the sensed value. The more important task here is to understand the order in which the events were detected at different sensors across the Sensing

Area and how far apart. This can give us clues to understanding several higher-level events e.g. if a person has been moving along a specific trajectory multiple times, if an intruder has moved into a restricted zone within the terrain, if a forest fire is spreading too far, too fast, if a crack is developing over a particular area of a structure etc. Other applications can include detection of toxic gas diffusion in a controlled environment for appropriate safety measures, trajectory detection of a customer in a retail store to understand retail shopping preferences, stealthy motion detection across a terrain indicating an intruder etc.

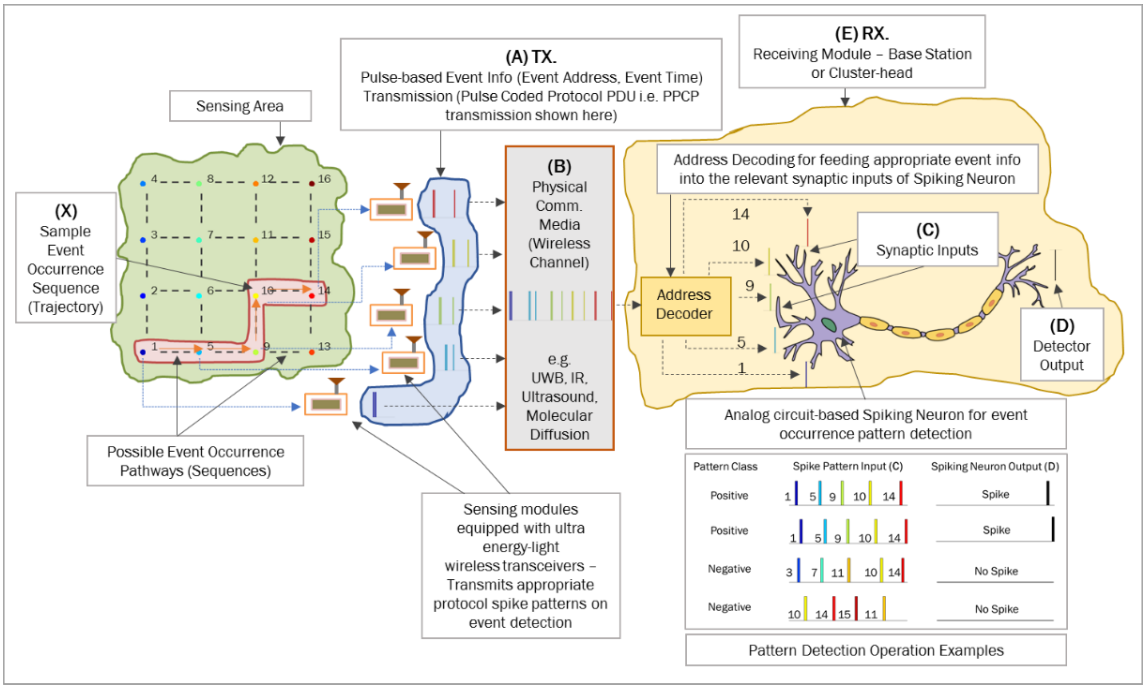


Figure 7.2. Spiking Neuron Learning in Pulse Communication Networks

In the architecture discussed here, each occurrence of a notable event will trigger generation of a spike train at the corresponding time. The spike train can be composed of one or multiple spikes depending on the transmission protocol choice and should incorporate the most important information about the event occurrence such as event location as well as other information such as granularity of detection etc. as relevant. Figure 7.2 shows such spike trains

being generated across multiple sensors (see (A) in Figure 7.2 (shaded in blue)) in response to an event occurrence trajectory (see (X) in Figure 7.2 shaded in red). Such spike trains from multiple sensing devices will travel across the communicating media (see (B) in Figure 7.2 shaded in orange) before they are collated at central locations such as Base Stations or cluster-heads (see (E) in Figure 7.2 shaded in yellow). Here, an appropriate address decoder is used to extract the event source id information and the relevant spike trains are fed to the correct inputs of a prior-trained Spiking Neuron based on where they were generated. The Spiking Neuron would be able to analyze the input spike trains and indicate detection of one or more event sequence patterns with suitable robustness. The neuronal synaptic inputs (see (C) in Figure 7.2) receive spikes from different synaptic inputs (shown in different colors) and based on these, the Spiking Neuron either generates an output spike or not (see (D) in Figure 7.2) to indicate detection of positive or negative patterns respectively. The panel on the lower right of Figure 7.2 shows the neuron's inputs and outputs for different positive and negative patterns. It is to be noted that there is a baseline positive pattern and all patterns, which are slightly jittered versions of the baseline, would be regarded by the Spiking Neuron as positive patterns. Negative patterns can have totally different sequence of synaptic inputs or vastly large jitter levels even if like the baseline pattern in terms of component inputs and this is indicated by the lack of an output spike generation by the Spiking Neuron.

7.2.1. Network Model

It is prudent to discuss some of the assumed characteristics of the network model that would be best suited to the event pattern detection architecture that we discuss in this chapter. The sensing nodes will be equipped with low energy networking functions which can be used to transmit appropriate spike trains when sensing events are detected.

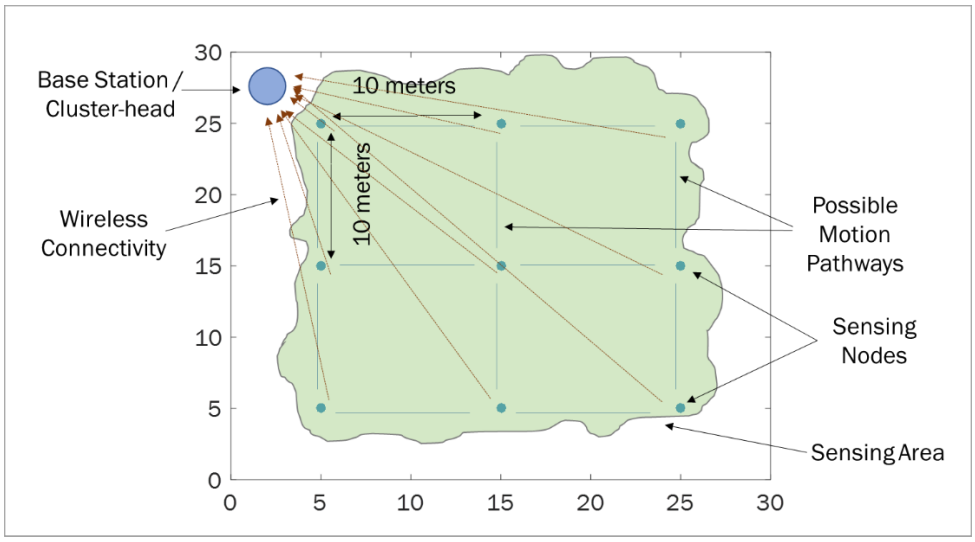


Figure 7.3. Network Topology

In Figure 7.3, we show a representative network of sensors in a square topology as is used for our simulation results discussed in Section 7.6. The nodes have Manhattan connectivity i.e. each node can transmit only to neighboring nodes located within a grid distance of 10 meters. The nodes can encode event location information in spike train form and, at the receiving sink such information can be decoded correctly and efficiently to recreate the approximate event occurrence pattern. This can then be fed into the Spiking Neuron for detection of known sequences. The need for encoding individual node event information in spike trains arises because the nodes need to function in various multi-access channel scenarios where simultaneous transmissions (due to simultaneous event occurrences) would cause collisions and garbled transmissions. The other motivation is to reduce energy cost of transmissions. As will be described in the pulse-based communication protocols in Section 7.4, certain time-domain techniques can be used to ensure communication with minimal collision while incorporating the essential event information in low-energy pulse trains. It is to be noted that such encoding and decoding is not necessarily error-free and can introduce drift errors into the recreated event pattern sequence that is input to the Spiking

Neuron at the Base station. The choice of Spiking Neuron as the learning architecture is particularly in order to protect against such jitter scenarios and still produce robust detection results. In the current work, we also consider only one-hop connectivity between source and sink as shown in Figure 7.3. However, the detection architecture and associated networking approach proposed in this chapter can also be extended to multi-hop networks.

7.3. Spiking Neuron Based Learning

7.3.1. Key Concepts

The primary motivation for the use of a Spiking Neuron-based learning scheme for the kind of applications discussed in Section 7.1 is the architecture's energy efficiency [80] and easy amenability to synchronous / asynchronous spike-based input patterns (which is how the energy-efficiently transmitted data from sensing nodes looks like using different pulse communication protocols as discussed in Section 7.4). Spiking Neuron-based learning schemes also provide robustness in terms of reasonable jitter tolerance and the ability to learn a larger diversity of patterns (compared to ordinary perceptron-style learning schemes) with similar number of neurons [80]. Finally, there is no need for manual coding as the pattern detection requirement changes i.e. the spiking neuron just needs to be re-trained with the new expected scenarios instead of requiring to be logically programmed for each unique pattern, thus saving on some circuit implementation costs as the neuron itself can be designed as a simple analog circuit instead of flexible (less energy-efficient) digital design.

7.3.2. Neuron Description and Tempotron Learning Rule

Since the spiking neural architecture is primarily inspired by the human brain, we use the characteristics of neurons in the brain in our architectural neurons as well. The Leaky Integrate

and Fire (LIF) Model [80] is one of the predominant neuron models in the literature which closely maps the operation of neurons in the mammalian brain. The LIF neuron essentially models a neuron as driven by exponentially decaying synaptic currents contributed by several input synapses. The synaptic currents drive the evolution of the neuronal voltage (integration) and when this voltage crosses a threshold, the neuron is said to create an action potential or 'fire' i.e. essentially a spike will be emitted which can be transmitted to other neurons connected to the output. The sub-threshold membrane voltage of an exponentially decaying LIF neuron at any time t is given by –

$$V(t) = \sum_i w_i \sum_{t_i} K(t - t_i) + V_{rest} \quad \ldots (7.1)$$

where $K(t - t_i) = V_0[e^{\left[-\frac{t-t_i}{\tau_m}\right]} - e^{\left[-\frac{t-t_i}{\tau_s}\right]}]$ and,

$t_i =$ spike times for synaptic input i,

K is a causal filter vanishing for $t_i > t$ and

τ_m, τ_s are the decay constants for membrane and synaptic integration respectively.

The neuron initially starts with a voltage V_{rest} and is then activated by several of its input synapses which try to raise the output voltage based on their relative importance indicated by the respective synaptic weights w_i. When the neuron voltage crosses a threshold, say V_{th}, the neuron emits a spike and the voltage returns to the original state i.e. V_{rest}. The voltage is held at V_{rest} for a short period called the refractory period irrespective of any input contribution during this period. After the refractory period, the neuron is ready to again accept synaptic input contributions and fire accordingly.

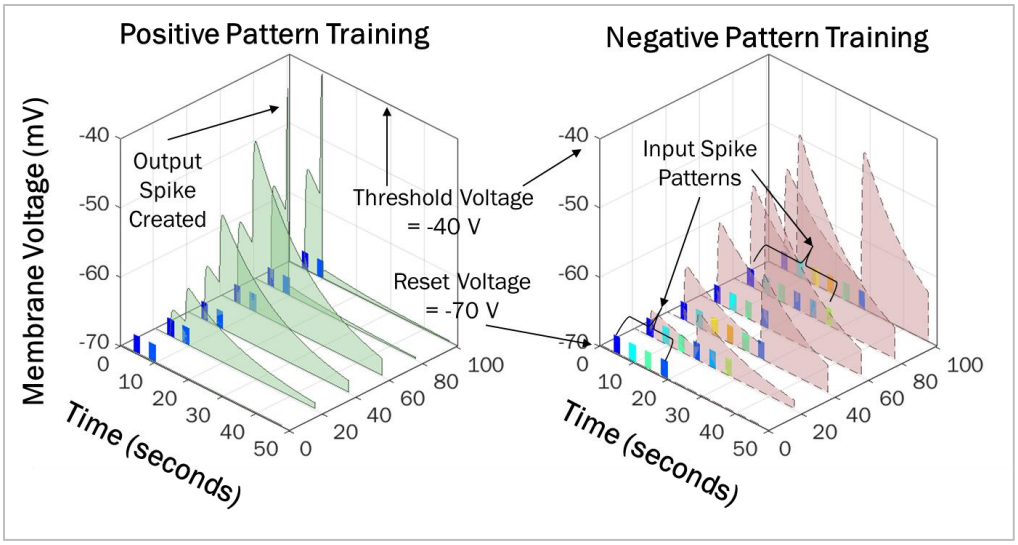


Figure 7.4. Synaptic Voltage Evolution across Training Epochs – Effect of Positive and Negative Training Patterns

The interesting fact here is that neuron's input synaptic weights can modulate the response of the neuron to different input patterns. So, the idea is that by properly adjusting (learning) the appropriate weights of the synaptic inputs, the neuron can be trained to fire only when certain spatiotemporal input spike patterns are detected. Figure 7.4 shows the response of an LIF neuron to different input patterns in terms of membrane voltage and output spike creation at different stages of training as will be discussed in more detail in Sections 7.5, 7.6. In the figure, the shaded portions show the change in voltage as different spikes (part of a pattern – positive or negative) are input at the neuron synapses. As can be seen, the voltage pattern corresponding to similar input patterns seems to change over time and for the positive patterns, near the end of training, the voltage pattern is such that the threshold is crossed right after the last spike in the pattern and this results in an output spike. In case of the negative patterns, even at the end of training, the input spike patterns should not be able to elicit an output spike i.e. the membrane voltage needs to below

threshold as is the case shown in Figure 7.4. Our intention is to come up with a training mechanism which can create such outcomes.

Various methods have been cited in the literature to accomplish this kind of training, out of which the Tempotron rule stands out because it was one of the first and robust enough to handle most of the application needs as mentioned in Section 7.1. The assumption of the Tempotron rule is that the neuron needs to be trained to distinguish between two sets of patterns i.e. a Tempotron neuron will be able to accomplish binary classification by firing a spike at the output when stimulated by a positive pattern spike train and, remaining quiescent when given a negative pattern spike train at the input. In order to achieve this, the Tempotron learning rule is applied over multiple epochs of training when a defined error in the output is used to motivate changes in the contributing input synaptic weights, thereby reducing the errors over time in a gradient descent-style training approach.

The Tempotron error is defined as follows –

For positive patterns, error = $V_{th} - V_{max}$

For negative patterns, error = $V_{max} - V_{th}$,

where V_{max} is the maximum voltage achieved during an epoch of training and V_{th} is the membrane threshold voltage.

This works because for positive patterns, error is generated when the max voltage in an epoch is lower than the threshold i.e. spikes are not fired. On the other hand, for negative patterns, the error case occurs when the max voltage is higher than the threshold voltage i.e. a spike is fired, though it is not expected. Based on this error function, the Tempotron learning rule is defined as follows –

For the positive error scenario i.e. if no output spike is elicited in response to a positive pattern, weights at synaptic input i are increased by –

$$\Delta w_i = \lambda * \sum_{t_i<t_{max}} K(t_{max} - t_i) \quad \ldots (7.2)$$

, where t_{max} = Time at which post-synaptic potential $V(t)$ reaches its maximal value and, $\lambda > 0$ specifies the maximum size of the synaptic update per input spike (or can also be looked upon as the learning rate for training purposes).

For the negative error scenario, i.e. if an output spike is elicited in response to a negative pattern, weights at synaptic input i are decreased by the amount specified in Eq. (7.2).

Eq. 7.2 shows how the time of occurrence of the maximum output voltage helps to determine the amount of training updates and how the kernel function helps to distribute the updates across the different synaptic inputs based on their contribution to the error. In particular, spikes which are far away from the max voltage time have little bearing on its value and thus weights for synaptic inputs corresponding to these are modified to a lesser extent than those which have input spikes very close to the max voltage time. The kernel function helps to modulate the rate at which the effect of a particular spike impacts voltage increments and then decay.

7.4. Baseline Pulse-based Networking Approaches

7.4.1. Pulse Position Coding Protocol

The Pulse Position Coding Protocol (PPCP) architecture, first explored in [102] aims to produce savings in terms of energy consumption by encoding any information to be conveyed in terms of the interval between consecutive spikes (discrete pulses). Depending on the value being encoded and the base of representation chosen, such intervals can be quite large. Hence, as shown

in [102], based on the energy consumption budget (transmission, reception, idling etc.) and the value to be encoded, we can choose an optimal base for representation of the value using multiple intervals (for each digit in the corresponding base representation) separated by discrete pulses or spikes. The trade-off is mainly in terms of energy vs delivery delay because the PPCP delivery latency depends on the interval between spikes and thus the value encoded and base of representation. If the number of spikes is increased by encoding the value in multiple small component intervals (digits), then delivery latency can be improved but at the cost of transmitting more spikes. On the other hand, larger intervals with less spikes can provide better energy utilization but at the cost of delay. However, this architecture can be very useful when the range of values to be encoded is limited. In either case, this approach has been shown to use significantly less energy compared to traditional packet-based networking approaches [102].

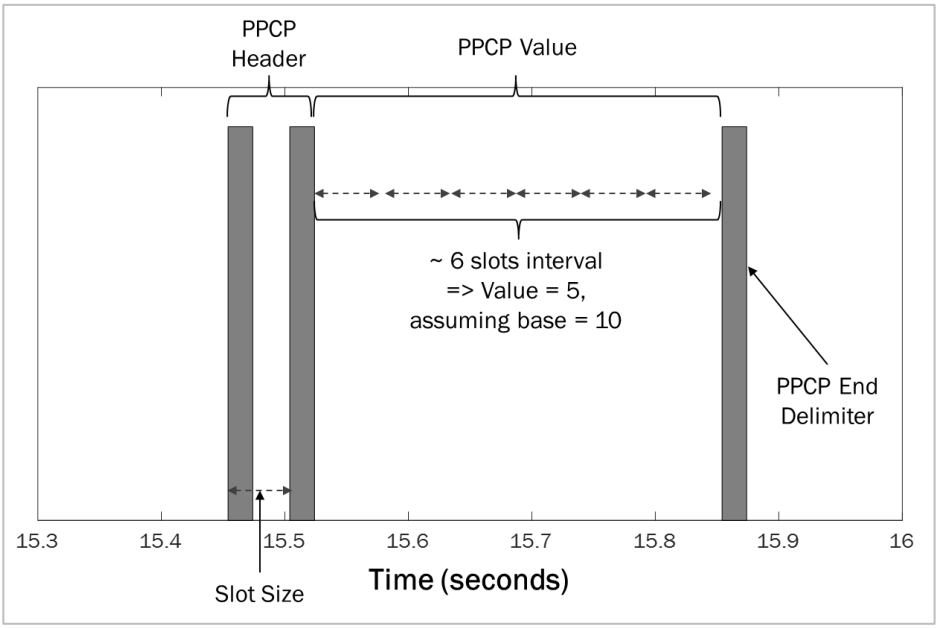


Figure 7.5. PPCP PDU Spike Representation

Figure 7.5 shows a sample PPCP representation of a value ‘6’ which in our application can indicate a node id = 6. The protocol data unit (PDU) consists of a header portion constituted by a

pre-selected number of spikes at short intervals (called slots) to indicate start of data, a tail portion using one less pulse compared to header to indicate end of data and the actual value encoded within with different digits separated by single spikes. The number of spikes needed for the different delimiters e.g. header, tail, digit separator etc. depends on the choice of base and the max value to be supported. For example, if we need to support only values between 0-9 and the base is chosen to be 10, then we will not require multiple digits, thus no digit delimiter spikes. We can choose two spikes for the header and 1 spike for the tail. This is the case shown in Figure 7.5. As requirements get more involved though, the number of spikes comprising header and tail will need to be increased to still distinguish these from each other and the digit delimiter.

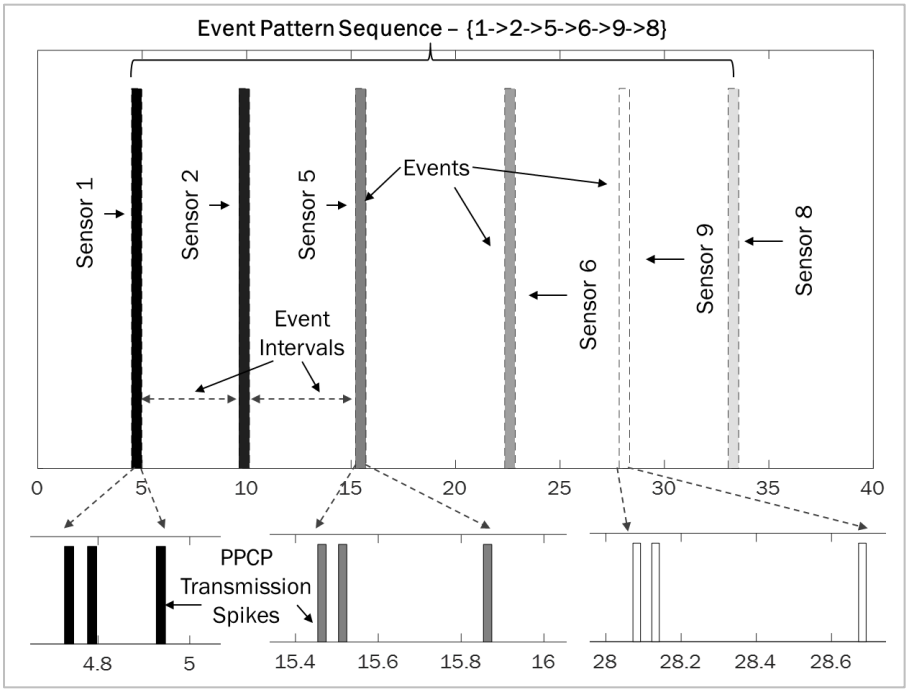


Figure 7.6. Event Spikes vs PPCP Spikes

The main idea here is that PPCP PDUs can be used to encode the essential information e.g. origin location of an event indicated by the corresponding sensing node id. This can be done using only three spikes (two for header, 1 for tail) if the value is limited in range and depending on the

base selection. Figure 7.6 shows how an event pattern in time would be translated into the corresponding PPCP spike patterns that will be sent over the channel with good energy utilization. The PPCP PDUs can be decoded at the sink to reasonably recover the original events' sequence pattern and the event spikes corresponding to the different nodes can be fed to the proper synaptic inputs of the pattern-detecting neuron in order. The Spiking Neuron can then fire on its output if the pattern closely matches one of the positive patterns.

7.4.2. Pulse Time Encoded Networking

The PPCP protocol discussed in the previous section does not require any synchronization among the nodes for operation. However, in many applications, such synchronization can be achieved with the use of a central Base Station which can periodically broadcast sync messages and keep the network in sync. This is especially true for small network scenarios. In [35] as well as in Chapters 3, 4, 5, 6 in the current thesis, the authors discuss a pulse networking protocol that is well-suited to such scenarios and can use the synchronization to achieve data transmission with very small number of pulses per event. For example, in the Pulse Networking protocol mentioned in [35], the event information can be encoded using a single pulse's presence or absence and event origin information can be encoded in the timing of the afore-mentioned pulse with respect to a synchronized time frame. Thus, using such a protocol, the number of pulses required is one-third of those required for PPCP transmissions in the best case. However, the trade-off here is the latency of such transmissions, because the synchronized time frame structure that is required here imposes inherent latency bounds on transmission. Moreover, synchronization is assumed which might not always be a given. However, assuming latency restrictions are relaxed, and synchronization is guaranteed, synchronous pulse networking can be a very efficient networking and medium access control policy. In the current work, we use a simplified version of the Pulse Networking

architecture as shown in Figure 7.7, with specific time slots assigned to every node for its event pulse transmission. This is also referred in the literature often as a Time Division Multiple Access scheme.

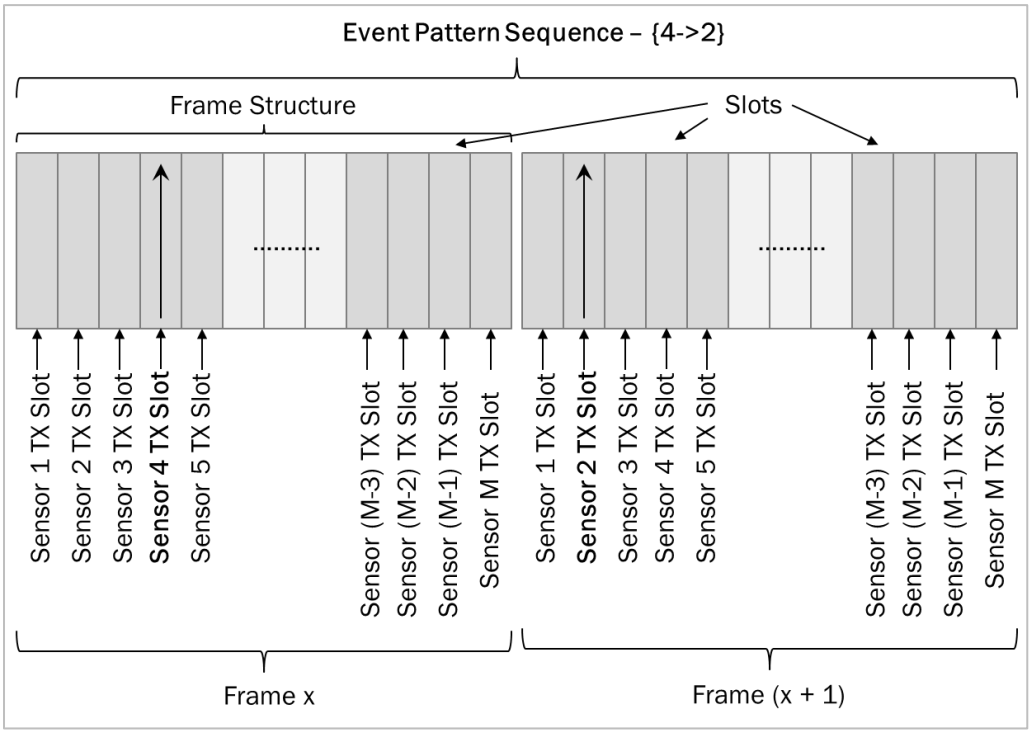


Figure 7.7. Pulse Time Encoding – Frame Structure

Each node has a specific slot assigned for its transmission and depending on the number of nodes which can interfere with each other's transmissions, we can design a frame structure with slots for each node in the local vicinity. For example, Figure 7.7 shows the case for a network with M nodes which can interfere with each other's transmissions, with the latter being mitigated using a medium access scheme comprising a frame with M slots. When an event is detected, nodes will transmit their information using a pulse only in their respective slot as shown for nodes 4 and 2 in Figure 7.7. Only one node can transmit in a specific frame in order to maintain event order resolution.

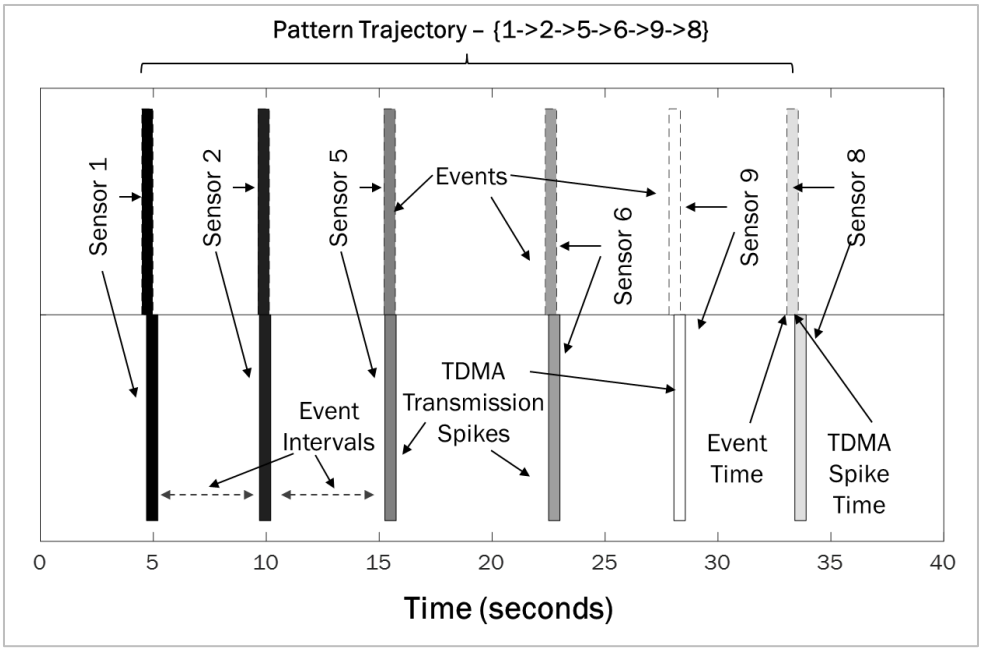


Figure 7.8. Event Spikes vs Pulse Time Encoding Spikes

In Figure 7.8, we show how this translates to our application scenario. Once the events are generated, the sensing nodes trigger spikes in their respective time slots and the resultant spike train is transmitted through the channel. The latter is like the original event generation spike train though slightly delayed because of the need to transport spikes from different nodes in their respective slots. This is because, a node's slot for transmission can be some time after the event is generated at the same node. This would make the transmitted pulse sequence somewhat different from the original event sequence pattern. However, when the number of nodes is limited and the slot size is small, the frame structure can be much shorter than the event intervals involved. Thus, small discrepancies in the spike timing due to framing can be smoothed out by the Spiking Neuron detection. Essentially, such small changes can be modeled like pulse drift and as we will show in Section 7.6, these can be handled very robustly using the Spiking Neuron architecture.

7.5. Adaptations of Spiking Neuron Learning for Pulse Networking

7.5.1. Networking Adaptations

As mentioned in Section 7.2, due to the need for a decoder at the receiver to read the event source id from the corresponding protocol specific PDU, the number of spikes that will be input to the Spiking Neuron at any synaptic input for a particular event will always be 1. What is more important is that the event intervals be sufficiently large compared to the protocol spike intervals or frame lengths. As mentioned in Section 7.2, we envision applications where the event intervals are on the order of seconds e.g. person walking across a network area with sensors spaced at a resolution of 10 meters or so. Such event intervals are indeed much larger compared to the actual transmission time for individual spikes in a PPCP PDU or the length of a TDMA frame. The latter is a function of the slot size used for transmission which is the minimal spacing between spikes such that they can be unambiguously resolved at the receiver and is generally on the order of milliseconds or less. We define the event intervals as the parameter beta and use values of beta ~ (4.5, 7.5) seconds as predicated by our sample application (discussed in Section 7.2), while using slot sizes in the order of milliseconds which leaves our event sequences unaltered by the choice of networking architecture.

7.5.2. Membrane Time Constant Selection

The membrane time constant and the synaptic integration time constants are two important parameters of the neuron model that decide which time scales the neuron will be working at and the temporal resolutions of patterns that the neuron can identify. If we have a general notion of the range for the event intervals that we are expecting for an application, then we can adjust the membrane and synaptic integration time constants accordingly such that the neuron is well-

equipped to handle event patterns for that application. Synaptic and membrane time constants determine the level of contribution for spikes at different times to the membrane voltage of the neuron. Thus, with a larger synaptic integration time constant, spikes would take longer to influence the membrane voltage. On the other hand, larger membrane time constants make sure that the spike contributions to membrane voltage do not die out fast. The membrane time constant selection is important based on the application, because it determines the resolution of the spike patterns i.e. if the precise timing of each individual spike will decide the membrane voltage or a certain subset of this. In the case of our application, we need the neuron to adapt to the intervals in the positive pattern specified but also not be too tied to the precise spike timing because we want to ensure sufficient generalization of the detection mechanism to effects like pulse drift. In order to achieve this, we chose a membrane time constant which is on the order of the event intervals chosen but slightly larger e.g. for event intervals in the range 4.5 – 7.5 seconds, we choose a membrane time constant around 16 seconds with a synaptic integration time constant much smaller than that (precisely 16 / 64 = 0.25 seconds) such that spikes have nearly immediate effect on the membrane voltage but fall off slowly (larger membrane time constant) such that effect of multiple consecutive spikes can be accumulated to result in the final voltage and thus output spiking.

7.5.3. Training Methodology

In our application scenario, event patterns can arise due to two scenarios – the order of the nodes triggering events and the timing between such event occurrence. In order to represent the first scenario, we define something called a trajectory. A trajectory is a sequence of network nodes which comprise an event order. For example, if a man walks around the network and triggers events at nodes 1, 2, and 3 in order, then the trajectory he traversed would be 1-2-3. Given the network topology and connectivity, we can only have certain number of valid trajectories. In the

application scenario, we will be interested in certain fraction of these valid trajectories as positive trajectories while all others will be designated as negative trajectories.

In order to train the neuron to identify the positive and negative patterns, the neuron needs to be exposed to representative positive and negative patterns over multiple epochs. In our application scenario, we envision having *p* valid positive trajectories and *n* valid negative trajectories. Each such trajectory can be used to generate a pattern by selecting the event intervals in the range corresponding to the application (controlled using the *beta* parameter). In order to keep the training uniform for positive and negative patterns, we expose the neuron to equal number of positive and negative pattern instances in every epoch. A collection of such epochs where all the positive and negative train patterns have been shown to the neuron is referred to as a batch. Every batch training is then repeated until the positive and negative training error values stop improving based on a convergence criterion as will be mentioned in more detail in Section 7.6.

The training and testing processes work as follows –

1) We design a set of positive patterns, one for each positive trajectory called the baseline positive patterns. During creation of the baseline patterns, we use various random event intervals in the specified range to provide generality. Similarly, a set of baseline negative patterns are also designed. For the negative baseline set, it is important to make an informed choice such that we can cover a large variety of negative patterns corresponding to all valid negative trajectories with a small number of negative baseline patterns. We have tried various strategies for selection of the negative patterns, but it appears that if we choose only patterns corresponding to trajectories which are most like the positive trajectory, we can get best results with least number of negative patterns. In order to calculate similarity, we use the relative percentage of similar spikes based on the gestalt pattern matching metric as available in the Python difflib helper called SequenceMatcher [107].

We modified this slightly to make sure that when the second pattern contains the first pattern, then the similarity value is capped at 100%.

2) Based on the positive and negative baseline patterns, we create some positive train and negative train patterns by introducing jitter in a reasonable amount to the baseline patterns as expected for the application. The motivation here is to expose the neuron to various generalized versions of the positive and negative patterns during training such that it can be robust during actual operation. The jitter is mainly used to model the effect of pulse drift errors. The number of both positive and negative training patterns is chosen as the maximum of the number of positive and negative baseline patterns. For both positive and negative cases, if the number of baseline patterns are less than the number of training patterns, some of the baseline patterns are repeated during training.

3) During training, the patterns in the training set are input to the neuron. Every positive train pattern is followed by a negative input pattern. For each pattern as input, the neuron integrates the input spikes and generates a spike or not based on the membrane voltage evolution. If the neuron response is not as expected (i.e. spike for positive pattern or no spike for negative pattern), an error is calculated based on the maximum voltage achieved by the neuron as stated in Section 7.3. Based on the error values, the input synaptic weights are updated to reflect the contribution to the error from multiple synaptic inputs. After the weights are updated, the process is repeated for the next training pattern and so on, until all training patterns are exhausted. This completes a batch of training. Training batches are repeated until the average error values over batch windows stop improving.

7.5.4. Test Methodology

We generate test positive patterns based on the positive baseline patterns and by introducing various amounts of jitter but unrelated to the train patterns' jitter. Thus, the positive test patterns are generalized versions of the baseline patterns, but different from the train patterns. For the negative patterns test, we decided to choose patterns based on all valid negative trajectories and we choose multiple instances of patterns with each trajectory but with different beta values in the range specified. We refer to these as unknown patterns as the Spiking Neuron has not necessarily been shown any of these (though some negative trajectories are covered in negative training) and we would like to train the neuron such that all or most of these unknown patterns can be detected. By varying both the trajectories as well as jitter, we aim to evaluate the generalization efficacy of the Spiking Neuron detection architecture.

7.6. Simulation Process and Performance Results

7.6.1. Simulation Environment and Process

In order to simulate the Spiking Neuron-based learning architecture using a Tempotron-style rule, we created a network simulation and learning framework in Python. In the framework, we can create arbitrary network topologies and simulate spike-based pulse communication using PPCP and Pulse Time Encoding-based protocols (as detailed in earlier section 7.4). In order to simulate the Linear Integrate and Fire neuron characteristics (as discussed in Section 7.3) and the corresponding output behavior in response to input spike patterns, we utilize the python interface to the neural simulation tool NEST also known as PyNEST [108]. This program can efficiently simulate the behavior of various neuron models, but we chose the LIF neuron because it is simple to understand yet sophisticated enough for our learning demonstration. The neuron parameter

settings such as membrane time constant, synaptic integration time constant, refractory period, threshold voltage etc. have been noted in Table 7.1. Most of these parameters have been chosen to use standard values [80], though we have adapted the membrane and synaptic integration time constants and spiking threshold voltage based on our application timing requirements as explained in the Section 7.5. We have also included various parameters related to the learning process in Table 7.1. These include the epoch time which was chosen to incorporate the longest possible pattern trajectory (i.e. trajectory of length = no. of nodes) and the largest possible event intervals (i.e. 7500 ms between each pair of events). The number of batches that the simulation learning process can run for is also kept high (500) and a convergence criterion is defined as follows which allows for training termination. We realized that the training should be regarded as converged when the training error can no longer be appreciably improved upon over subsequent batches of training. Thus, we defined the training as converged when the relative average error change over a window of the last 5 batches falls below 1%.

Table 7.1. Experimental Parameters

Symbol	**Parameter Name**	**Parameter Value**
N	No. of synaptic inputs	9
n	%. of negative trajectories chosen for negative pattern generation	Variable – 1, 2, 4, 8, 16.
β	Event Interval	Uniform (4.5, 7.5) seconds
s	Slot Size	50 ms
λ	Learning Rate	Variable – 0.01 (default)
V_{thres}	Threshold Voltage	-40 mV
V_{rest}	Reset Voltage	-70 mV
E	Epoch Length	$N * \beta_{max}$
W	Set of Synaptic Weights	All initialized to 0.01 (small, identical, non-zero value)

Table 7.1. (cont'd)

τ_m	Membrane Time Constant	Variable - 16 secs (default)
τ_s	Synaptic Integration Time Constant	$\frac{\tau_m}{64}$
t_{ref}	Refractory Period	2 ms
j	Jitter (%) - % of beta used as jitter	Variable – 0, 4, 8, 16.

Using the above-mentioned settings, we were able to train the Spiking Neuron with a variety of positive and negative patterns as discussed in Section 7.5 and then evaluate the learning efficacy using a range of selected test patterns. We finalized on using small, non-zero and identical weights for all synaptic inputs to start the training in order to remove any initialization bias. We also noticed that this performs better compared to a random initialization of synaptic weights because negative patterns have less chance of spiking in the absence of initial bias. For our negative training patterns, we chose only the ones which were very similar to the positive pattern trajectory as mentioned in Section 7.5. However, we made sure that the negative patterns which have trajectories with the positive pattern trajectory as a prefix are not part of the negative training set. This makes sense because, for example, if we have a positive trajectory 1->2->5->6, we would not expect our neuron to properly classify 1->2->5->6->9 as a negative because the neuron has already seen the first part as a positive trajectory. Hence, we do not include such trajectories as part of negative training and the failure to identify these is the limit of the Spiking Neuron performance. We report various statistics mainly regarding the detection accuracy performance of the neuron in a variety of pattern scenarios. We cover some internal probes into the learning architecture as well and discuss how these demonstrate the learning mechanism working efficiently. We also aim to demonstrate through our findings that the Spiking Neuron architecture discussed here generalizes well for a variety of application and error scenarios underscoring the importance of this research work.

7.6.2. Effect of varying the event interval range (beta)

In order to understand whether the current neuron architecture is robust enough to generalize its event sequence detection to a suitably large range of individual event intervals, simulations were performed with various event interval ranges, starting from a narrow one (4.5 to 4.75 seconds) up to a reasonably large one (4.5 to 7.5 seconds). The latter is large enough to mimic event intervals in a target application e.g. detection of person walking (speeds ~ [3 – 5 Kmph]) on a given trajectory inside the network area. As can be seen in Figs. 7.9 and 7.10 on the next page (for PPCP and TDMA cases respectively), the true positive rates and the unknown positives rates for all the interval scenarios are very similar when a moderately high value of n is used (e.g. n > 8% of all valid negative trajectories), which tells us that the platform is suitable for application across a wide variety of event interval ranges and should thus be applicable to a number of similar application scenarios. We also notice that there is minimal variability as a result of the networking protocol used i.e. PPCP or TDMA and thus the architecture should work well for either communication protocol scenarios.

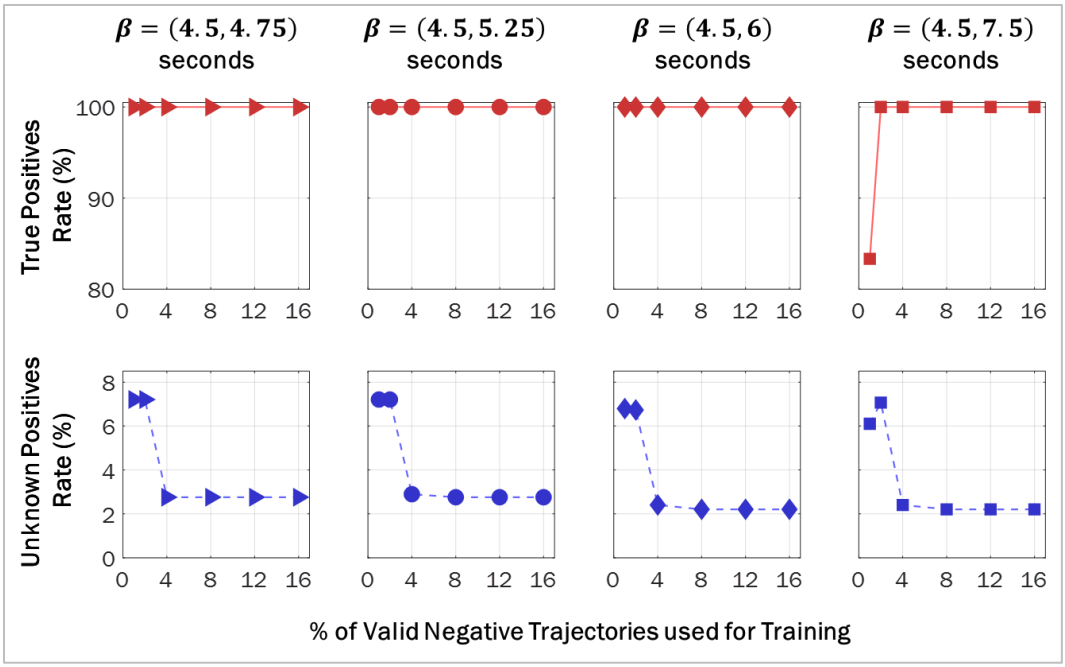


Figure 7.9. Effect of event interval range i.e. β on detection accuracy using PPCP communication

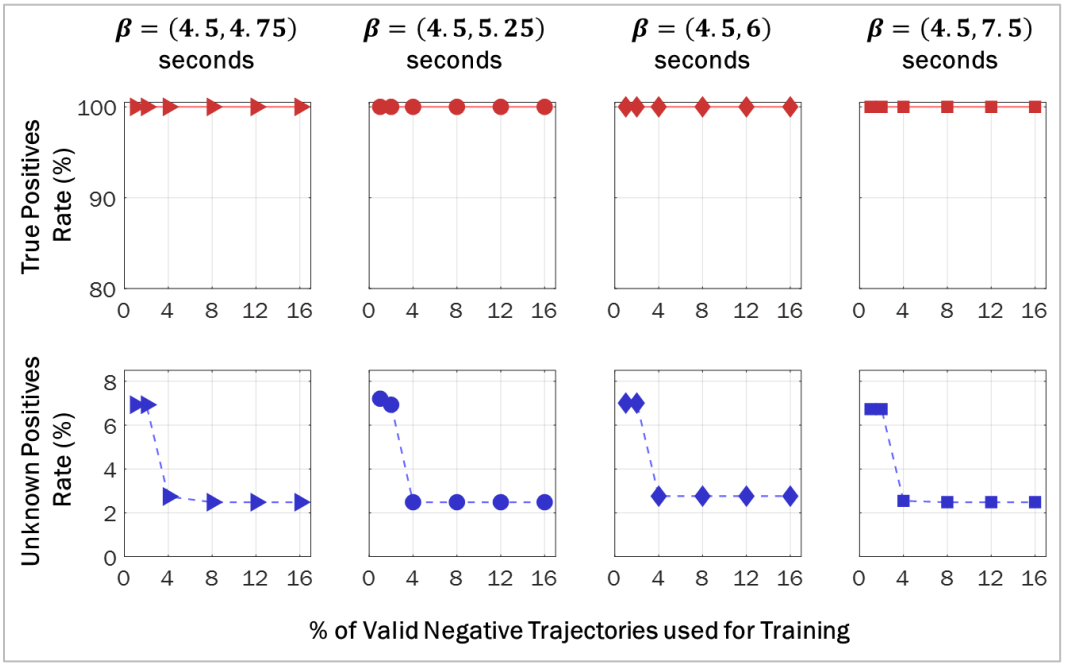


Figure 7.10. Effect of event interval range i.e. β on detection accuracy using TDMA communication

7.6.3. Effect of varying positive pattern length

In Figure 7.11, we show the effect of positive pattern choice of different lengths (indicated by parameter *pl*) on the detection accuracy performance. As is evident from the figures, the unknown accuracy results are best for the case of the positive patterns of largest length. This is intuitively expected of a well-working detection architecture, since the number of negative (unknown) patterns which can be similar to the positive patterns decrease as the positive pattern length increases.

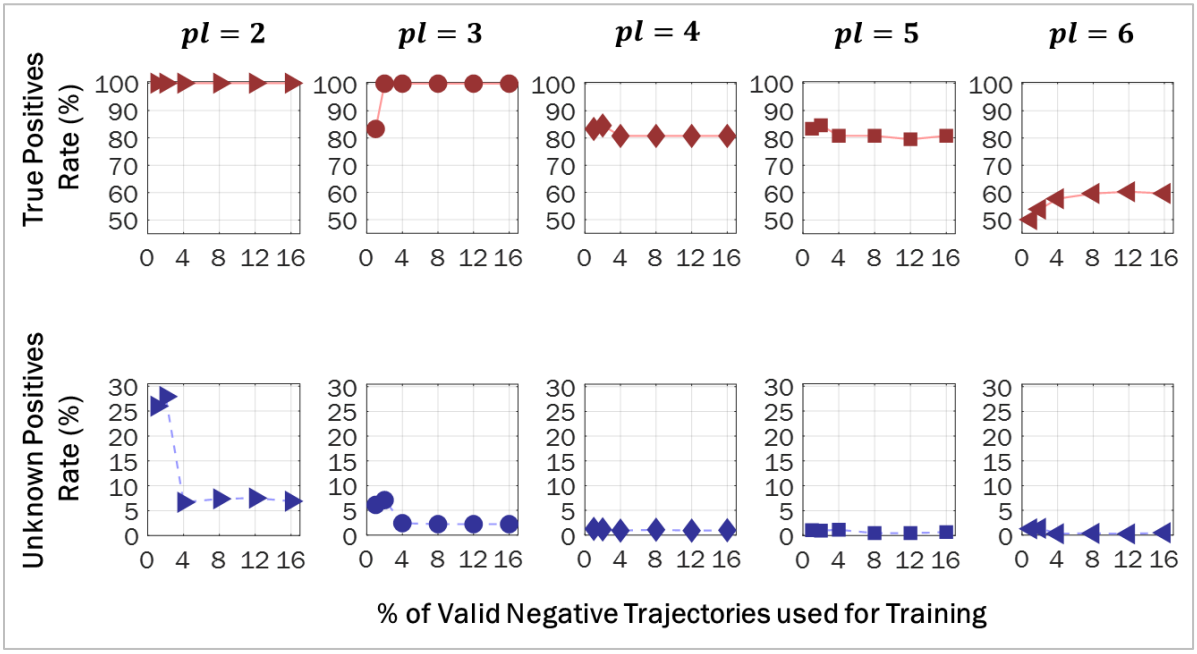


Figure 7.11. Effect of positive pattern length (pl) on detection accuracy across different negative training set sizes when using PPCP communication

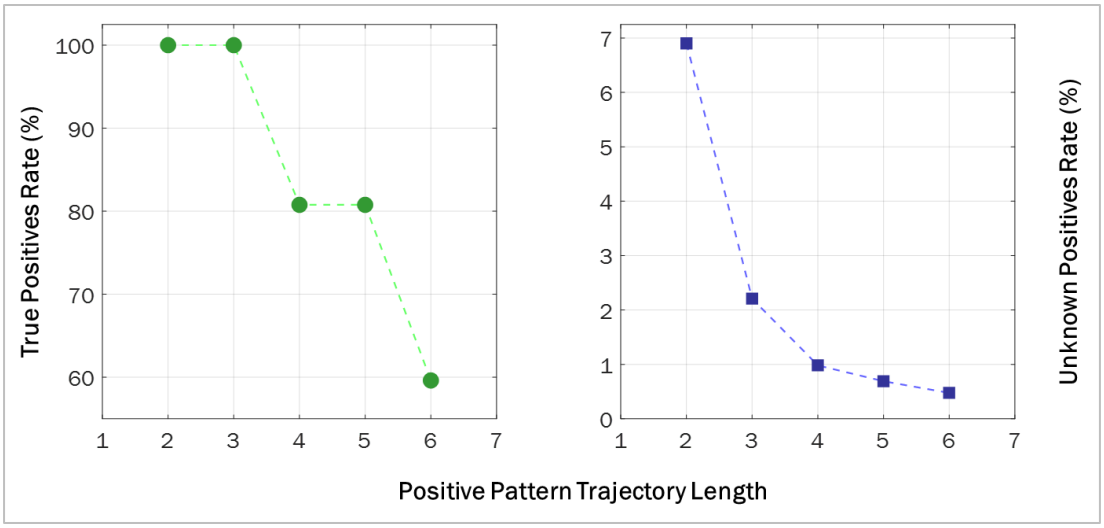


Figure 7.12. Effect of positive pattern length on detection accuracy when using PPCP communication

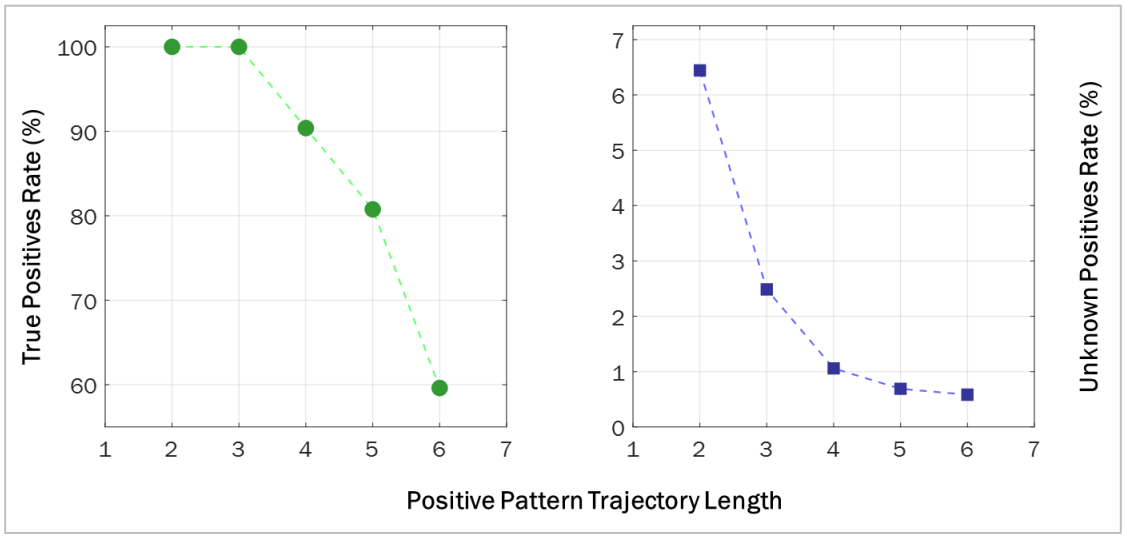


Figure 7.13. Effect of positive pattern length on detection accuracy when using TDMA communication

As seen in Figs. 7.12 and 7.13, the true positive results seem to decrease with increase in the trajectory length, but this is a trade-off with the unknown positive (which goes down). We can adjust the threshold voltage of the output neuron to ameliorate this situation as our application prioritizes true positives vs unknown positives. The unknown positives rate is exactly the result we expected from our close negatives training process because the training should be able to adjust the weights to distinguish the positive pattern from all but the ones very similar to it i.e. ones which

have the positive pattern as some subset. In particular, as evidenced by the data logs collected, in the case for highest n (% of negative trajectories used for training) for each positive trajectory length, the Spiking Neuron failed to identify only unknown patterns which were based on trajectories that had the positive pattern trajectory as a prefix. All other patterns were correctly flagged as negative. This is also evidenced from the pattern similarity vs unknown positives rate performance as shown in Figure 7.14.

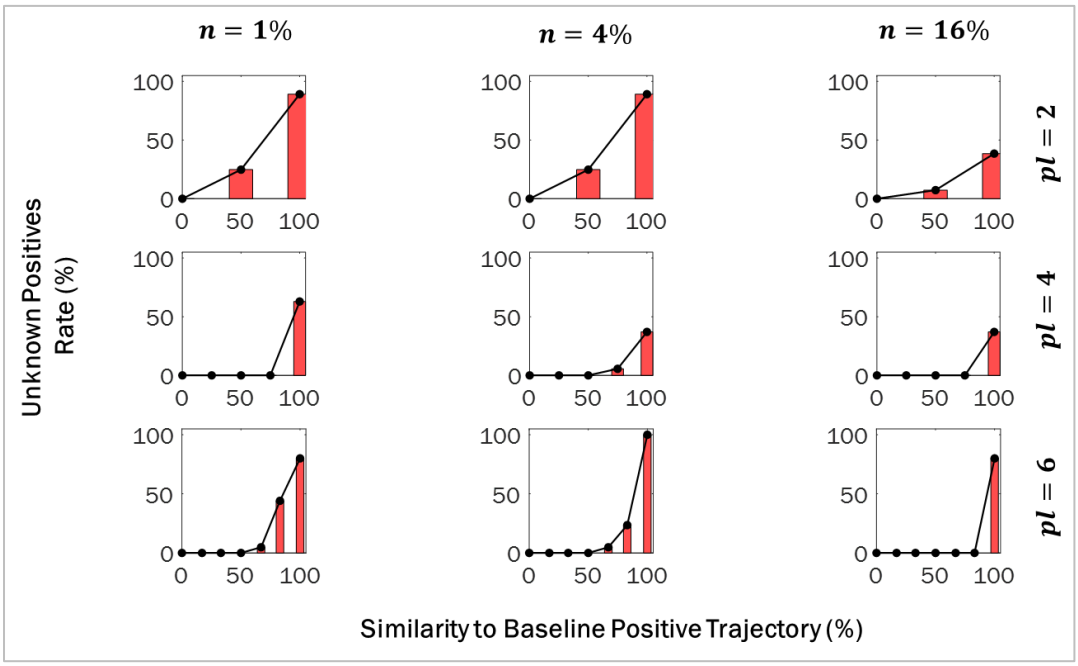


Figure 7.14. Similarity of unknown pattern to positive pattern trajectory and its effect on detection accuracy for different positive pattern trajectory lengths - PPCP

At low *n* values, some unknown patterns which have less similarity to the positive patterns are also mis-identified. However, with higher n, the percentage of such detections goes down and only a small percentage of unknown positives which are very similar (100%) to positive pattern are identified wrongly. Another observation is that the performance trajectories are similar for both PPCP and TDMA networking paradigms.

7.6.4. Effect of Pattern Type

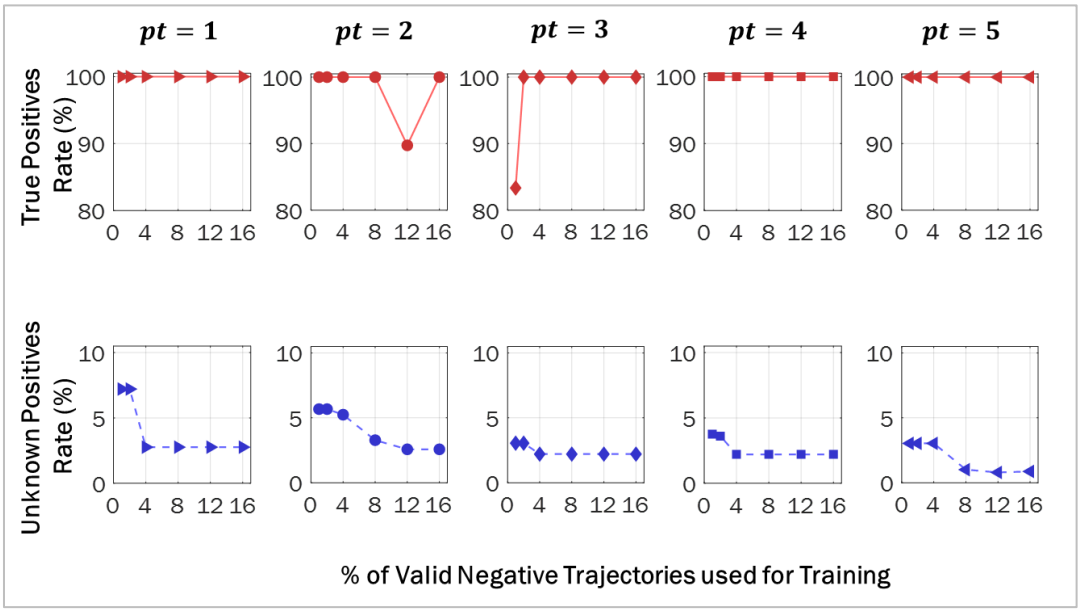


Figure 7.15. Detection Accuracy for Different Positive Pattern Types - PPCP

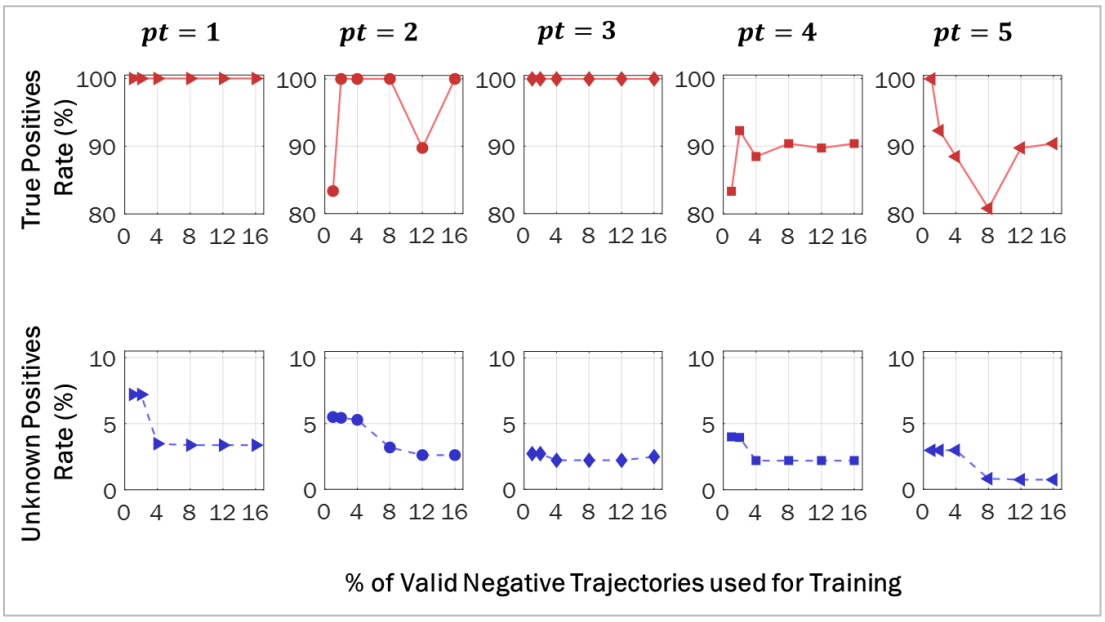


Figure 7.16. Detection Accuracy for Different Positive Pattern Types – TDMA

In order to understand whether our detection accuracy results are generalizable over a large variety of patterns, we decided to investigate the accuracy results for different positive patterns of

the same length. The results shown in Figure 7.15 and 7.16 are for different pattern types indicated by the value of *pt* = 1, 2, 3, 4, 5, corresponding to the positive pattern trajectories {1->2->5}, {6->9->8}, {7,->4->5}, {3,->6->9} and {2->5->8} respectively, each of length 3. As can be seen, the performance is largely similar across the pattern types in terms of both true positives and unknown positives and so is the case irrespective of the protocol used (PPCP / TDMA).

7.6.5. Effect of Spike Jitter

In Figs. 7.17 and 7.18 (see the next page), we show the effect of varying amounts of spike jitter on the detection performance of the Spiking Neuron. Spike Jitter as defined before is an amount of drift introduced in the train as well as test patterns in order to incorporate the practical implications of real-life pulse drift errors. Such pulse drifts can occur due to transmitter, receiver issues as well as energy constraints in the system which delay transmission / reception to conserve energy. Spike Jitter is incorporated as a relative proportion of the beta intervals used i.e. the event interval range. As can be seen in the results graphs, spike jitter up to a range of 16% seems to have little if any effect on the true positives as well as unknown positives rates for the Spiking Neuron. Only at the highest jitter in this range is the True Positives Rate slightly affected while the Unknown Positives Rate remains largely unchanged. This indicates that the Spiking Neuron architecture described here should work well for varying amounts of realistic spike jitter ensuring that the neuron's detection is suitably general for the purposes of the applications proposed earlier.

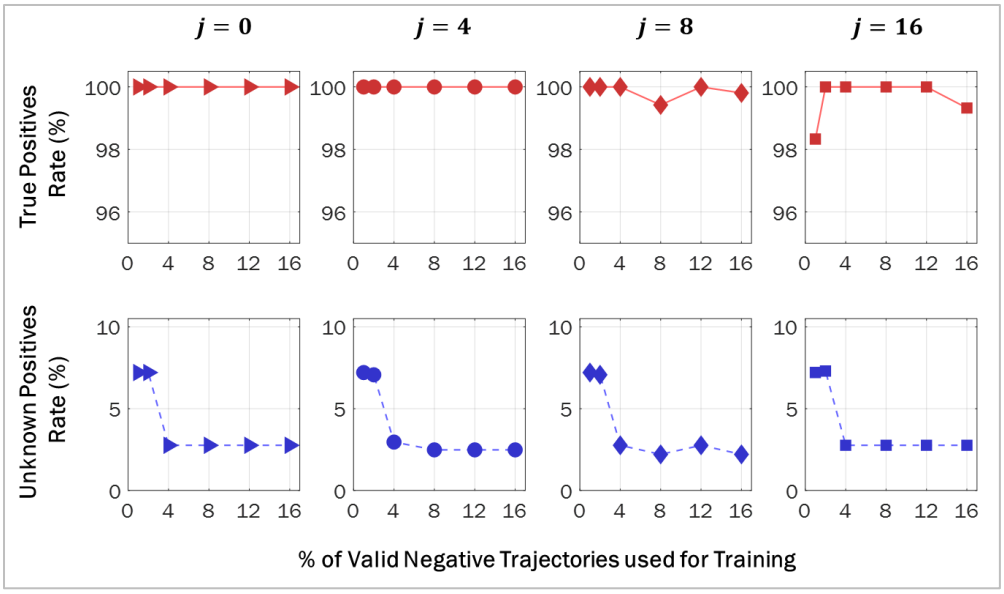


Figure 7.17. Detection Accuracy for Different Pattern Jitter Levels - PPCP

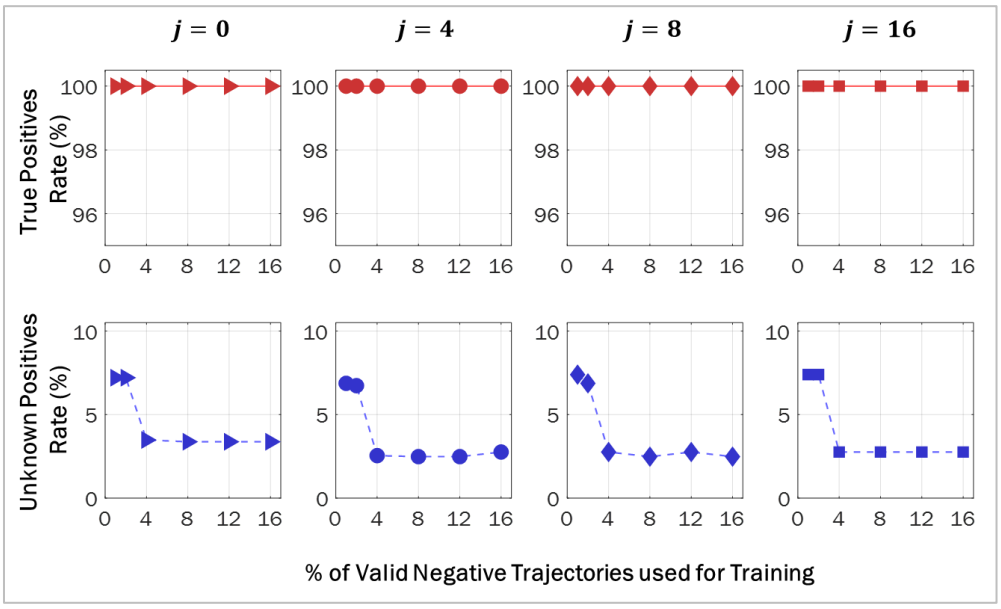


Figure 7.18. Detection Accuracy for Different Pattern Jitter Levels – TDMA

7.6.6. Effects of Learning Rate Selection for Training

In order to understand the effect of learning rate on the event detection performance, we consider the Spiking Neuron learning performance at three different learning rates from 0.01 to 1.0 as shown in Figure 7.19.

As is evident from the performance plots, a lower learning rate seems to provide better performance, though we see that below 0.05, the performance improvement is not appreciable, so using 0.05 can be good enough especially because convergence would faster in this case. In our current application domain, convergence time is not of prime importance, so we can afford to use an arbitrarily low learning rate. This is turn ensures that the training process is smooth and can go down the error gradient without much unnecessary oscillation and thus reach the proper optima (and not get stuck at locally optimal solutions).

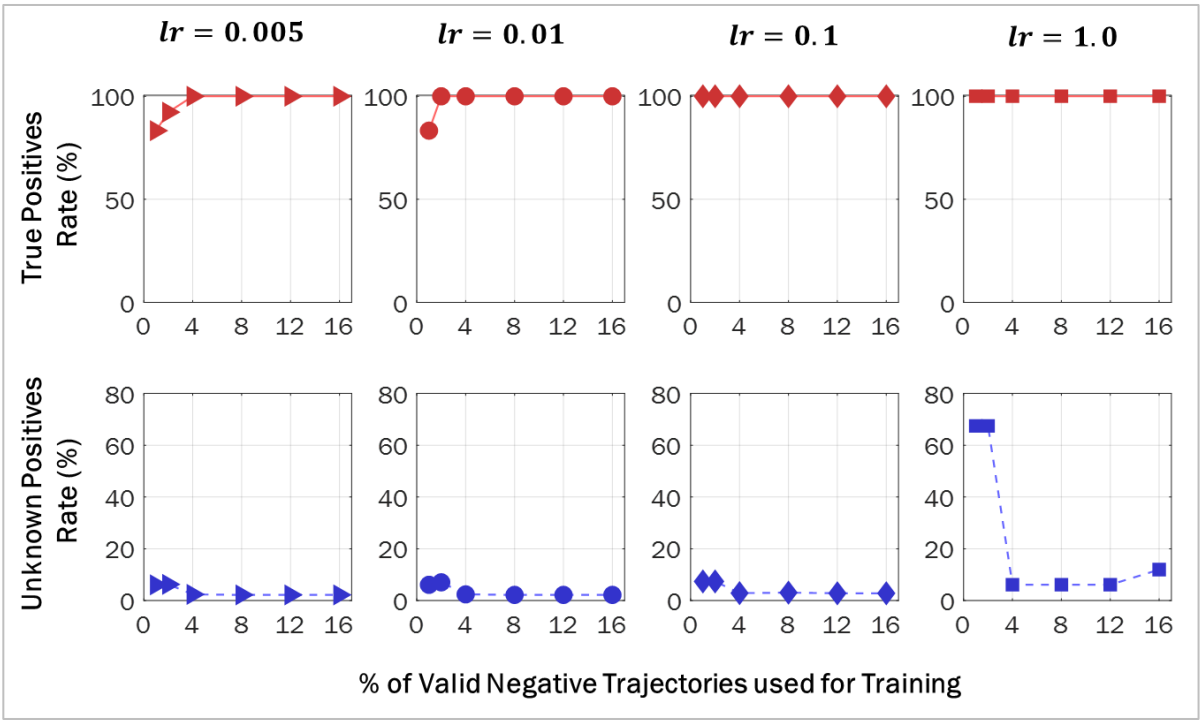


Figure 7.19. Detection Accuracy for Different Learning Rates

7.6.7. Effects of Membrane Time Constant Selection for Training

As shown in Figure 7.20, we have also investigated the effect of the neuron membrane constant choice on the detection performance. As can be seen, the performance is best when the membrane time constant is chosen at 16 seconds and falls off on either side for membrane time constants an order of magnitude higher and lower, especially for the lower scenario. It is to be noted here that the membrane time constant in our experiments is tied to the synaptic integration time constant as well (which is a fraction of the membrane time constant) and thus this parameter changes in sync as well. We can justify the membrane time constant = 16 seconds' superior performance by noting that this membrane time constant is of the order of the event intervals chosen which are in the range 4.5 – 7.5 seconds.

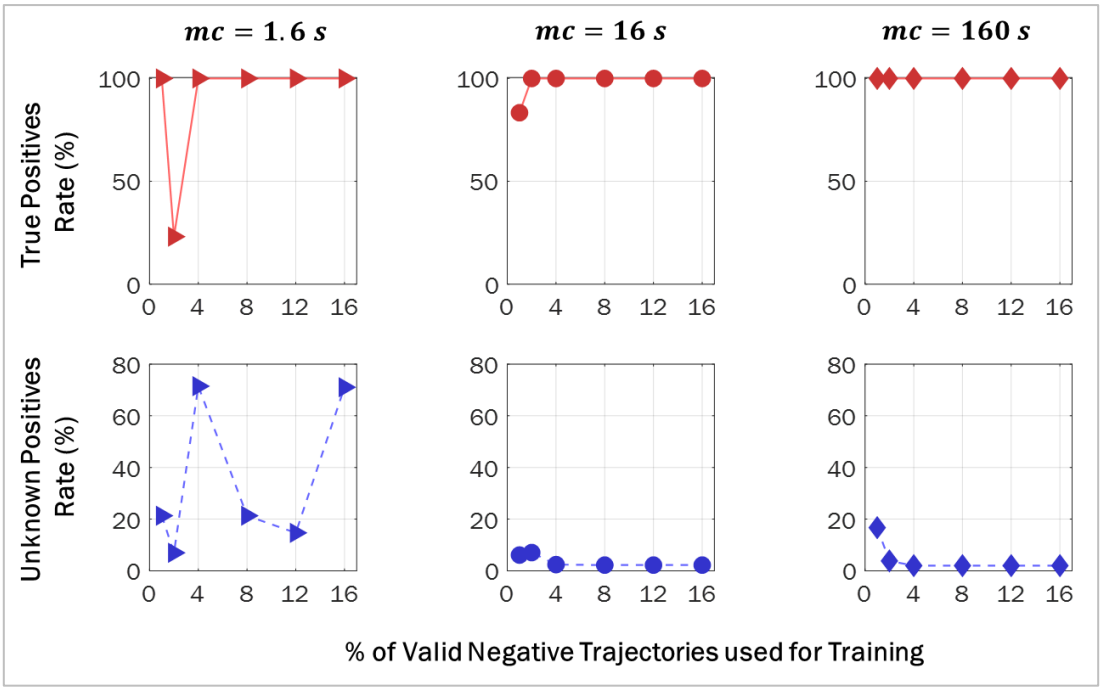


Figure 7.20. Detection Accuracy for Different Membrane Time Constants

7.6.8. Training Error Analysis

If we analyze the error plots, shown in Figure 7.21, we can see that the error values initially fall sharply due to positive pattern training until positive error falls to a very low value. This is when close negative patterns start contributing to the negative error until some of the extraneous synaptic weights are reduced with the final positive and error values settling. Convergence criteria based on relative change of windowed error seems to identify convergence well. Absolute threshold-based convergence criteria might miss the negative pattern training completely giving sub-par results, while our final convergence criteria based on relative change of the windowed average error stops the training much more gracefully and at a point where we would intuitively like to stop the training process. The appropriateness is also indicated by the superior detection accuracy results mainly in terms of unknown positives.

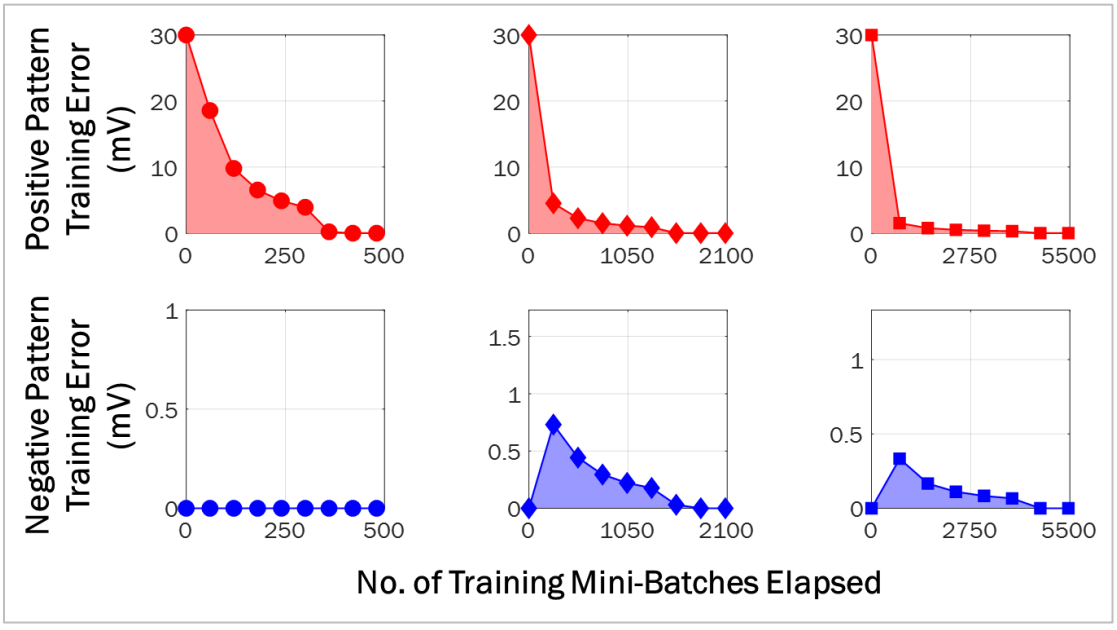


Figure 7.21. Positive and Negative Pattern Training Error Evolution across Training Mini-Batches

7.6.9. Synaptic Weights Evolution Analysis

In Figure 7.22, we show the synaptic weights evolution for different synaptic inputs as the training proceeds from start to convergence. In the case illustrated in Figure 7.22, the pattern trajectory is 1->2->5 and we would expect that the weights for these synaptic inputs should go up while others should go down. This is exactly the case as we go toward convergence. Moreover, we see that as n is increased i.e. more negative patterns are used during training, the weights of synaptic inputs which are not part of the positive pattern are also changed, in this case on the negative side such that the patterns which are close to the positive pattern but not the same cannot trigger unwanted spikes.

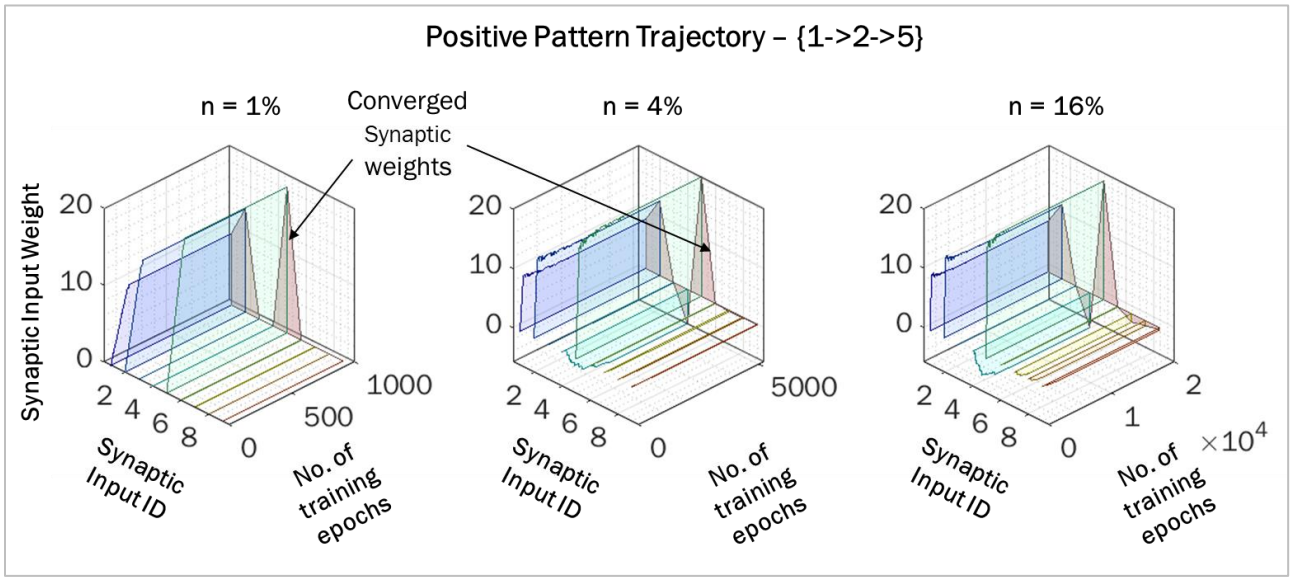


Figure 7.22. Synaptic Weights Evolution across Training Epochs for different sizes of negative train set

7.6.10. Effects of Pulse Loss and False Positive Errors

In order to demonstrate good adaptability of the current learning approach to practical application scenarios, we also evaluated the performance of the network in various spike insertion and deletion error scenarios. The former scenario is often referred to as a Pulse Loss scenario

which can result due to certain communicated pulses of the event pattern not reaching the final sink / destination due to channel collisions, reception problems etc. The latter scenario i.e. spike insertion in the original pattern can occur due to channel noise, receiver abnormality etc. and is generally referred to as a False Positive Error scenario. These two kinds of errors are important to consider in the context of the Spiking Neuron performance because abrupt deletion or insertion of spikes has the possibility to create valid patterns which might be mis-identified by the Spiking Neuron. It is to be noted here that, errors in the system will not all necessarily reach all the way up to the Spiking Neuron and affect performance. Since there is a decoder at the sink module (see Section 7.2) right before the Spiking Neuron, certain more obvious abnormalities in the pulse data can be recognized and removed / remedied even before reaching the Spiking Neuron and thus the neuron performance would not be affected. However, in certain cases as allowed by the network topology, communication protocol semantics etc., pulse loss or false positive errors might not create invalid patterns that can be detected at the decoder and thus creep into the Spiking Neuron input. We want to demonstrate here that in such situations, we can adapt the Spiking Neuron to the error scenarios by appropriately anticipating such errors and training the neuron to properly identify and classify these. This would be achieved with the trade-off of losing some granularity in negative pattern detection.

It is to be noted here that the Pulse Loss Error as well as False Positive Pulse Error probabilities are generally very low, on the order of 10^{-5} [8] for the network architectures we are considering. Hence, we consider only single pulse insertion or deletion errors, assuming that two or more such errors within a single pattern would be much less probable (because the error rates get multiplied as the number of errors go up, each error being independent of others) and can be neglected for all practical purposes. For example, in the case of the positive pattern trajectory 1-2-

5-6, we can assume pulse loss errors to create patterns such as 1-2-5, 2-5-6 (one pulse loss) etc. and not 1-2, 2-5 (two pulse losses) etc. False positive errors, on a similar token, would create patterns such as 4-1-2-5-6, 1-2-5-6-3 etc. In order to not mis-classify such error patterns as negative, we train with pattern instances of these trajectories as positive patterns while pattern instances of these trajectories are removed from the negative training scope. For each positive and negative trajectory chosen, we use multiple instances as before during training, such that the Spiking Neuron can generalize well for both event intervals and event occurrence sequence trajectories.

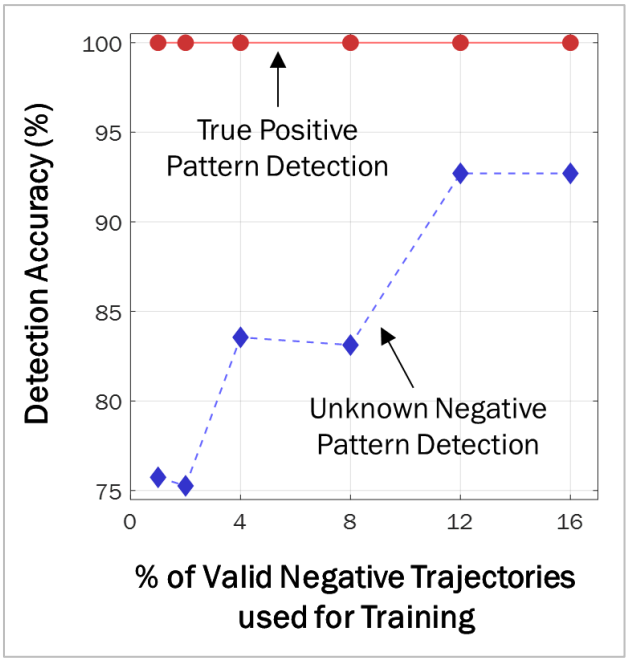


Figure 7.23. Performance in the presence of Single Pulse Loss Errors - Positive and Unknown Pattern Detection Accuracy across different number of negative trajectories used for training

As shown in Figs. 7.23 and 7.24, the true positive pattern detection (output spike) as well as unknown negative pattern detection (no output spike) accuracy approaches close to ideal value (100%) for high n i.e. negative training with a reasonably large percentage (12%-16%) of the total number of valid negative trajectories. The True Positives detection accuracy is particularly

consistent for Pulse Loss errors across all n (see Figure 7.23), while in the False Positives case, there is more variability (see Figure 7.24), albeit in a small range (90 – 100%), but the highest n performance is ideal i.e. 100%. Unknown Positives Detection Accuracy gets progressively lower with increasing n for both Pulse Loss and False Positive Error scenarios with the final values at n = 16% close to the ideal scenario for the positive trajectory used. Thus, we can reasonably conclude that with proper adjustments to the training procedure, we can adapt the Spiking Neuron architecture discussed here to effectively handle Pulse Loss and False Positive Error scenarios.

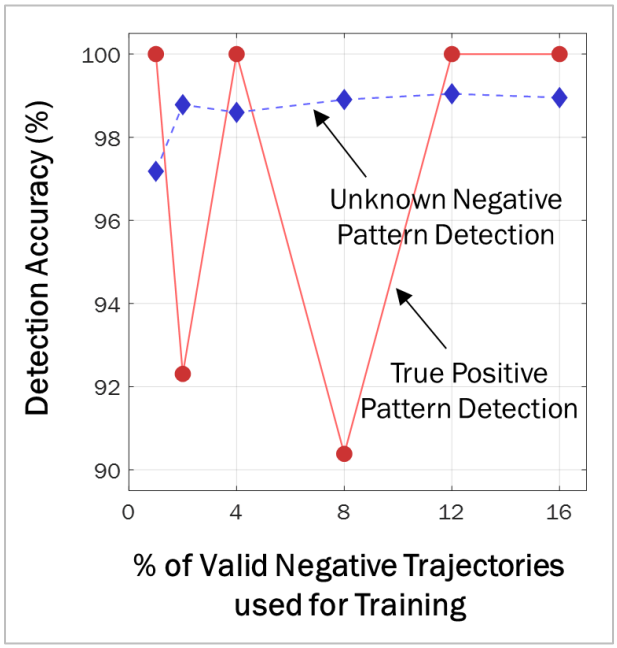


Figure 7.24. Performance in the presence of False Positive Pulse Errors - Positive and Unknown Pattern Detection Accuracy across different number of negative trajectories used for training

7.7. Summary

In this chapter, we demonstrate that using a single-layer Spiking Neuron architecture, we can efficiently and effectively detect the occurrence of pre-defined event occurrence sequence patterns which can be valuable in applications like Structural Health Monitoring. The proposed

architecture can easily be interfaced with various pulse networking architectures (for energy-efficient transport) and operate with high detection accuracy generalized over a reasonable range of event interval variation. The detection results are robust to decent amount of pulse drift errors and detection performance can be extended to cover pulse loss and false positive error scenarios using modified training pattern sets. We lay down details of the architecture implementation and learning parameters, training methodology and the corresponding rationale. Because of the simple architectural design but robust performance, this approach can be a good choice for resource-constrained applications. In the next chapter which will conclude this thesis, we will talk about future research work in this direction. Especially, we will discuss how we can intend to extend the application and network size and scope to demonstrate that the Spiking Neuron detection performance scales well irrespective of the size of the network. We will also outline plans to use layering of the spiking neurons to achieve even better performance when the problem complexity is increased.

CHAPTER 8: SUMMARY AND FUTURE WORK

8.1. Summary

In this thesis, we have laid the foundations for the development of a holistic framework for applications like Structural Health Monitoring using maintenance-free wireless sensor networks. The proposed solution would be powered by ambient vibration energy harvesting to provide maintenance-free operation. Communication across the network will be based on energy-efficient through-substrate ultrasonic pulse-based communication. This would enable reliable networking performance and consistent network uptime despite the unpredictability of harvesting-powered operation. Through-substrate links based on piezo-electric transducers would enable wire-free communication without the need for retro-fitted wireless radio infrastructure deployment. We also outlined a Spiking Neuron based low-complexity event pattern detection architecture. The latter would enable easy identification of structural anomaly patterns based on the spatiotemporal binary event information available from across the structure. The low-energy Spiking Neuron architecture can also enable some amount of in-network processing, even on intermediate energy-constrained network sensing modules, instead of delegating all processing to the Central Base Station. This can provide savings in terms of networking cost and faster detection and response.

In the various chapters of this thesis, we have developed the different components of the final envisioned architecture. This includes scalable and energy-aware pulse-based networking, through-substrate pulse networking in energy-harvesting-powered systems as well as design and evaluation of a single-layer Spiking Neuron based spatiotemporal event pattern detection architecture.

Future work on this broad topic can go along various routes. To start with, further research will need be carried out on the Spiking Neuron based detection architecture including an extensive evaluation of the same when used in more complicated (non-binary) pattern detection scenarios. Specifically, the weaknesses of the single-layer spiking neuron architecture will be scrutinized in such scenarios and improvements such as multi-layer designs will be considered and evaluated. The Spiking Neuron detection architecture can also be incorporated into a realistic Structural Health Monitoring application based on pulse networking, for instance, on an airplane wing and performance of the complete system can be evaluated in terms of detection accuracy to establish the advantages of this architecture. Beyond this, further research can also look into developing energy-harvesting-awareness mechanisms within the pulse networking framework (in addition to the energy-awareness syntaxes as discussed in prior chapters of this thesis) to make it even better suited to ambient harvesting-powered operation.

In the following sections, we first discuss the application architecture envisioned in this thesis and then consider the future work proposals in some more detail.

8.2. Application Architecture

In Figure 8.1, we depict a high-level vision of the application architecture being proposed. As an outcome of this thesis, we envision a Structural Health Monitoring application based on through-substrate pulse communication and spiking neuron based low-power detection of anomalous spatiotemporal event sequence patterns. In an example application scenario of an airplane wing structure monitoring, the network of sensors would be deployed over the wing substrate as shown in Figure 8.1 (a). The distributed sensors will be grouped into a cellular abstraction (hexagonal cells shown here) with multiple sensors per cell for redundancy. All event addressing will be on a cellular resolution.

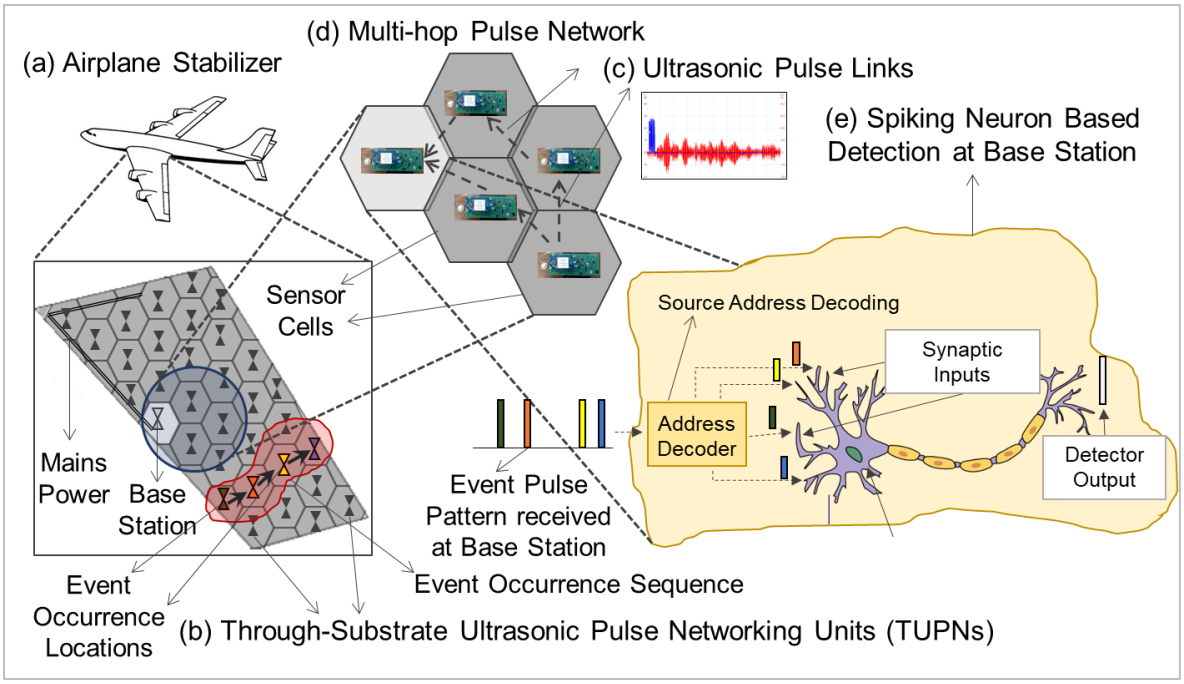


Figure 8.1. Structural Health Monitoring Platform based on Pulse Communication and Spiking Neuron Based Detection

It is to be noted that all the sensors will be equipped with ultrasonic through-substrate communication-enabled pulse communication transceivers and such modules would be referred to as Through-Substrate Ultrasonic Pulse Networking (TUPN) units. When an event is detected based on local sensing, the corresponding TUPN unit would transmit a pulse to indicate occurrence of the event. This pulse would then be routed multi-hop along the network toward the Base Station over ultrasonic links based on the pulse networking semantics. The pulse networking protocol enables preservation of the source id information as well as next-hop routing information which enables the event information to reach from source to sink, that is the Base Station for further processing.

As shown in the shaded red area on the bottom of the wing network, four cells (shown by the colored sensors) have event occurrences in a specific order as indicated by the arrows. This is

a sequence of events laterally across the wing and in a sequence from left to right, which might be indicative of a stress pattern of note. In order to detect such an event sequence pattern at the Base Station, the sequence of event pulses will be fed into a Spiking Neuron after appropriate address decoding to differentiate the pulse sources. The Spiking Neuron will look at the event sequence pattern and based on its pre-trained synaptic input weights, process the input pattern and create an output spike when an event pattern of note is detected. We have already evaluated various components on this architecture in Chapters 4 – 7. In future work, we intend to incorporate all these elements into a real application such as airplane wing monitoring and evaluate the performance in various ambient energy availability and network architecture scenarios. Using the performance analysis of such a study, we aim to establish the proposed architecture as a prime candidate for use in SHM applications.

8.3. Extending Single-Layer Spiking Neuron-based Event Pattern Detection

In Chapter 7, we have shown preliminary results on the detection performance of a single-layer Spiking Neuron based detection architecture for a simple application scenario and a limited network topology. Notably, we have considered detection of a single positive pattern vs multiple negative patterns. We have done quite some analysis on the effect of spike jitter to ensure robustness of the detection mechanism as well as how to adapt the learning when channel errors are appreciable. We have also shown that the detection architecture is well adapted to work in consort with various energy-efficient discrete pulse-based networking mechanisms. In a more typical application though, we might have many such positive patterns which need to be detected by the same application. We have also noticed in some preliminary experiments that when the number of positive patterns increases, a single-layer Spiking Neuron architecture might be limited in terms of detection performance. Hence, future work can involve exploring multi-neuron and

multi-layer Spiking Neuron approaches and when they would be better suited compared to a single-layer single neuron architecture to ensure detection performance. The premise is that multiple neurons can share the detection load by handling different parts (sub-patterns) of the chosen pattern. A progression of layers can be trained to look at higher level features with low granularity in the starting layers, requiring individual neurons in the starting layers to be less accurate and less computationally powerful. In such scenarios, important considerations would be how to design the multi-layer neuronal connectivity to achieve best detection performance with the least number of additional neurons. Work can also be done in applying other Spiking Neuron learning rules apart from the Tempotron learning mechanism as used here. Use of more sophisticated learning mechanisms might enable better performance with minimal number of neurons added (and thus energy expended).

8.4. Development of Energy-Harvesting Awareness in Pulse Networking

In Chapter 5 - 6, we have developed energy-aware syntaxes within the discrete pulse-based networking framework. These enable the protocols to operate well in slow harvesting scenarios such as ambient energy harvesting. It is to be noted that such performance does not assume any knowledge of the energy harvesting availability. It only considers the energy availability in each network node's energy storage. If some information or prediction of the energy harvesting availability is available, theoretically an improved utilization of the energy input can be achieved. Upcoming work can be aimed at tackling this aspect. Specifically, research attempts can be made to predict energy harvesting profiles based on machine learning approaches to enable a true energy-harvesting-aware pulse networking platform for optimal network energy utilization. If historical data on the harvesting availability profiles are available, various time series prediction mechanisms can be utilized for forward prediction and planning including traditional mechanisms like

exponential moving averages and Auto Regressive Integrated Moving Averages (ARIMA) as well more sophisticated mechanisms such as the one mentioned in [109] as well as Deep Neural Network-based mechanisms that utilize recurrent layers such as Long Short Term (LSTM) memory networks [110]. The idea is that being able to predict the harvesting availability will enable better planning of system resources and enable more efficient utilization with less wastage.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. F. Othman and K. Shazali, "Wireless Sensor Network Applications: A Study in Environment Monitoring System," in *Procedia Engineering*, vol. 41, pp. 1204–1210, Jan. 2012.

[2] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore, "Environmental Wireless Sensor Networks," in *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1903–1917, Nov. 2010.

[3] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, 2003, pp. 139–148.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, New York, NY, USA, 2002, pp. 88–97.

[5] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2004, pp. 214–226.

[6] Hanbiao Wang, J. Elson, L. Girod, D. Estrin, and Kung Yao, "Target classification and localization in habitat monitoring," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, 2003, vol. 4, pp. IV–844 - IV-847.

[7] N. Xu *et al.*, "A Wireless Sensor Network For Structural Monitoring," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2004, pp. 13–24.

[8] S. Das and S. Lorenz, "Through-Substrate Event Reporting using Harvested Energy in Ultrasound Sensor Networks," in *Proceedings of IEEE Globecom 2015*, San Diego, CA, 2015.

[9] M. Li and Y. Liu, "Underground Structure Monitoring with Wireless Sensor Networks," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, New York, NY, USA, 2007, pp. 69–78.

[10] S. Kim *et al.*, "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, New York, NY, USA, 2007, pp. 254–263.

[11] A. Arora *et al.*, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," in *Computer Networks*, vol. 46, no. 5, pp. 605–634, Dec. 2004.

[12] Wei-Peng Chen, J. C. Hou, and Lui Sha, “Dynamic clustering for acoustic target tracking in wireless sensor networks,” in *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, Jul. 2004.

[13] S. Pattem, S. Poduri, and B. Krishnamachari, “Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks,” in *Information Processing in Sensor Networks*, 2003, pp. 32–46.

[14] J. Liu, J. Reich, and F. Zhao, “Collaborative In-network Processing for Target Tracking,” in *EURASIP J. Adv. Signal Process*, vol. 2003, pp. 378–391, Jan. 2003.

[15] D. J. Abadi, S. Madden, and W. Lindner, “REED: Robust, Efficient Filtering and Event Detection in Sensor Networks,” in *Proceedings of the 31st International Conference on Very Large Data Bases*, Trondheim, Norway, 2005, pp. 769–780.

[16] V. C. Gungor and G. P. Hancke, “Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches,” in *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[17] F. Salvadori *et al.*, “Monitoring in Industrial Systems Using Wireless Sensor Network With Dynamic Power Management,” in *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3104–3111, Sep. 2009.

[18] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” in *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[19] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks,” in *Wireless Networks*, vol. 12, no. 1, pp. 63–78, Feb. 2006.

[20] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks,” in *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, Oct. 2001.

[21] Mao Ye, Chengfa Li, Guihai Chen, and J. Wu, “EECS: an energy efficient clustering scheme in wireless sensor networks,” in *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, pp. 535–540, 2005.

[22] V. Cerf and R. Kahn, “A Protocol for Packet Network Intercommunication,” in *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, May 1974.

[23] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks,” in *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, 2004, pp. 224-235.

[24] H. Hassanein and Jing Luo, “Reliable Energy Aware Routing In Wireless Sensor Networks,” in *Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, 2006, pp. 54–64.

[25] R. C. Shah and J. M. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” in *2002 IEEE Wireless Communications and Networking Conference, 2002. WCNC2002*, 2002, vol. 1, pp. 350–355 vol.1.

[26] I. Stojmenovic and X. Lin, “Power-aware localized routing in wireless networks,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, Nov. 2001.

[27] M. Cardei, M. T. Thai, Yingshu Li, and Weili Wu, “Energy-efficient target coverage in wireless sensor networks,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2005, vol. 3, pp. 1976–1984 vol. 3.

[28] Seema Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, 2003, vol. 3, pp. 1713–1723 vol. 3.

[29] Y. Yao, Q. Cao, and A. V. Vasilakos, “EDAL: An Energy-efficient, Delay-aware, and Lifetime-balancing Data Collection Protocol for Heterogeneous Wireless Sensor Networks,” in *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 810–823, Jun. 2015.

[30] M. Dong, K. Ota, and A. Liu, “RMER: Reliable and Energy-Efficient Data Collection for Large-Scale Wireless Sensor Networks,” in *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 511–519, Aug. 2016.

[31] L. Lin, N. B. Shroff, and R. Srikant, “Asymptotically Optimal Energy-aware Routing for Multihop Wireless Networks with Renewable Energy Sources,” in *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1021–1034, Oct. 2007.

[32] M. K. Jakobsen, J. Madsen, and M. R. Hansen, “DEHAR: A distributed energy harvesting aware routing algorithm for ad-hoc multi-hop wireless sensor networks,” in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*, 2010, pp. 1–9.

[33] M. Younis, M. Youssef, and K. Arisha, “Energy-aware routing in cluster-based sensor networks,” in *Proceedings. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, 2002, pp. 129–136.

[34] J. C. C. Restrepo, C. G. Gruber, and C. M. Machuca, “Energy Profile Aware Routing,” in *2009 IEEE International Conference on Communications Workshops*, 2009, pp. 1–5.

[35] Q. Huo, B. Dong, and S. Biswas, “Cellular Pulse Switching: An Architecture for Event Sensing and Localization in Sensor Networks,” in *Distributed Computing and Networking*, D. Frey, M. Raynal, S. Sarkar, R. K. Shyamasundar, and P. Sinha, Eds. Springer Berlin Heidelberg, 2013, pp. 208–224.

[36] S. C. Liu and M. Tomizuka, “Vision and Strategy for Sensors and Smart Structures Technology Research,” in *Proceedings of the 4th International Workshop on SHM*, Stanford, CA, 2003, pp. 42–52.

[37] F.-K. Chang, “Structural Health Monitoring 2003: From Diagnostics & Prognostics to Structural Health Management,” in *Proceedings of the 4th International Workshop on Structural Health Monitoring,* Stanford University, Stanford, CA, September 15-17, 2003. DEStech Publications, Inc, 2003.

[38] Farrar Charles R and Worden Keith, “An introduction to structural health monitoring,” in *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 303–315, Feb. 2007.

[39] H. Salehi, S. Das, S. Chakrabartty, S. Biswas, and R. Burgueño, “Structural Assessment and Damage Identification Algorithms Using Binary Data,” in *Proceedings of the ASME 2015 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, vol. 2, Integrated System Design and Implementation; Structural Health Monitoring; Bioinspired Smart Materials and Systems, pp. 1-10, Sep. 2015.

[40] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, “Dependable Structural Health Monitoring Using Wireless Sensor Networks,” in *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 363–376, Jul. 2017.

[41] M. Bocca, J. Toivola, L. M. Eriksson, J. Hollmén, and H. Koivo, “Structural Health Monitoring in Wireless Sensor Networks by the Embedded Goertzel Algorithm,” in *2011 IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*, 2011, pp. 206–214.

[42] Q. Huo, B. Dong, and S. Biswas, “A cellular pulse switching architecture for binary event sensing,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 379–384.

[43] N. G. Elvin, N. Lajnef, and A. A. Elvin, “Feasibility of structural monitoring with vibration powered sensors,” in *Smart Materials and Structures*, vol. 15, no. 4, pp. 977, 2006.

[44] C. Huang and S. Chakrabartty, “An Asynchronous Analog Self-Powered CMOS Sensor-Data-Logger With a 13.56 MHz RF Programming Interface,” in *IEEE Journal of Solid-State Circuits*, vol. 47, no. 2, pp. 476–489, Feb. 2012.

[45] N. A. Alrajeh, S. Khan, and B. Shams, “Intrusion Detection Systems in Wireless Sensor Networks: A Review,” in *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, p. 167575, May 2013.

[46] V. Jakkula and D. J. Cook, “Anomaly detection using temporal data mining in a smart home environment,” in *Methods Inf Med*, vol. 47, no. 1, pp. 70–75, 2008.

[47] G. Werner-Allen *et al.*, “Deploying a wireless sensor network on an active volcano,” in *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, Mar. 2006.

[48] M. Li, Y. Liu, and L. Chen, “Nonthreshold-Based Event Detection for 3D Environment Monitoring in Sensor Networks,” in *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 12, pp. 1699–1711, Dec. 2008.

[49] X. R. Wang, J. T. Lizier, O. Obst, M. Prokopenko, and P. Wang, “Spatiotemporal Anomaly Detection in Gas Monitoring Sensor Networks,” in *Wireless Sensor Networks*, 2008, pp. 90–105.

[50] A. Jain, A. Jain, M. Gruteser, and M. Gruteser, “Benefits of packet aggregation in ad-hoc wireless network,” in *Computer Science Technical Reports (CU-CS-960-03)*, University of Colorado, Boulder, 2003.

[51] Z. Wang, E. Bulut, and B. K. Szymanski, “A Distributed Cooperative Target Tracking with Binary Sensor Networks,” in *IEEE International Conference on Communications Workshops, 2008. ICC Workshops ’08*, 2008, pp. 306–310.

[52] C. Fragouli, A. Orlitsky, É. Polytechnique, and F. Lausanne, “Silence is Golden and Time is Money: Power-Aware Communication for Sensor Networks,” in *Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005, pp. 1026-1035.

[53] Q. Huo, S. Biswas, and A. Plummer, “Ultra wide band impulse switching protocols for event and target tracking applications,” in *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2011, pp. 197–205.

[54] Q. Huo, J. Rao, and S. Biswas, “Pulse Switching: Toward a Packet-Less Protocol Paradigm for Event Sensing,” in *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 35–50, Jan. 2013.

[55] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” in *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May 2009.

[56] T. van Dam and K. Langendoen, “An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2003, pp. 171–180.

[57] C. M. Vigorito, D. Ganesan, and A. G. Barto, “Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks,” in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON ’07*, 2007, pp. 21–30.

[58] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2000, pp. 56–67.

[59] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power Management in Energy Harvesting Sensor Networks,” in *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, Sep. 2007.

[60] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, “Design Considerations for Solar Energy Harvesting Wireless Embedded Systems,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, Piscataway, NJ, USA, 2005.

[61] Jian Meng, Xuedan Zhang, Yuhan Dong, and Xiaokang Lin, “Adaptive energy-harvesting aware clustering routing protocol for Wireless Sensor Networks,” in *7th International Conference on Communications and Networking in China*, 2012, pp. 742–747.

[62] S. Abdollahzadeh and N. J. Navimipour, “Deployment strategies in the wireless sensor network: A comprehensive review,” in *Computer Communications*, vol. 91–92, pp. 1–16, Oct. 2016.

[63] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, 2001, vol. 2, pp. 472–476 vol.2.

[64] M. Younis and K. Akkaya, “Strategies and techniques for node placement in wireless sensor networks: A survey,” in *Ad Hoc Networks*, Jun. 2008, vol. 6, no. 4, pp. 621-655.

[65] F. Restuccia, G. Anastasi, M. Conti, and S. K. Das, “Analysis and Optimization of a Protocol for Mobile Element Discovery in Sensor Networks,” in *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 1942–1954, Sep. 2014.

[66] F. Restuccia and S. K. Das, “Lifetime optimization with QoS of sensor networks with uncontrollable mobile sinks,” in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015, pp. 1–9.

[67] R. Figura *et al.*, “ICELUS: investigating strategy switching for throughput maximization to a mobile sink,” in *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2016, pp. 1–8.

[68] A. D. Orcesi and D. M. Frangopol, “Optimization of bridge maintenance strategies based on structural health monitoring information,” in *Structural Safety*, vol. 33, no. 1, pp. 26–41, Jan. 2011.

[69] S. M. Khan, S. Atamturktur, M. Chowdhury, and M. Rahman, “Integration of Structural Health Monitoring and Intelligent Transportation Systems for Bridge Condition Assessment: Current Status and Future Direction,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2107–2122, Aug. 2016.

[70] Y. Q. Ni and K. Y. Wong, “Integrating bridge structural health monitoring and condition-based maintenance management,” in *Proceedings of the 4th International Workshop on Civil Structural Health Monitoring*, Berlin, Germany, 6-8 Nov. 2012.

[71] J. P. Lynch, K. J. Loh, J. P. Lynch, and K. J. Loh, “A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring,” in *The Shock and Vibration Digest*, vol. 38, no. 2, Mar. 2006, pp. 91 - 128.

[72] Park Gyuhae, Rosing Tajana, Todd Michael D., Farrar Charles R., and Hodgkiss William, “Energy Harvesting for Structural Health Monitoring Sensor Networks,” in *Journal of Infrastructure Systems*, vol. 14, no. 1, pp. 64–79, Mar. 2008.

[73] W. K. G. Seah, Z. A. Eu, and H. Tan, “Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges,” in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, 2009, pp. 1–5.

[74] N. G. Elvin, A. A. Elvin, and M. Spector, “A self-powered mechanical strain energy sensor,” in *Smart Mater. Struct.*, vol. 10, no. 2, pp. 293–299, Apr. 2001.

[75] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, “Modeling and Optimization of a Solar Energy Harvester System for Self-Powered Wireless Sensor Networks,” in *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2759–2766, Jul. 2008.

[76] D. Niyato, E. Hossain, M. M. Rashid, and V. K. Bhargava, “Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management,” in *IEEE Wireless Communications*, vol. 14, no. 4, pp. 90–96, Aug. 2007.

[77] L. Huang and M. J. Neely, “Utility Optimal Scheduling in Energy-Harvesting Networks,” in *IEEE/ACM Transactions on Networking*, vol. 21, no. 4, pp. 1117–1130, Aug. 2013.

[78] Z. A. Eu, H. Tan, and W. K. G. Seah, “Opportunistic routing in wireless sensor networks powered by ambient energy harvesting,” in *Computer Networks*, vol. 54, no. 17, pp. 2943 - 2966, Dec. 2010.

[79] Z. Guan, G. E. Santagati, and T. Melodia, “Ultrasonic intra-body networking: Interference modeling, stochastic channel access and rate control,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2425–2433.

[80] R. Gütig and H. Sompolinsky, “The tempotron: a neuron that learns spike timing-based decisions,” in *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, Mar. 2006.

[81] R. Florian, “The Chronotron: A Neuron That Learns to Fire Temporally Precise Spike Patterns.” in *Nature Precedings*, Nov. 2010, https://doi.org/10.1038/npre.2010.5190.1.

[82] F. Ponulak and A. Kasiński, “Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting,” in *Neural Computation*, vol. 22, no. 2, pp. 467–510, Oct. 2009.

[83] T. Masquelier, R. Guyonneau, and S. J. Thorpe, “Competitive STDP-Based Spike Pattern Learning,” in *Neural Computation*, vol. 21, no. 5, pp. 1259–1276, Dec. 2008.

[84] R. Urbanczik and W. Senn, “A Gradient Learning Rule for the Tempotron,” in *Neural Computation*, vol. 21, no. 2, pp. 340–352, Sep. 2008.

[85] N. Elvin and A. Erturk, “Advances in Energy Harvesting Methods,” in *Springer Science & Business Media*, 2013.

[86] S. R. Anton and H. A. Sodano, “A review of power harvesting using piezoelectric materials (2003–2006),” in *Smart Mater. Struct.*, vol. 16, no. 3, pp. R1–R21, May 2007.

[87] P. Shashank, “Advances in energy harvesting using low profile piezoelectric transducers,” in *Journal of Electroceramics*, vol. 18, issue 1, pp. 167-184, Sep. 2007.

[88] H. S. Kim, J.-H. Kim, and J. Kim, “A review of piezoelectric energy harvesting based on vibration,” in *Int. J. Precis. Eng. Manuf.*, vol. 12, no. 6, pp. 1129–1141, Dec. 2011.

[89] S. Roundy and P. K. Wright, “A piezoelectric vibration based generator for wireless electronics,” in *Smart Materials and Structures*, vol. 13, no. 5, pp. 1131–1142, Oct. 2004.

[90] G. P. Bierwagen and D. E. Tallman, “Choice and measurement of crucial aircraft coatings system properties,” in *Progress in Organic Coatings*, vol. 41, no. 4, pp. 201–216, May 2001.

[91] Dassault Systèmes Simulia Corp., “Abaqus Analysis User’s Manual,” Dassault Systèmes, Providence, RI.

[92] “Abaqus 6.13 Documentation.” [Online]. Available: http://dsk.ippt.pan.pl/docs/abaqus/v6.13/index.html. [Accessed: 14-May-2019].

[93] Michael Bauccio, “American Society of Materials (ASM) Metals Reference Book,”, 3rd Edition, ASM International, 1993.

[94] T. Benson, “FoilSim III.”, National Aeronautics and Space Administration (NASA), 2014.

[95] “ATtiny25 - 8-bit AVR Microcontrollers.” [Online]. Available: https://www.microchip.com/wwwproducts/en/ATtiny25. [Accessed: 14-May-2019].

[96] “ATtiny45 - 8-bit AVR Microcontrollers.” [Online]. Available: https://www.microchip.com/wwwproducts/en/ATtiny45. [Accessed: 14-May-2019].

[97] “ATtiny85 - 8-bit AVR Microcontrollers.” [Online]. Available: https://www.microchip.com/wwwproducts/en/ATtiny85. [Accessed: 14-May-2019].

[98] S. R. Anton and D. J. Inman, “Vibration energy harvesting for unmanned aerial vehicles,” in *Active and Passive Smart Structures and Integrated Systems 2008*, 2008, vol. 6928, p. 692824.

[99] “Piezoelectric Sheets & Plates | PIEZO.COM.” [Online]. Available: https://piezo.com/collections/piezo-sheets-plates?_=pf&pf_t_quantity=Quantity__1. [Accessed: 14-May-2019].

[100] L. N. Kanal, “Perceptron,” in *Encyclopedia of Computer Science*, Chichester, UK, John Wiley and Sons Ltd., pp. 1383–1385.

[101] S. Das and S. Biswas, “A scalable pulse protocol for structural health monitoring,” in *Proceedings of the 10th International Conference on Communication Systems and Networks (COMSNETS)*, Jan. 2018, pp. 206 - 213.

[102] D. Feng, F. Hajiaghajani, S. Das, and S. Biswas, “Pulse position coded PDUs: A new approach to networking energy economy,” in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017, pp. 498–503.

[103] J. Liu, P. Cheung, F. Zhao, and L. Guibas, “A Dual-space Approach to Tracking and Sensor Management in Wireless Sensor Networks,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, New York, NY, USA, 2002, pp. 131–139.

[104] G. Simon *et al.*, “Sensor Network-based Countersniper System,” in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2004, pp. 1–12.

[105] L. Yu, N. Wang, and X. Meng, “Real-time forest fire detection with wireless sensor networks,” in *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, 2005, vol. 2, pp. 1214–1217.

[106] J. Vitola, F. Pozo, D. A. Tibaduiza, and M. Anaya, “Distributed Piezoelectric Sensor System for Damage Identification in Structures Subjected to Temperature Changes,” in *Sensors (Basel)*, vol. 17, no. 6, May 2017.

[107] “difflib — Helpers for computing deltas — Python 3.8.0 documentation.” [Online]. Available: https://docs.python.org/3/library/difflib.html. [Accessed: 13-Nov-2019].

[108] J. M. Eppler, M. Helias, E. Muller, M. Diesmann, and M.-O. Gewaltig, “PyNEST: a convenient interface to the NEST simulator,” in *Front. Neuroinform.*, vol. 2, 2009.

[109] A. Cammarano, C. Petrioli, and D. Spenza, “Pro-Energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks,” in *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, 2012, pp. 75–83.

[110] S. Kumar, L. Hussain, S. Banarjee, and M. Reza, “Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster,” in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 2018, pp. 1–4.