DESIGN, IMPLEMENTATION, AND EVALUATION OF NAPALI: A NOVEL DISTRIBUTED SENSOR NETWORK FOR IMPROVED POWER QUALITY MONITORING.

A THESIS SUBMITTED TO MY COMMITTEE
IN FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

MARCH 2020

By

Sergey Negrashov

Thesis Committee:

Philip Johnson, Chairperson
Peter-Michael Seidel
Edo Biagioni
Lipyeow Lim
Matthias Fripp

Keywords: Sensor Network, Distributed Sensing, Grid Computing, Power Quality

ABSTRACT

Today's big data world heavily relies upon providing precise, timely, and actionable intelligence, while being burdened by the ever increasing need for data cleaning and preprocessing. While in the case of ingesting large quantity of unstructured data this problem is unavoidable, when it comes to sensor networks built for a specific purpose, such as anomaly detection, some of that computation can be moved to the edge of the network. This thesis concerns the special case of sensor networks tailored for monitoring the power grid for anomalous behavior. These networks monitor power delivery infrastructure with the intent of finding deviations from the nominal steady state, across multiple geographical locations. Aforementioned deviations, known as power quality anomalies, may originate, and be localized to the location of the sensor, or may affect a sizable portion of the power grid. The difficulty of evaluating the extent of a power quality anomaly stems directly from their short temporal and variable geographical impact. I present a novel distributed power quality monitoring system called Napali which relies on extracted metrics from individual meters and their temporal locality in order to intelligently detect anomalies and extract raw data within temporal window and geographical areas of interest.

The claims of this thesis are that Napali outperforms existing power quality monitoring gridwide event detection methods in resource utilization and sensitivity. Furthermore, Napali residential monitoring is capable of power grid monitoring without deployment on the high voltage transmission lines. Final claim of this thesis is that Napali capability of extracting portions of the events which did not pass the critical thresholds used in other detection methods allows for better localization of power quality disturbances. Napali claim validation was performed through deployment at the University of Hawaii. Fifteen OPQ Box devices, designed specifically to operate with Napali were located in various locations on campus. Data collected from these monitors was compared with smart meters already deployed across the University. Additionally, Napali was compared with standard methods of power quality event detection running along side the Napali systems.

Napali methodology outperformed the standard methods of power quality monitoring in resource consumption, event quality and sensitivity. Additionally, I was able to validate that residential utility monitoring is capable of event detection and localization without monitoring higher levels of the power grid hierarchy. Finally, as a demonstration of Napali capabilities, I showed how data collected by my framework can be used to partition the power delivery infrastructure without prior knowledge of the power grid topology.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Power quality research is a subset of power distribution research which focuses on studying deviations from nominal power grid operating conditions. Devices connected to the power grid, as well as the distribution equipment, expect a certain frequency, voltage and harmonic content of the voltage waveform they operate on. While most equipment maintains some operational hysteresis with respect to deviations from the nominal, large enough deviations may cause equipment failure and instability in the power grid as a whole. In a practical sense, power quality monitoring concerns itself with monitoring, collecting and analyzing power quality anomalies on a live and functioning grid. In some cases, for example when performed by the utility, this information is used to make real-time decisions, to maintain the stability of the power grid. However, data collected by power quality monitoring equipment can also be used to diagnose local power quality problems, or to further power generation and delivery research. For example, power quality data can be very useful in understanding issues with the design and implementation of"smart" grids which incorporate large amounts of distributed, intermittent power generation.

Power quality monitoring fits very well into the paradigm of remote sensing and sensor networks, particularly into the newly emerging field of edge computing. Edge computing goes beyond the naive approach of transmitting the entirety of the collected data from the sensor location, and extends it by either feature extracting, preprocessing or filtering the data at the computing node itself. This research project is centered around the design, implementation, and evaluation of a novel edge computing architecture called Napali which combines feature extraction at the edge level and two way communication between the sink and the edge node. I evaluated Napali in part by implementing it in the power quality monitoring domain.

1.1 Overview of power grids

Modern power grids are hierarchically structured. Higher voltage is useful for transporting electricity over long distances, connecting cities and towns to power generation facilities. Transmissions lines of 100kV and above are used to minimize losses in long distance runs, since the same amount of power can be transmitted using much lower current, and thus much more efficiently, than the comparable low voltage line. Close to the point of distribution, transmission voltage is stepped down to 1kV-40kV range using large power transformers. This is done because the losses incurred in the final leg of transmission are minimal, while extremely high voltage equipment is expensive and requires special precautions.[35] Finally, at the consumer level the voltage level is stepped down once more to the household voltage, for example $120V_{ac}$ for North America. It is important to note that voltage across every part of the power grid is synchronized to a phase and frequency set by the utility. This allows multiple power producers to contribute to electricity gen-

eration without interfering with each other.[7] In North America the 60Hz utility frequency is used as the baseline, and its long term stability is guaranteed by the power company. How close the power AC frequency is to the nominal value is a measure of how closely the electricity demand is balanced by the electricity generation.

Traditional power generation sources involves applying mechanical torque to an alternating current generator. If the load on the generator increases without increasing the torque, it will slow down the generator and thus the utility frequency decreases. Similarly, if the demand drops but the torque is not decreased, the frequency of generated power will increase. Even small deviations in frequency can have adverse effects on equipment which runs synchronous to the power grid, such as synchronous electric motors and other industrial equipment.[25] Nonlinear loads, or loads that don't draw a consistent amount of current through out an AC cycle, are highly prevalent in today's power grid. These devices contribute to the harmonic noise in power system in both current and voltage waveform. This effect, known as harmonic distortion, can have various unintended consequences on the power distribution system and connected devices. The current harmonic distortion affects the efficiency of the distribution network, while voltage harmonic distortion may propagate across the power distribution infrastructure and affect neighboring devices.[26] Distributed renewable generation may also create unintended harmonics. Distributed generators are commonly DC systems, which utilize inverters to generate in-phase AC waveform to feed into the power grid. Depending on the inverter design, the AC waveform may have spurious harmonics present.[25]

Large and sudden changes in load-to-generation ratio do not immediately impact frequency due to the rotational inertia of large generation systems. Instead it will cause the line voltage to change proportional to the load until the generation can catch up. If the load suddenly increases, caused for example by a large motor stall, grid voltage will experience a sharp drop, known as a sag. Similarly a large load drop will cause an voltage increase, called a swell. Voltage sags and swells propagate throughout the entire grid infrastructure, however the dynamics of the power grid are quite complex, and hard to predict. For example a voltage sag on one sub-transmission chain may manifest and as a voltage swell in another.[17] Finally very fast changes in load, such as short circuits, opening and closing of re-closers, and lightning strikes manifest as voltage transients. Voltage transients are energetic short-lived swells on the order of a single AC cycle, which can travel across the distribution grid. Transients may interfere with sensitive grid connected equipment, as well as trigger protection equipment such as uninterpretable power supplies, and other over-voltage protection devices. Transients, harmonic distortion, and RMS fluctuations and their combinations make up the majority of power quality problems which affect the voltage waveform in the grid connected devices. [3] All of these issues can cause power quality problems, as will be discussed further in Section 1.4.

1.2 Edge computing approach to anomaly detection

Edge computing is an emergent field in distributed systems. Edge computing is a consequence of ever decreasing power consumption of computational devices found on the sensor nodes, as well as incremental improvements in battery technology. With ever-increasing computational capabilities in sensor networks, it becomes possible to process and store the acquired data on the device itself, as opposed to the centralized sink. Thus the idea of edge computing leverages available computing power of the sensor node to allow for smarter distributed sensing. Edge computing with respect to remote sensing allows for several new approaches to anomaly detection.

Anomaly detection is a common topic across many disciplines and domains. In cyber-security research, anomalous network traffic and program behavior is often indicative of malicious behavior. In seismic monitoring, anomalies in ground vibrations may be precursors to an earthquake or a volcano eruption. In observational astronomy, anomaly detection is used for detection of transient events such as gamma ray bursts. Sensor networks are commonly tasked with anomaly detection and must often act on them. Traditionally, stringent constraints on power consumption of battery powered wireless sensor network nodes mean that low bandwidth and low complexity methods are preferred. Furthermore, many sensor networks are often hindered by local noise, thus requiring higher level filtering in order and in network processing to determine if an anomaly has occurred. If the signal to noise of the local measurements is quite high this problem becomes trivial: one simply collects all the distributed measurements if one or more of the measurements indicates an anomaly. Unfortunately, in the real world such problems are rare and instead the distributed signal is dominated by extraneous local noise. For example individual seismic sensors can't distinguish between a global anomaly such as an earthquake and local noise such as vibration caused by a passing semi-truck.

The problem of global anomaly detection with distributed sensing has been explored in self-organizing wireless sensor networks. However, these approaches are insufficient when applied to edge computing. Edge computing relies on Internet for transport, and thus the cost of communicating with the local sink and the local node is similar. Indeed in some cases it is impossible to achieve node-to-node communication without an intermediary due to firewalls, and other security mechanisms. In this research, I only consider approaches which rely on a sink node to facilitate anomaly detection.

One important problem in Internet-enabled sensor networks is called the "local noise problem". In many cases, the local noise presents itself with a similar signature as anomaly in question. Only through consensus of multiple devices, is it possible to separate local noise from true system-wide phenomenon. There are several solutions for dealing with the local noise problem in an Internet-enabled sensor network. A naive solution is to simply transmit every distributed measurement to a centralized data sink. This sink, as well as the infrastructure down stream of it will have a view of the entire state of the system and can thus detect anomalies using either real-time

or batch processing. An alternative to the naive solution is to let the individual sensors decide which temporal regions of the measurement constitute an anomaly. This approach, called self-triggering, has the benefit of the reducing the bandwidth constraints for each sensor without the requirement for two way communication between the sensor and the sink. On the other hand, the self-triggering has significant drawbacks. Local nodes may miss event which fall outside of their detection thresholds, resulting in an incomplete dataset for the system-wide anomaly. Furthermore, self-triggered detection, will incur a false positives due to the local noise effects.

1.3 Napali: hybrid edge computing for anomaly detection.

In this thesis I present, a framework called Napali which combines the strengths of the previously mentioned methods. In Napali, each sensor node maintains two way communication channel with the sink, as well as a temporal window containing all the recent data it collected. Each sensor's on board processing is used to extract features from the collected measurements, and initially only these features, instead of the entire measurement set, are forwarded to the sink for processing. The sink thus acquires a low resolution view of the state of the entire sensor network. While this may not be enough for rigorous anomaly analysis and classification, properly selected features(metrics) from every node should be enough to detect the occurrence of an anomaly. Finally, if an anomaly is detected, the sink can requests high resolution data from all of the affected devices.

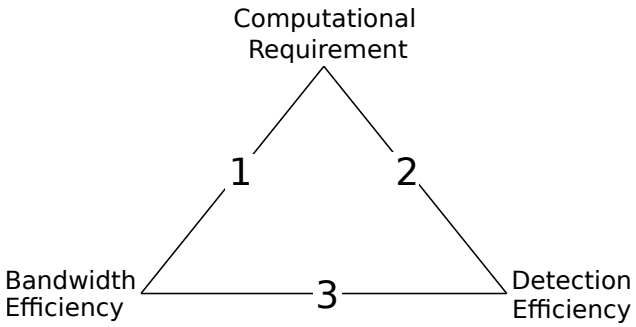


Figure 1.1: Comparison of the three event detection methodology across three metrics. Methods are as follows: naive method (2), self triggering (1), Napali, hybrid solution (3)

Figure 1.1 illustrates the strengths and weaknesses of the three approaches to anomaly detection. The naive method provides the best detection ability and the smallest node computation requirement. However, it does so at the cost of the largest bandwidth consumption. The self triggering method has the lowest bandwidth consumption of the three. The disadvantage of this method is twofold. First, in order to maintain a high detection rate, a reasonably low threshold for anomalies has to be used. This may cause a large number of false positives due to local noise and sensor noise. Second, global anomalies will often diminish in magnitude as a function of distance from the epicenter, thus far removed sensor nodes may completely miss events. These low threshold

events may be invaluable for reconstructing the dynamic of the anomaly propagation, however they will be missed by the detection algorithms.

Napali's hybrid approach provides the opportunity for much better anomaly detection rate and background noise rejection, by correlating the features that are computed on the sensor nodes. Napali has moderate bandwidth consumption, however the bandwidth consumption is tunable since the features can be computed for varied temporal windows, the length of which can be adjusted in real time. Napali requires the highest sensor node computing power, since not only does it need to extract the triggering features from from raw data, it needs to maintain a buffer of sensor measurements to send to the sink if an anomaly is detected.

The main operational downside of the Napali framework, is the requirement for two way communication between the sink and the sensor node. In order to participate in event detection Napali sensors need to be able to both send raw and feature extracted data, as well as receive the control signals from the sink. In contrast, the naive and self triggered approaches only require a one way link to send the raw data to the sink. As such, standalone devices can be retrofired to participate in distributed event detection in the naive and self triggered approaches, while Napali requires custom devices specifically tailored for cooperative event detection.

The advantages of the Napali methodology compared to the naive and self triggered approaches are enumerated below:

1. **Bandwidth usage is minimized:** Instead of sending the entirety of raw data, only extracted features are sent. This features will have a tiny fraction of the bandwidth requirement when compared to raw waveforms. Furthermore, the temporal window which encompasses a single feature can be adjusted in real time. Thus as soon as an anomalous behavior is observed in a subset of sensors, this window can be adjusted for a finer grained feature extraction.

2. **Effects of latency are minimized:** Even at 1Msample/second at 16bits of resolution, the memory requirement to store 5 minutes of raw waveform without compression are on the order of 512MB, which is well within the realm of cheap single board computers. With compression specifically suited to the signal of interest, the memory requirement can be reduced even further. This gives the triggering stream sink plenty of time to respond to the anomalies in the data and request raw waveform from the monitoring devices.

3. **Sink processing requirements are minimized:** Since most of the feature extraction is already performed at the device level the triggering stream sink computational resources can be minimized. With the advent of IOT, the computational capacity of the edge devices is increasing. Napali exploits these resources, and thus minimizes the sink computational requirements.

4. **Sub-threshold data acquisition is a cost-effective way to improve understanding of grid-local anomalies:** The triggering stream sink makes the decision to acquire raw

waveform from sensor nodes. This allows researchers to collect data from devices which were only mildly affected or not affected by the disturbance. This provides new possibilities for investigation of disturbance propagation across the sensed area. Furthermore, subthreshold data acquisition from leaf node allows for monitoring of the internal system state.

5. **Increased resiliency with respect to power failure:** In the case of the complete power failure or communication blackout, if the monitoring device has a battery backup capability, each sensor has a record of the entire raw waveform leading up to the power interruption. Prior to shutdown the sensor node will transfer all of the raw data from the volatile memory to on-board permanent storage. Once the power or communication channel is restored, select portions of the buffer may be sent back to the data sink. This creates an additional layer of resiliency for the anomaly detection network.

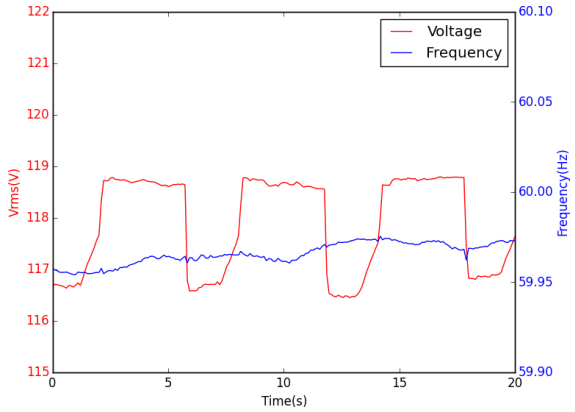
6. **Increased flexibility with respect to privacy protection policies:** Anomalies which were only observed at a single point are most likely local noise and pose little value for global state monitoring. IIt can be up to the owner of each sensor node to decide how to process disturbances which affect their sensor. For example, a node owner may choose to record a full waveform, only certain features, or record nothing at all. Secondly, if the saturation of the device is high enough only a subset of them would need to send a triggering stream for event detection, while the rest will be used for acquiring raw waveform for small temporal regions containing global events.

7. **Temporal locality allows Napali to provide improved insights into power quality anomalies over traditional triggering algorithms:** By exploiting the temporal locality it is possible to ascertain the geographical extent of the anomaly with only coarse features. This allows for a simple robust algorithm which may be deployed at the sink node for anomaly detection.

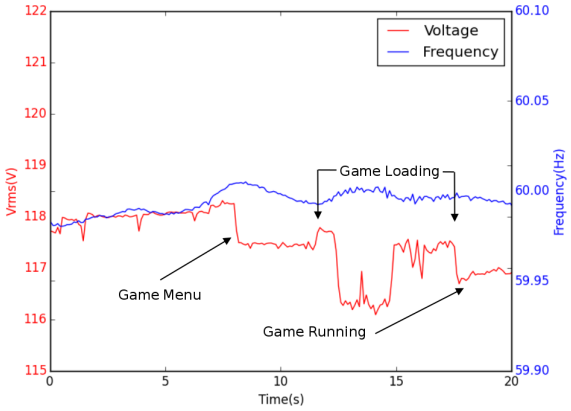
1.4 The problem of power quality

Power quality monitoring, along with other smart grid domains is a field well suited for distributed sensor network monitoring.[24] [30] As the power grid moves from centralized generation with a few centralized power plants to distributed generation with residential power production, a distributed consumer level monitoring system becomes ever more important. Traditionally, utilities do not monitor the quality of power besides the consumption beyond the substation level. This is due to the fact that the last opportunity that the utility has to regulate and filter the grid voltage in the hierarchical power distribution is at the substation, or neighborhood level. This methodology is unsustainable, as residential renewable energy production if not properly monitored and controlled will have an adverse effect on overall grid stability. Furthermore, lack of residential monitoring

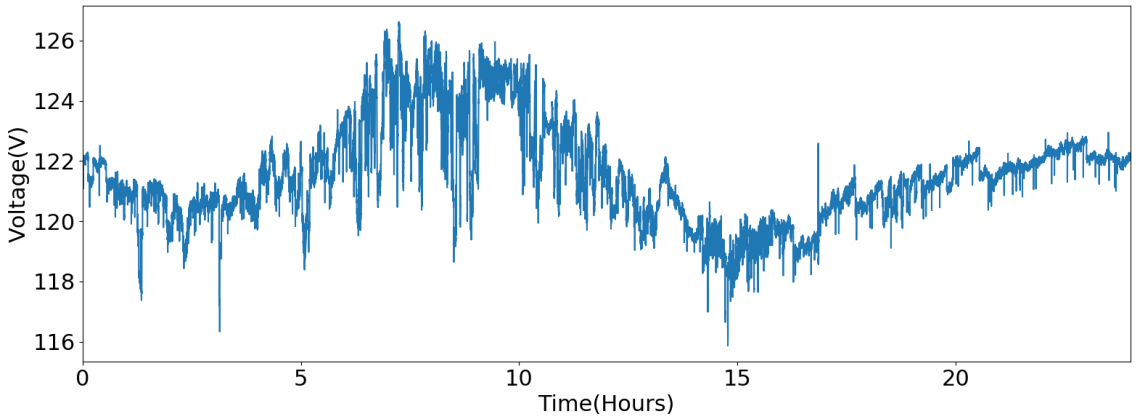
may lead to dangerous conditions such as islanding, where an otherwise powered-down grid may have a small powered island sustained only by the unmonitored residential renewable generation. Finally, residential power quality monitoring gives utility costumers an opportunity to evaluate the quality of power delivered to their household. As consumer electronics are becoming more and more complex, they become more sensitive to the power anomalies. Grid monitoring is traditionally the responsibility of the utility, however in most cases utilities only have to disclose power interruptions lasting several minutes. Small interruptions, and partial interruptions such as voltage sags, swells, frequency fluctuations and transients will often go undisclosed by the utility and unnoticed by the consumer, but may cause premature failure in grid connected electronic devices. It is in the best interest of the consumers to monitor the quality of the power that is delivered to them, meanwhile the same same monitoring system will also allows researchers and utilities to monitor the entirety of smart grid.



(a) Voltage signal produced by a hotplate.



(b) Voltage signal produced by a desktop PC running a complex task



(c) Line voltage recorded over 24 hours in a residential household with photovoltaic installation.

Figure 1.2: v_{rms} waveform generated form the consumer side of the meter under various conditions. All waveform were recorded using the OPQ Box 2.

Residential power quality monitoring presents a number of issues, as illustrated in Figure 1.2.

First of all, it is difficult to discern which "side" of the utility meter a power disturbance came from. Is the origin of the disturbance within a building and flow outward through the meter into the grid, or did it originate elsewhere in the grid and flow into the building through the meter? Consider a voltage sag generated by a 1kW hot plate shown in Figure 1.2a. Since the output impedance of the power grid is non-zero, a high powered device can cause a significant voltage sag affecting every device connected to the same circuit. Second, recoding the voltage waveform resulting from non-linear load can result in privacy leaks for the end user. Consider the voltage waveform produced by a PC running a video game as shown in Figure 1.2b. Throughout the game loading process, the power load varies based on which components are in use. Furthermore, regions with CPU load, hard disk load and video card load can be readily identified by measuring the resulting voltage sag. Recording these event's has an adverse effect on end-users privacy and offers no immediate benefit in studying grid stability. Finally distributed power generation such as rooftop solar has significant effect on the residential voltage waveform. Consider the voltage waveform shown in Figure 1.2c. This waveform was recorded over 24 hours in a household with a rooftop photovoltaic installation. Similar to the voltage sag case, since the power grid impedance is non-zero residential power generation will cause a voltage swell during peak hours of sunlight as evident by the waveform. Combined with the global voltage sag during hours of peak demand (3pm) combined with the lack of PV production during that time, photovoltaics installations can result in a daily $10V_{rms}$ swing. Residential power quality monitoring can be accomplished via a distributed sensor network made up of power quality monitoring devices with high degree of penetration across the end points of a power grid. Furthermore, in order to monitor the dynamics of the entire power grid via residential line voltage monitoring, it is imperative to monitor across multiple locations simultaneously. This combined with temporal and spatial correlations of data produced by the sensors provides the potential for identification of grid-wide anomalies without a high rate of false positives. Additionally, not all power quality events affect the entire grid, due to the hierarchical structure of the power distribution.

1.5 An edge computing approach to power quality monitoring.

IEEE1159 standard describes the techniques for single location power quality monitoring. For transient monitoring it suggests a sampling rate of at least 7680 samples/second, up to 1 Megasample/second. This implies that if a power quality event detection system relies on raw data from all monitors((i.e. the naive method)) it would do so at a very large bandwidth cost. At 20 Ksamples/second at 16bit per samples a single monitoring device would generate 300Kb/second. Several thousand devices could easily saturate at 1Gb link with no obvious benefit. Collecting and recording all of the raw waveform data from residential power quality meters for batch processing presents some privacy concerns as outlined above. On the other hand, an on board event detection methodology, allows the measurement devices to select which temporal regions of measurements constitute

an event. This is a perfect strategy from the consumer's perspective, since it would allow for recording of power quality information which directly impact their residence. As mentioned in Section 1.2, this method relies on a threshold based approach where every device has a computes several metrics from the raw waveform and compares than to preprogrammed threshold values. Metrics such as V_{rms}, fundamental frequency and THD are easily adapted for single point power quality measurements. Temporal regions during which these metrics surpass a threshold are considered by the device as a power quality event, and thus are recorded.

The problem with the self-triggered method is that grid-wide power quality events do not affect the entire grid in the same way. For example, due to the grid's hierarchical structure, a voltage sag on one sub-circuit can manifest as as a sag of a different magnitude or even a swell on another.[17] This may result in a situation where some of the monitoring devices will not consider a power quality anomaly as an event, because it did not surpass the metric threshold, and simply ignore it. From the research perspective, however, it is important to get raw data from all of the affected devices not just, the ones that were the most affected. This additional information could be used to localize the disturbance, as well as better evaluate it's impact. This makes a hybrid centralized/decentralized event acquisition strategy more attractive for distributed residential power quality monitoring. In this scheme all monitoring devices use local processing resources to feature extract the incoming waveforms while storing them locally for several minutes. A sink collects all of the extracted features and looks for anomalies which are present in the feature data stream which we will refer to as the triggering stream. If an anomaly is present in only a single device, it is highly probable that the disturbance occurred in the residence. Depending on the user's privacy preference, raw data for a single device anomaly can be be recorded for later analysis, or in case of a highly privacy conscious user, ignored. On the other hand, if the triggering stream shows an anomaly temporally collocated across multiple devices, the entire network or a subset of the network may be queried for raw waveform data for a temporal region which corresponds to the disturbance in the triggering stream.

The main disadvantage of this method is that while there are plenty of power quality event detection methodologies for single location, there has been little research into the distributed event detection methods and metrics. The two problems are quite similar, indeed one may use the same metrics for distributed event detection as with the single point power quality monitoring. However, it's also important to consider temporal locality of anomalies detected across multiple devices as well as the effects of device synchronization. Power quality anomalies such as voltage sags and transients will propagate through the transmission lines at the speed of light, however due to the non-linear elements which make up the power-line junctions, a certain temporal spread in event detection across multiple locations is expected. For large power grids such continental United States grids, large frequency fluctuations propagate in a highly nonlinear ways. In these cases the event propagation is limited by the inherent rotational inertia of the power generation systems,

and the speed at which the grid protection elements such as reclosers and circuit breakers operate. Regardless, the closer local anomalies are detected in time, the more likely are they to be a result of a gridwide event. Unfortunately, it is unfeasible to perfectly temporally synchronize the distributed power quality monitors. While methods such as GPS can in principle provide synchronization of up to 10ns jitter across a large geographical region, they require a line of sight to the sky, and add a non-trivial cost to the bill of materials for every power quality meter. Furthermore, GPS is prone to losing signal depending on atmospheric conditions, and can be very sensitive to fluctuations in the power supply voltage, a critical time in power anomaly detection. An alternative to GPS is Network Time Protocol. Network time protocol can provide timing synchronization on the order of $10ms$ across Internet, which is on the order of $\frac{1}{2}$ of a grid cycle. NTP performance could be further improved by using geographically close time servers which are themselves synchronized via GPS. Consider a situation where two devices are located in household which experience a local $100ms$ power quality disturbance every 10 minutes. Even with a $10ms$ synchronization jitter, it will take on average 21.5 days before the two disturbances are observed within $20ms$ of each other. If a third device is introduced, it is highly unlikely that all 3 would observe unrelated local anomalies within $20ms$ of each other over the lifetime of the power quality monitoring network. This implies that combination of temporal and threshold based correlation on the feature extraction data would allow one to build a robust residential based power grid monitoring system which would yield a very low rate of false positives.

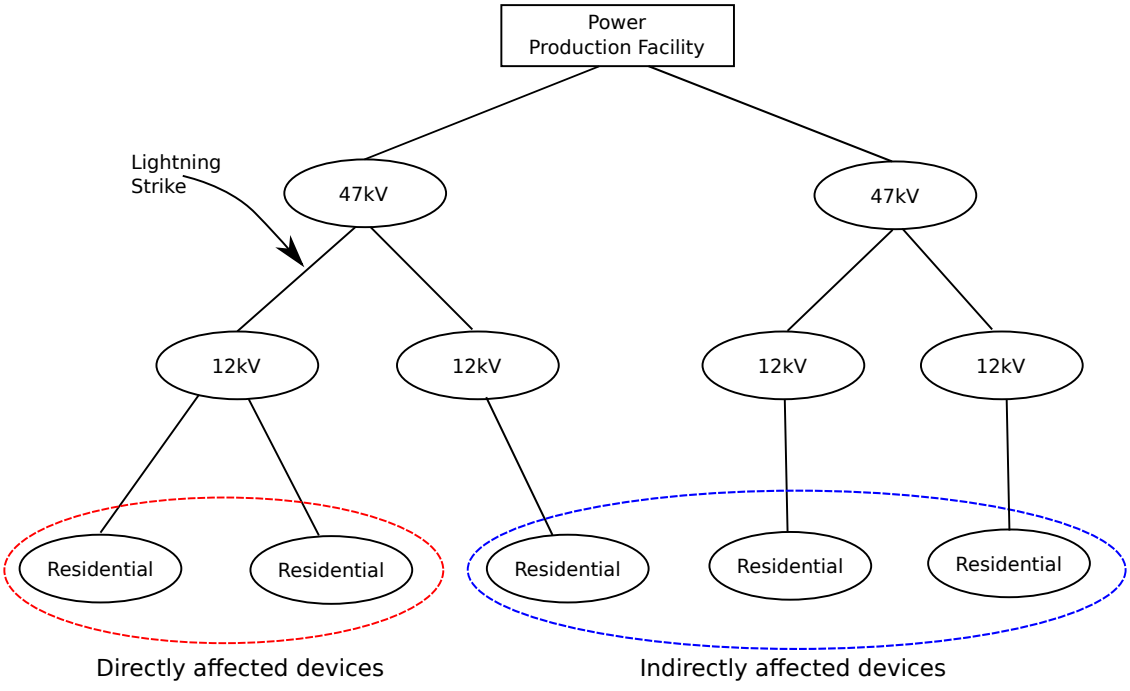


Figure 1.3: Power quality anomaly propagation example.

1.6 Thesis claim and evaluation

Today's big data world is plagued with issues of data cleaning and validation, even though it's being increasingly relied on for timely, accurate and actionable intelligence. With large ingress of unstructured data these issues are unavoidable, and preprocessing will remain a large portion of the analysis workload. However, in the case of sensor networks designed for a specific purpose, the tasks of anomaly detection can be pushed to the edge of the network using the Napali methodology.

The claim of this thesis is that Napali provides a novel architecture that is both a feasible solution to the problem of distributed power quality monitoring and provides significant benefits over the two standard alternative architectures (the self-triggered method, where all computation/storage is performed at the edge nodes, and the naive method, where all computation/storage at the occurs at the sink). Furthermore, the Napali architecture can, in principle, provide benefits for other domains beyond power quality.

In order to evaluate the claim of this thesis, I implemented it as part of the Open Power Quality(OPQ) system and applied it to the problems of power quality monitoring. Combined with higher level anomaly analysis, Napali provides important services for Open Power Quality power quality monitoring network. This network is made up of a group of monitoring devices as well as a centralized data sink server. This system was deployed for testing at the University of Hawaii at Manoa campus, by deploying power quality monitors across 15 University buildings. The University of Hawaii at Manoa campus is a unique testbed for such a network, since the entire campus is isolated to its own microgrid, connected to the municipal Oahu grid via a 46kV feeder. Furthermore, the University of Hawaii has deployed a set of smart power monitors at the key positions in the grid, which was used as a state-of-the-art ground truth for evaluation of OPQ performance.

The OPQ system relies on a custom residential power quality monitor called OPQ Box, designed specifically for distributed monitoring using the Napali framework. Instead of performing local analysis on the voltage waveform with the aim of PQ anomaly detection, or forwarding all the recording measurements to the centralized sink, OPQ boxes computes a small subset of features on the input voltage waveform. These features are then forwarded to the Napali framework's centralized sink which performs the anomaly detection on reduced data, while the raw waveform is retained for a short time on the OPQ Box. If the sink determines that a possible anomaly has occurred, a request is sent to the affected and nearby devices for raw data.

The goal of Napali is not to provide a low rate of false positives for a particular type of a power quality disturbance. Indeed, once the raw data is acquired by the sink, filtering through the potential anomalies is trivial using well established methods. Instead, the focus of Napali is balancing the low bandwidth required for detection with a low rate of false negatives. Furthermore,

monitoring at the leaf notes relies on the hierarchical nature of the power grid in order to ascertain the state of the entire power generation and delivery system. As noted in the literature, power quality disturbances tend to propagate down the hierarchy as shown in Figure 1.3.

Consider a lightning strike on a hypothetical 12kV feeder line in a hierarchical power grid. The directly affected devices will be the ones downstream from the disturbance. These devices will experience the most severe effects, most notably transients, as they propagate throughput the power delivery infrastructure. The indirectly affected devices will expedience a power anomaly mainly attributed to the power production entities trying to compensate for the large disturbance caused by the lightning strike. Thus, monitoring of the leaf nodes of the power delivery system can in principle provide insights into the disturbances that originate deep inside the power distribution network.

The Open Power Quality system is designed to be a test bed for development of new power quality detection and analysis algorithms. It can facilitate development of new techniques and methods for studying power system, by utilizing the Napali framework as the main anomaly detection methodology. I assessed the data collected by the OPQ network at the University of Hawaii in order to determine if the claimed benefits of the architecture are observed in practice.

To reiterate: the central claim of this thesis is that Napali provides a feasible solution with significant benefits. I will provide evidence for this central claim by evaluating the following subclaims about Napali:

1.6.1 Napali minimizes bandwidth usage

Bandwidth consumption of the OPQ system was carefully monitored, recorded and compared to the bandwidth required to transmit the equivalent amount of raw data. Furthermore, a Self-Triggered data acquisition ran along side the Napali event detector. Thus, all three methods were compared in bandwidth utilization. A more detailed description of this evaluation is found in Section 4.3.1.

1.6.2 Napali mitigates device latency effects

I examined the latency limits of the triggering system. Latency effects on the Naive and Napali event detection methodologies were evaluated. Since these limits are heavily dependent on the raw data storage ability of the OPQ Box, latency effects were tested under various amounts of memory allocated for this task. A more detailed description of this evaluation is found in Section 4.3.3.

1.6.3 Napali minimizes sink processing requirements

Synthetic benchmarks were carried out on the sink node to determine the scalability of the triggering system. These scalability metrics were compared with a synthetic benchmark of running

multiple copies of the OPQ Box analysis software on the same node. This allowed me to compare the scalability of the sink node in the case of sending the entirety of raw data stream versus the Napali framework's approach of only sending extracted metrics. A more detailed description of this evaluation is found in Section 4.3.2.

1.6.4 Napali temporal locality triggering results in a low false negative detection

Data acquired from the UH building level meters was compared with the data acquired via the Napali triggering framework. This allowed me to establish the rate of false negatives and evaluate the temporal locality triggering algorithm. A more detailed description of this evaluation is found in Section 4.3.5.

1.6.5 Sub-threshold data acquisition is a viable event detection strategy

I examined a subset of events that can only be detected via the subtheshold triggering. This was performed by analysing UH meter data, searching for events that Napali could only detect via subthreshold detection. This provided a baseline comparison of Napali with a commercially deployed system. A more detailed description of this evaluation is found in Section 4.3.6.

1.6.6 Napali failure resiliency and flexible privacy

Although power failure resiliency and flexible privacy are claimed benefits of the Napali architecture, they were not evaluated as part of this thesis research. Flexible privacy required a much larger deployment, and a user study, which was beyond the scope of this project. Evaluating the power failure resilience of the Napali framework would require a significant development effort for the battery management system. Since complete power failures are quite rare, there is no guarantee that a single power outage would occur on the UH campus during the deployment.

1.6.7 Napali in other domains

A final part of the central claim of this dissertation is that the Napali architecture can be applied to other domains. Once the Napali framework was fully characterized, and its strengths and weaknesses were well understood, I performed a literature review of other domains which could benefit from Napali-like approach to event detection. I further characterized the kinds of design changes to existing sensors that the Napali Framework required in order to apply it to these domains. This work is can be found in Section 5.1.

CHAPTER 2
RELATED WORK

2.1 Edge computing

Projections performed by Forbes suggest that by 2025, more than 75 billion Internet of Things(IOT) devices will be connected to the Internet.[1] As the devices at the edge become more computationally capable and more numerous, it becomes imperative to share the computational load not only across the cloud services, but across the devices themselves. Furthermore, a large portion of these devices, such as home automation, do not require a connection to the cloud in the first place. Instead they require a connection to the edge IOT hub, or need the cloud service only to establish or broker communication with another IOT device. Edge computing is a subset of IOT research, which concerns itself with distributing the computational load across the devices at the edge of the of the network. [32]

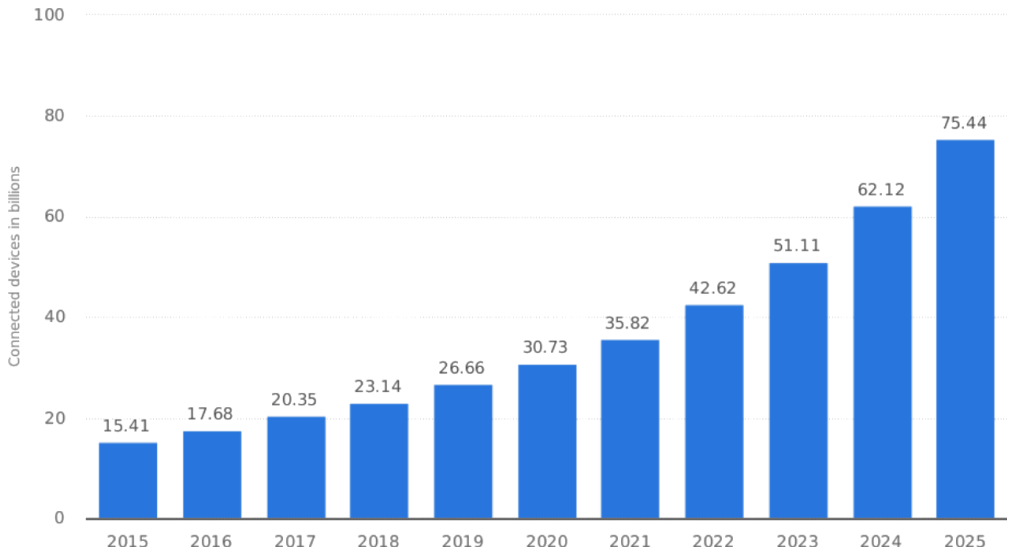


Figure 2.1: Projected number of IOT devices worldwide.[1]

This change in computational strategy may seem inconsequential at first. However, upon deeper reflection it becomes clear that this is a major paradigm shift which brings IOT closer to the sensor network world it is often compared with. While pioneering work in IOT always assumed a one or two-way communication between the IOT device and the cloud service, utilizing TCP/IP as an end-to-end protocol, it is becoming clear that this approach is unsustainable, and is often undesirable. This communication model has clear disadvantages in wasted communication, computation, and privacy. Furthermore, the rigid computation communication model is not flexible enough to support

devices which are beyond the edge of the TCP/IP network without resorting to ad-hoc routing. [13]

The first attempt to address the bandwidth and latency issues arising with widespread IOT adoption came in form of content delivery networks (CDNs).[13] CDNs circumvent the generic cloud information delivery problems by placing transparent caches geographically spread across the application domain as shown in Figure 2.2. When a user or an IOT device makes a request for an object, this request is forwarded to the nearest CDN node for processing. If the node contains the object in its cache it is immediately forwarded to the requestee. Otherwise a request for the object is forwarded to the centralized cloud data store, and returned to the requesting device, as well as placed in the local cache. This approach has the advantage of moving the data closer to the end user, thus reducing latency, and taking advantage of geographical locality. Another advantage of this method is the added resiliency of the CDN architecture to a single point failure. If a local cache node fails, its userbase can be forwarded to another node, although incurring additional latency. Additionally, if a centralized data store becomes unreachable, the local cache nodes can to some extent mask it's outage by forwarding the data available locally. This approach does have some drawbacks. While it makes it easier to enable faster transactions regarding data, it is not trivial to move application logic to the local cache nodes. Furthermore, the CDN methodology, still relies on a central mediator for device communication, even if the devices are located in the same room.

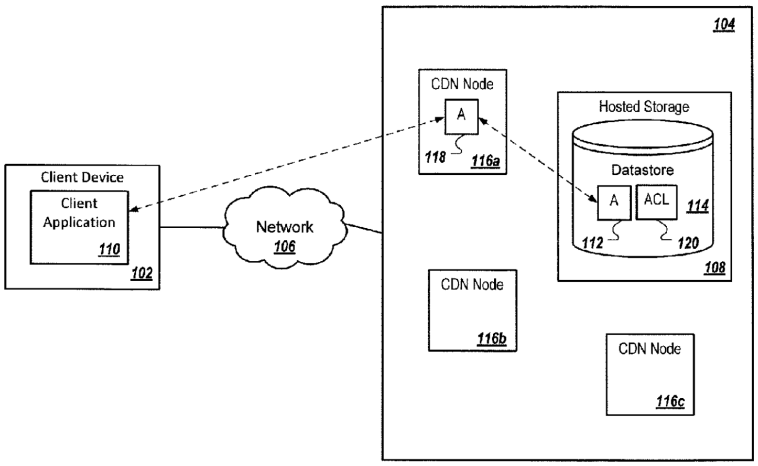


Figure 2.2: Content delivery network architecture. As described in the Google patent.[13]

In an attempt to support a more diverse IOT ecosystem, current research is focused on moving the cloud service ever closer to the edge of the network. Since the majority of IOT devices are located within one hop of the Internet, the next logical place to locate a content provider is at the wireless basestation.[32] These servers, commonly referred to as cloudlets or fog servers, are collocated with various wireless basestations, which allows them to provide a location specific

service to the user without using the Internet. This approach also provides uniform access and simplifies intercommunication between a variety of devices, including those that don't use TCP/IP. Unlike the localized cloud cache approach relied on by the CDNs, fog servers are built with the notion of moving not only data but also the application logic to the edge of the network. To facilitate inter-device communication between the devices using differing wireless protocols, fog servers can no longer rely on TCP/IP routing. Instead, TCP/IP becomes yet another transfer protocol along with Bluetooth, Zigbee, 3g etc, with routing between the devices implemented as a software service.[36] A few use-cases of such technology are already found in industry. Examples of these include airline/bus in-flight entertainment, and shopping mall directory apps. A block diagram of this infrastructure is shown in figure 2.3. In the future, emerging technologies which are sensitive to latency, such as virtual and augmented reality will benefit from fog computing, since it's inherently lower latency than the cloud counterparts.

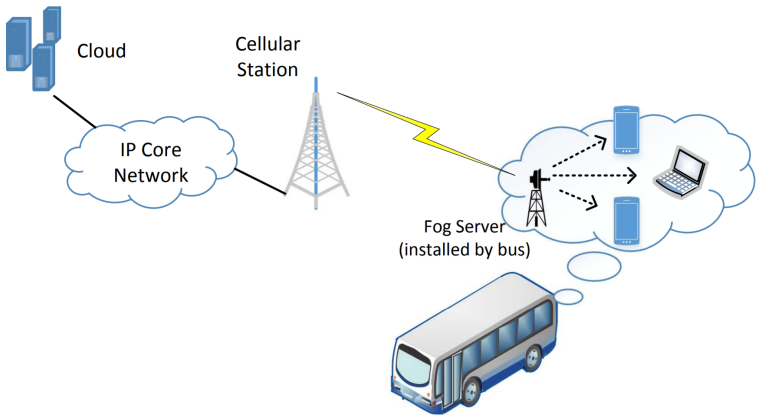


Figure 2.3: Fog computing use in transportation. The bus cloudlet provides a cache for common data such as commuter schedules and traffic information, while routing other queries to the Internet.[36]

With the architecture for low latency communication between the edge devices and the fog provider established, the state of the art in edge computing research is focused on intelligent sharing of the computational resources in the fog system. Edge servers generally have far more computational capacity than the edge device, however, they service many such devices. Additionally, due to the fickle nature of radio links connecting the edge device to the fog service, the work sharing protocol must be able to cope with link and packet loss. Finally, in the case of battery powered devices, the energy cost of transmitting the computational job, and receiving its result may exceed the cost of performing the computation locally. Finally, with mobile edge nodes, such as smart phones, and smart cars, computational offloading algorithms must be able to handle constant network reconfigurations as the edge nodes enter and leave the fog server geographical area. A number of algorithms have been proposed for efficient and robust computational sharing in fog environments. [28] [39] [40]

Napali fits in-between the CDN and Fog server architectures. The power quality disturbances are generally localized to a specific area, so a sink placement which covers a small geographical area is preferred in order to reduce latency and reduce unnecessary communication with the centralized cloud location. Furthermore, sink-driven measurement rate allows the OPQ Box to dynamically scale the computational and communication overhead. Finally the event, classification and analysis are similar to the computational offloading strategies currently under development in the edge computing field.

2.2 Distributed Power Quality Monitoring

Power quality monitoring is a long established field in the smart grid domain. However, the vast majority of research so far has focused on single point power quality monitoring.[34] Such research has extremely useful applications in industry, since it allows one to ascertain the absolute quality of the delivered power at a given location. However, since power quality disturbances can originate both from local sources and from gridwide disturbances, single point monitoring is not particularly useful for smart grid research. Several projects have developed a distributed approach to power quality monitoring, the most prominent being the FNet project and the Power Standard Lab PQube deployment.

The FNet project designed, manufactured and deployed a Phase Measuremnt Unit (PMU), across over 300 locations across the united states.[44] PMU devices plug into an outlet, and sample the power line voltage at the rate of about 1.5kS/s. The sampling is disciplined by GPS, and as such FNet devices are extremely sensitive to voltage frequency and phase angle. The precision of the FNet devices is $0.5mHz$ for frequency and 0.02° for phase angle. Collected data is sent to the collection service at 100ms intervals via the Ethernet connection. Using these devices FNet was able to observe several large power disturbances in the US power grid. The robustness and sensitivity afforded by the GPS receiver makes this project an excellent source of frequency data across large geographical area, however, the sampling rate of 25 samples per grid cycle is far too low to properly sample fast transients and sags. Furthermore, FNet provides no methodology for acquiring raw data for event disturbances which it records.

Power Standards Lab (PSL) has been an industry leader in power quality monitoring, and has authored several standards on the topic. Furthermore, PSL has developed and deployed a large number of power quality monitors called PQube across the world. The exact number of deployed devices is uncertain since a lot of the devices are deployed industrially and are not available to the public. However PSL has several publicly available devices, as well as several PQ datasets accessible for smart grid researchers. PQube devices are an industry standard for power quality monitoring, sampling at $12.8kSps$ for both the voltage and current waveforms.[2] Each PQube device is supplied with a NIST certificate of compliance and complies with the IEC 61000-4-7:2002 standard for PQ measurements. Incidentally, this standard was authored in part by the PSL staff.

Similarly to the FNet PMU, PQube devices are GPS disciplined, additionally the sampling is phase-locked to the voltage waveform allowing for an even more precise metric extraction. Finally each PQube device is configurable with custom thresholds which allow it to record raw PQ event data for the location it's monitoring. PQube offers a centralized data collection option with flexible communication schemes ranging from Ethernet to Cellular. Since PQube devices monitor current in conjunction to voltage, its installation requires it to be placed into the electrical box of the target, by a licensed electrician.[38] Furthermore, the GPS synchronization requires addition of extra conduit to the electrical box to allow for an antenna. Finally, since the PQube devices are designed for single location measurements, distributed event detection using the PQube network is particularly difficult, with a lot of low magnitude gridwide events being incomplete or missing.

Unlike the single point monitoring solutions, the Napali framework is incapable of operating as standalone PQ monitor without a cloud sink. Furthermore, even with the event detection sink, the goal of Napali is to reject local anomalies in order to reduce the communication and computational overhead. While not as sensitive as the PQube device, the deployment price per unit is two orders of magnitude lower, while providing better sensitivity than compared to the FNet device, when running with GPS. The ability of OPQ Box nodes to utilize NTP, with WIFI connectivity, means that the OPQ Box deployment is much simpler than the FNet and PSL offering, without requirements of a clear view of the sky or additional wiring for Ethernet and GPS antennas. Finally, Napali distributed event detection system allows for acquisition of the entirety of the PQ disturbances including in locations where the disturbance has been greatly attenuated by the electrical distance. Thus, Napali is able to provide a more complete picture of the disturbance propagation throughout the smart grid.

2.3 Anomaly detection in Power Quality Monitoring Networks.

Anomaly detection in PQ monitoring networks remains an active topic of research. The goal of PQ event detection is to isolate the temporal regions where the voltage or current waveform deviates from the nominal by a given threshold. In some cases the aim is simply to notify a higher level control system in a realtime manner that a disturbance is taking place. In other cases, the goal is to acquire the raw disturbance data for off-line analysis. Most of the detection methods rely on statistics and thresholding in order to detect PQ disturbances. Most of the literature concerns itself with single point detection, for purposes of protection of equipment downstream.[14][18] [33] Distributed power quality projects will generally utilize single single point detection across multiple devices in order to reconstruct gridwide propagation.[38]

With a wide deployment of smart meters, PQ researchers gain access to a networked platform which is perfectly positioned for PQ monitoring.[15] The major issue for smart meter real time monitoring is bandwidth constraints. Smart meter deployment is envisioned to communicate via a mesh network with a stationary or mobile base station used for data aggregation. As such the band-

width and connectivity is limited, thus requiring methodology which is capable of event detection in such environments. Generalized local likelihood ratio detection is one method for overcoming these limitations. This approach requires only a single bit to be forwarded from each smart meter indicating whether a disturbance is taking place or not. These bits are aggregated at the "master" meter, and if their sum exceeds a threshold a higher level control system is notified of the ongoing disturbance.[22] This approach is resilient to bandwidth limitations, and communication instability, however tuning thresholds for each individual meter requires a significant manual effort.

Systems designed for distributed PQ event detection using custom meters are prevalent in literature. A study at CERN utilized PQube devices with gapless recording which were later analyzed off-line, in order to ascertain the propagation mechanics of PQ events.[17] In a realtime domain Shang Li and Xiaodong Wang extended their work in [22] from smart meters to standalone devices, again advocating for single bit statistical based triggering generated by asynchronous meters.[21] Unfortunately their work has never been verified beyond simulation. The Transimeter project utilized an analog hardware event detector comprised of a high pass filter and a comparator for transient detection. These devices had two data paths for the voltage waveform, one to the National Instruments DAC board, one to the hardware trigger circuit. If a trigger circuit detected a transient, a flag was set on the NI DAC, which would in turn instruct the connected PC to send the data to the central server.[10] Unfortunately, the lack of cooperative detection and an inflexible trigger circuit makes this approach unappealing for modern power quality monitoring. Some of the more exciting work in PQ detection is modeling the most efficient placement of PQ meters in order to provide complete coverage for the power grid. [41] Another is using distributed detection for localization of the event source.[29] [31]

Napali differs from the smart-meter approach in the use of WiFi for communication, which greatly decreases the communication constraints of the system. Since the OPQ Box is always connected to the power grid it is monitoring, power concerns are minimal. This allows Napali to implement more robust computational and communication strategies, not commonly possible with smart meter PQ monitoring. Since the triggering stream is generated in software, it is possible to switch the detection metrics without redesigning new hardware. Napali combines both cooperative PQ event detection and PQ event acquisition which makes it useful for future PQ event localization, and propagation research.[29] [31]

CHAPTER 3
OPEN POWER QUALITY

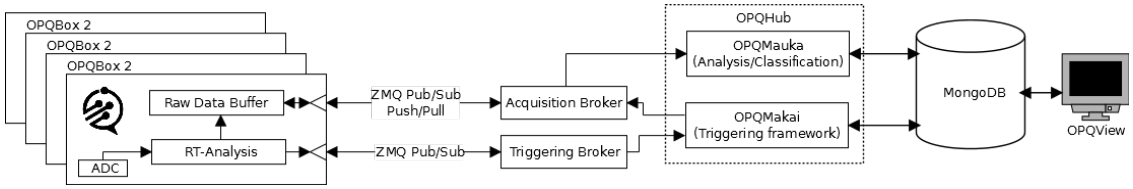


Figure 3.1: Block diagram of the OPQ Power Quality monitoring system.

The Open Power Quality (OPQ) power quality monitoring network utilizes residential power quality meters, called OPQ Boxes, in order to detect anomalies in the electricity distribution across the Oahu power grid. In addition to OPQ Boxes, the OPQ project utilizes cloud-based aggregation services for power quality event detection, classification and display. The block diagram of the OPQ network is shown in Figure 3.1 .

The major components of OPQ are:

- OPQ Box: An open source power quality meter which conforms to Napali Framework requirements for the "source".
- Makai: data aggregation and event detection service that I designed and that runs as part of the sensor network "sink".
- Mauka: event analysis and classification service designed by my colleague Anthony Christe.

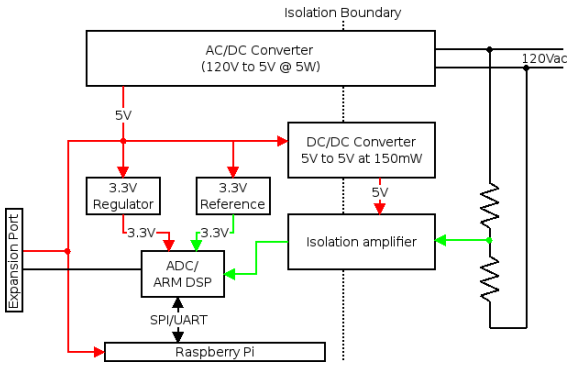
The following sections describe the OPQ network components, services and protocols.

3.1 OPQ Box

OPQ Box is a power quality meter I designed for the OPQ project, which focuses on providing the means for cheap, extensible and accurate residential power quality measurements. The block diagram of the current revision of OPQ Box, OPQ Box2 is shown in Figure 3.2a. A complete device is shown in Figure 3.2b.

3.1.1 Hardware

The power system of the OPQ Box2 electrically isolates most of the device from the AC mains power. An isolated AC-DC converter generates $5V_{dc}$ from the mains $120V_{ac}$. 5V is used to power the Raspberry Pi, equipment connected to the expansion port, 3.3V regulators and voltage reference



(a) OPQ Box2 Block Diagram. The power path is in red, signal path is in green and the digital IO is in black.

(b) OPQ Box2 in an ABS plastic enclosure.

Figure 3.2: (a) OPQ Box2 block diagram and (b) production OPQ Box ready for deployment

and an isolated DC/DC converter. 3.3V is used to power the isolated end of the isolation amplifier as well as the STM32F3 analog to digital converter/digital signal processor (ADC/DSP). The hot side of the isolation amplifier is powered from the isolated DC/DC converter. This allows OPQ Box to function with a battery attached to the expansion port, so that it may retain data and continue to operate during a power outage.

The analog signal path of the OPQ Box2 is complicated by the fact that the STM32F3 ADC/DSP is electrically isolated from the mains power. A previous iteration of the OPQ Box, OPQ Box1, overcame this by employing a small circuit board mount isolation transformer. Unfortunately it was found that the frequency response of these transformers varied wildly between individuals, thus incurring a lengthy calibration process for each device. Design on the OPQ Box2 solved this issue by using an AMC1100 isolation amplifier as the isolation component. Internally AMC1100 consists of a single die comprised of a $\Sigma\Delta$ analog to digital and digital to analog converters. These converters are separated by a silicon glass region on the integrated circuit which acts as a coupling capacitor. Since the signal passes the isolation boundary as a $\Sigma\Delta$ encoded digital signal, it incurs no distortion and no additional calibration is required. In order to match the dynamic range of the AMC1100 the $120V_{ac}$ is passed through the resistor divider to attenuate it to $120mV_{ac}$. The input and output of the isolation amplifier is filtered with a passive first order anti-aliasing filter. The isolated signal is then digitized via a 16bit ADC of the STM32F3 DSP at $12KSps$, which gives 200 data samples per grid cycle. Internally the digitization process runs asynchronously with respect to the DSP CPU, in order to minimize timing jitter. It was verified that the sampling jitter of the ADC is less than 1us, however due to limited precision of equipment an exact figure was not established. Data stream in its digital form is transferred to the Raspberry Pi single board computer (SBC) for analysis.

Raspberry Pi SBC is responsible for signal analysis and anomaly detection. The Raspberry Pi

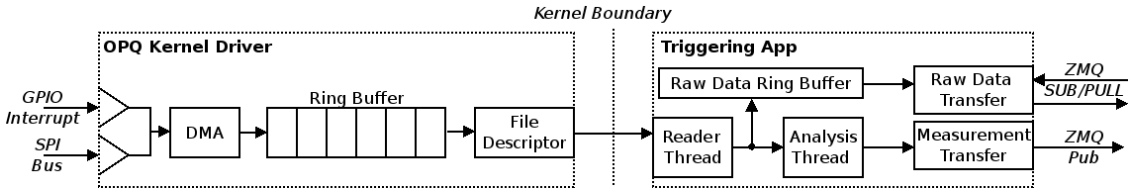


Figure 3.3: Block diagram of the OPQ Box 2 software stack.

model used in OPQ Box is the Pi Zero W equipped with 256MB of main memory and a single core 1GHz ARM11 CPU. Furthermore, Pi Zero W is equipped with an on-board 802.11n WIFI transceiver, which removes the need for an external WIFI dongle used in previous OPQ Box devices.

3.1.2 Software

The software stack of the Raspberry Pi aims to deliver a full featured power quality analysis framework despite its rather limited hardware capabilities. A block diagram of the software stack is shown in Figure 3.3. Digital data is transferred from the DSP to the Raspberry Pi via Serial Peripheral Interface, with the Pi acting as the master and the DSP as a slave device. A hardware interrupt line is used to inform Pi software that the DSP is ready for the data transfer. During the initial design of the OPQ Box 2 software, SPI data transfer was attempted in userland. However due to the lack of support for DMA in the SPI kernel-to-userland bridge, a large portion of the CPU time was spent facilitating data transfer, resulting in degraded analysis performance as well as missed data samples. Current revision of the OPQ Box 2 software stack utilizes a kernel driver for management of SPI bus. Internally the OPQ driver maintains a ring buffer of 16 windows each 200 data samples in size. Upon receiving the interrupt for the DSP, the CPU sets up the DMA transfer and the DMA engine transfers a 200 sample window into the kernel memory without CPU interaction. This scheme requires the CPU to only service 60 interrupts a second, with each interrupt requiring on the order of 100 instructions, for a CPU utilization of less than 1% in normal operation. Userland applications communicate with the kernel driver using a file descriptor, where every *read* system call yields 200 samples of raw waveform. Thus the smallest window that a userland application may process is a single AC cycle of the grid mains.

The userland component of the OPQ Box 2 software is a multi-threaded extensible analysis framework called Triggering. The reader thread is responsible for transferring and accumulating data from the kernel driver. The smallest data buffer that the Triggering application processes at any given time is 10 grid cycles or 2k samples. Once the cycles are transferred to the userland and timestamped, they are passed to the analysis thread for feature extraction, as well as to the Raw Data Ring Buffer (RDRB). Since internally all data is addressed using shared pointers, during data duplication no copying is required. RDRS is capable of buffering up to an hour of historic data before it's overwritten resulting in the RDBS maximum size of 100MB.

Analysis thread of the Triggering application performs feature extraction of the raw data windows of 2000 samples. Four metrics are extracted from the data stream:

- Fundamental frequency.
- RMS Voltage.
- Total Harmonic Distortion.
- Transient.

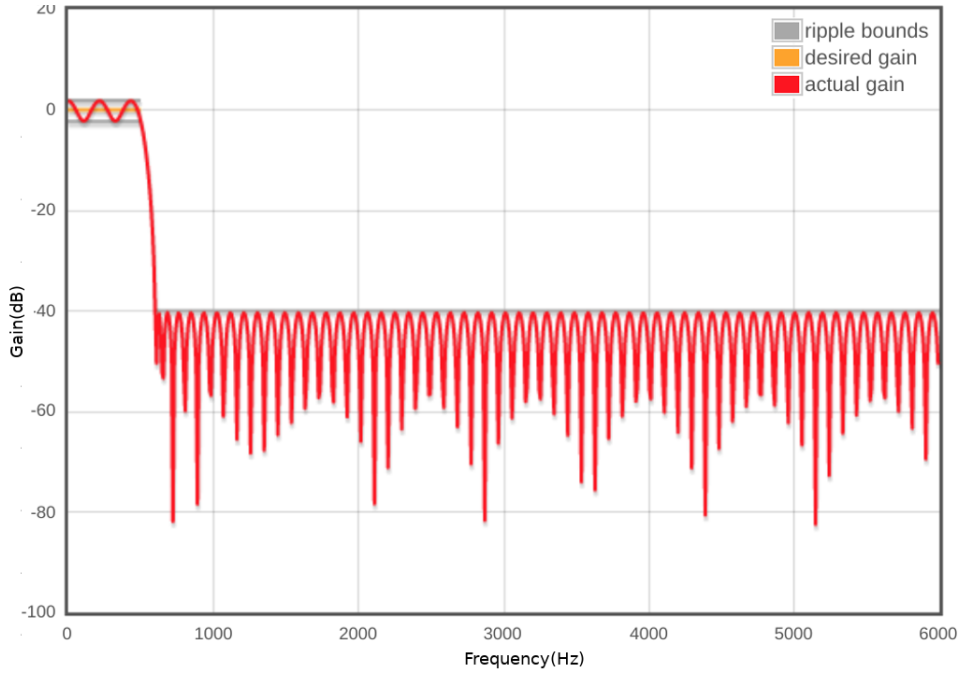
3.1.3 Fundamental Frequency

Fundamental frequency is calculated by computing the zero crossings of the AC waveform. Since a sinusoid has two zero crossings for a full cycle the frequency can be calculated as:

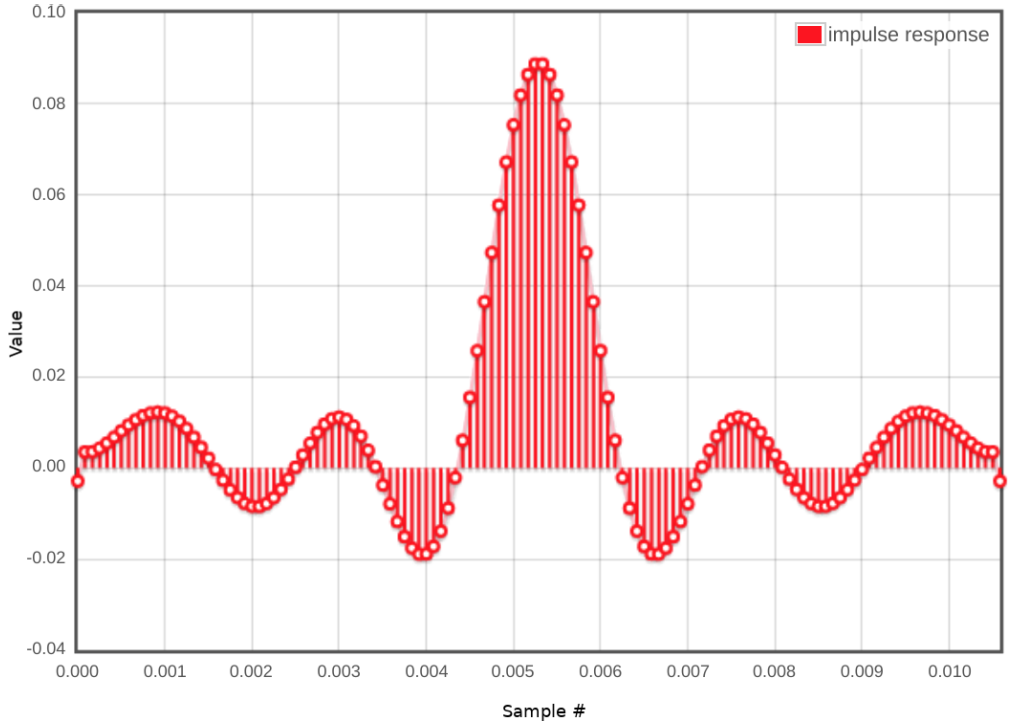
$$f = \frac{1}{\frac{2}{n}\sum_{k=0}^{k=n} \Delta t_k} \tag{3.1}$$

Where the Δt_k is the k'th time between two adjacent zero crossings. In order to improve the accuracy of the frequency calculation one must first filter out as much noise as possible. Since our sampling rate is quite high (12kSps) and the fundamental frequency is quite low (60Hz) it is very computationally expensive to perform this filtering in a single step. Instead filtering is accomplished via a set of two finite impulse response (FIR) filters shown in Figure 3.4b and 3.4d. First the Down sampling filter is applied to the raw waveform, which results in the frequency response shown in Figure 3.4a. As is evident by the plot the frequency content of the result is 0-600Hz, Thus it can be downsampled to the $\frac{N}{10}$, or 200 samples without aliasing. Next the low pass filter is applied to the downsampled waveform with the frequency response shown in Figure 3.4c.This resulting frequency content is 0-100Hz, thus all of the higher frequency harmonics and noise are removed. Finally the twice filtered downsampled waveform is used to estimate the fundamental frequency according to the Equation 3.1. The zero crossings themselves were calculated by using linear interpolation between two points which bracket the time axis.

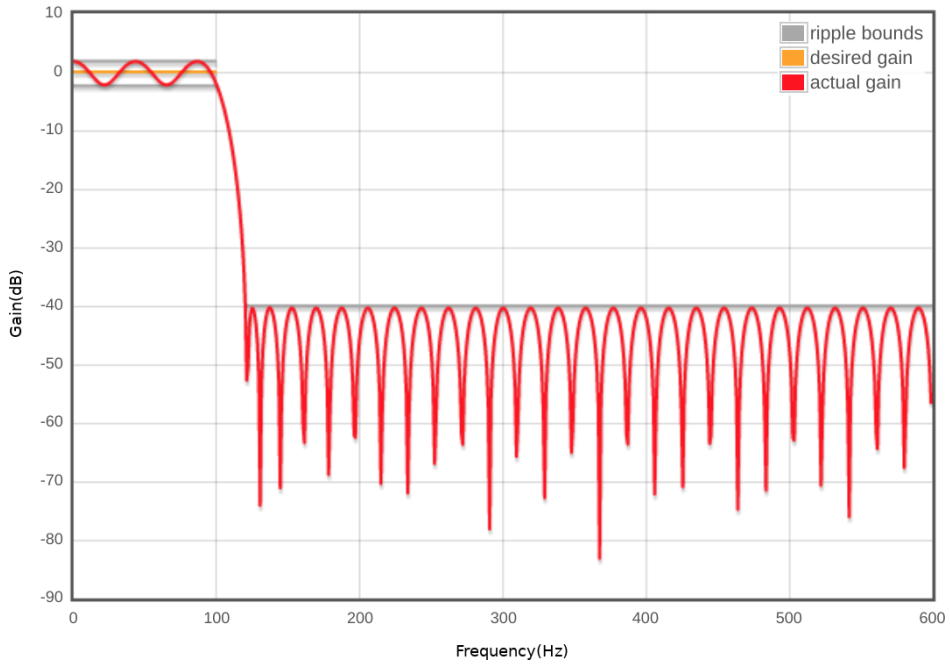
All electrical generation systems connected to the grid run synchronously with each other, meaning that while small variations in voltage are common across locations, the fundamental frequency and phase must remain strictly in sync. This effect is demonstrated in Figure 3.5, which is a frequency fluctuation event recorded on November 8 2017. While the two devices were separated by ten miles, their frequency measurements track closely together.


ripple bounds
desired gain
actual gain
20
0
-20
-40
-60
-80
-100
Gain(dB)
0
1000
2000
3000
4000
5000
6000
Frequency(Hz)

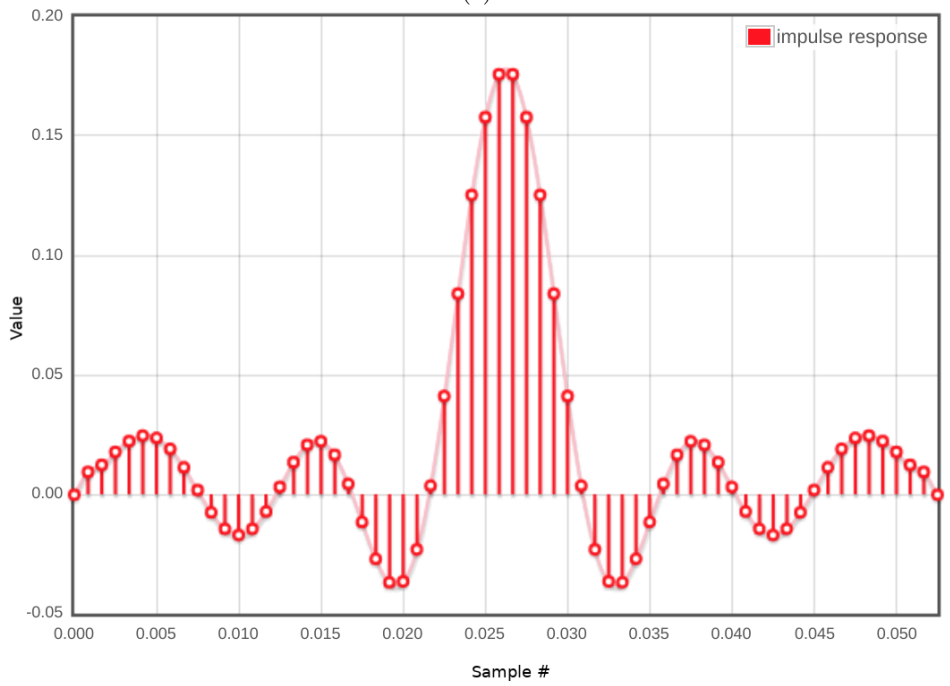
(a)


impulse response
0.10
0.08
0.06
0.04
0.02
0.00
-0.02
-0.04
Value
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.010
Sample #

(b)



(c)



(d)

Figure 3.4: Filters used for mains frequency calculation. (a) Downsampling filter gain. (b) Downsampling filter impulse response. (c) Lowpass filter gain. (d) Lowpass filter impulse response.

3.1.4 Root Mean Square Voltage

Root mean square voltage (V_{rms}) in electrical power is the equivalent value of DC voltage which would dissipate the same power in the resistive load. V_{rms} is a convenient measure for detecting

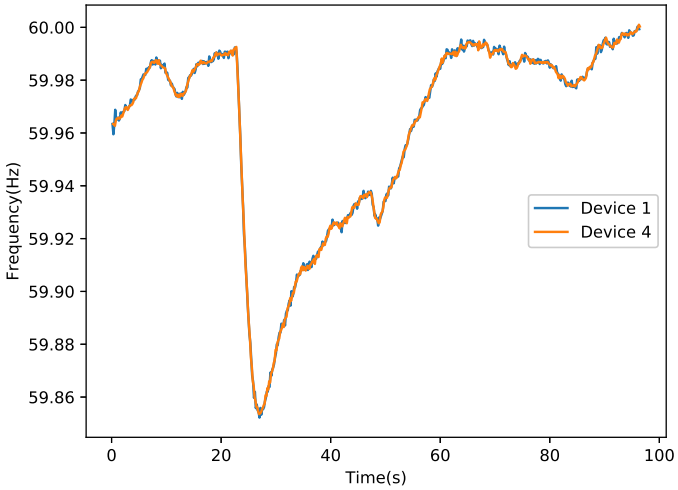


Figure 3.5: Frequency measurement across two devices recorded during a lighting strike.

voltage sags and swells, since they result in nominally higher and lower computed value. For the sinusoidal signal V_{rms} can be calculated from the peak to peak value (V_{pp}) as:

$$V_{rms} = \frac{V_{pp}}{2\sqrt{2}} \tag{3.2}$$

It is common for multimeters to employ the equation above for computing V_{rms}. However this equation is only valid for a perfect sinusoid, and thus does not result in a suitable metric for identifying power quality disturbances. Instead OPQ Box 2 computes V_{rms} as follows:

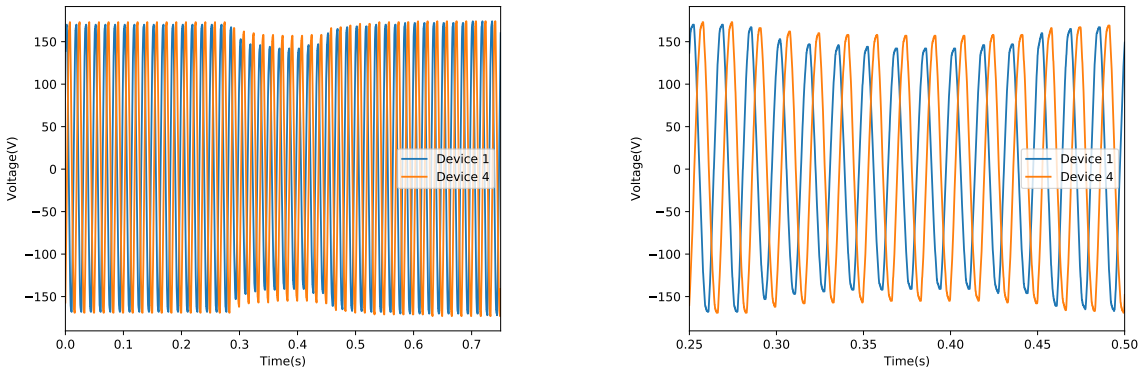
$$V_{rms} = \sqrt{\frac{1}{n}\sum_{k=0}^{k=n} V_k^2} \tag{3.3}$$

Similarly to the frequency calculation OPQ Box 2 uses a 10 cycle window for a single V_{rms} calculation, however unlike the frequency calculation the input is not filtered a priori. An example of a power quality disturbance which exhibits low V_{rms} is shown in Figure 3.6a and 3.6b. These disturbances are the result of a lighting strike recorded by two OPQ Box 2 devices on Nov 1, 2017.

3.1.5 Total Harmonic Distortion

OPQ Box calculates THD using industry standard methodology. In the power delivery industry THD is defined as:

$$THD = \frac{\sqrt{\sum V_n^2}}{V_f} * 100\% \tag{3.4}$$



(a) (b)

Figure 3.6: A lightning strike recorded by two OPQ Box 2 devices separated by 10 miles. (a) A lightning strike manifested as a V_{rms} dip which lasted 11 cycles. (b) As a consequence of using NTP these devices have a $\frac{1}{2}$ cycle mismatch in reported timestamps.

Where V_f is the fundamental 60Hz power and V_n is the power at n^{th} harmonic. It should be noted that in the power quality domain THD is expressed as a percentage as opposed to $\frac{dB}{\sqrt{Hz}}$ as used in other disciplines. Operationally, OPQ Box computes THD for 10 cycles of the fundamental frequency. First an FFT transforms the real voltage samples into its frequency components. Next, the square of the harmonic bins is accumulated and scaled by the magnitude of the fundamental power.

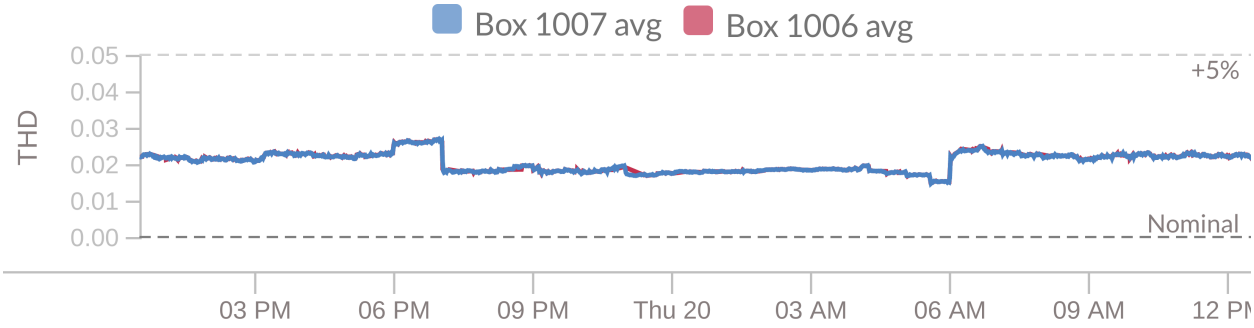
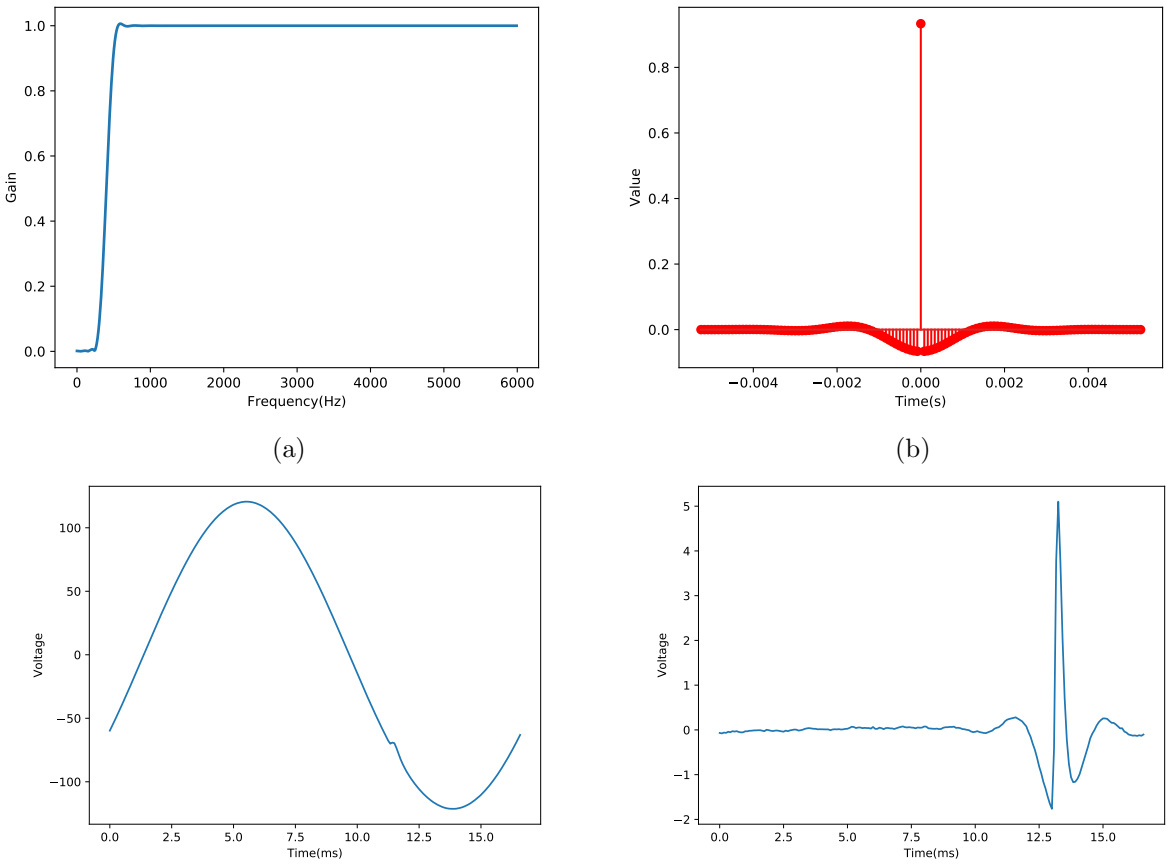


Figure 3.7: A common THD trend across two OPQ Box devices each deployed in the two Flexible Response to Ongoing Growth buildings on UH campus.

Figure 3.7 shows a common trend observed by all OPQ Box devices installed on the UH campus. For clarity only two devices are shown. It is assumed that the large drop observed daily from approximately 6pm to 6am corresponds to the automatic response of the power delivery system to the reactive power in the grid, by deploying a large capacitor bank to compensate for the current phase lag.

3.1.6 Transient Detection

OPQ Box transient detection is performed via filtering out of the fundamental frequency via an FIR high pass pass filter and searching for a maximum value in the remainder. The high pass filter has a cutoff of $400Hz$, and the filter coefficients and response are shown in Figure 3.8b and Figure 3.8a respectively. The result of the high pass filter operation is shown in Figure 3.8. Figure 3.8c shows a synthetic signal generated via a SDG1025 signal generator and fed into the OPQ Box. This signal contains a $5V_{pp}$ transient injected at 11ms. Filtered signal is shown in Figure 3.8d, with the fundamental removed and the transient preserved. OPQ Box scans for the highest sample in the filtered waveform and uses its magnitude as a transient detection metric.



(a) (b)

(c) (d)

Figure 3.8: THD detection filtering. (a) Filter gain. (b) Filter response. (c) A 5V transient superimposed onto a fundamental. (d) Filter result from (c).

It should be noted, that this transient detection method is susceptible to THD fluctuations, since any harmonic above $400Hz$ will remain in the filtered waveform. However, since the THD

information is transmitted along with the transient detection metric, they can be correlated in downstream transient detection. This effect can be seen in Figure 3.9. This figure shows both the THD and transient detection metric during a transient event. A small transient of approximately 1.6V was observed occurring at 2600s, while the sensitivity of the transient metric is clearly visible, particularly between 3000s and 4500s.

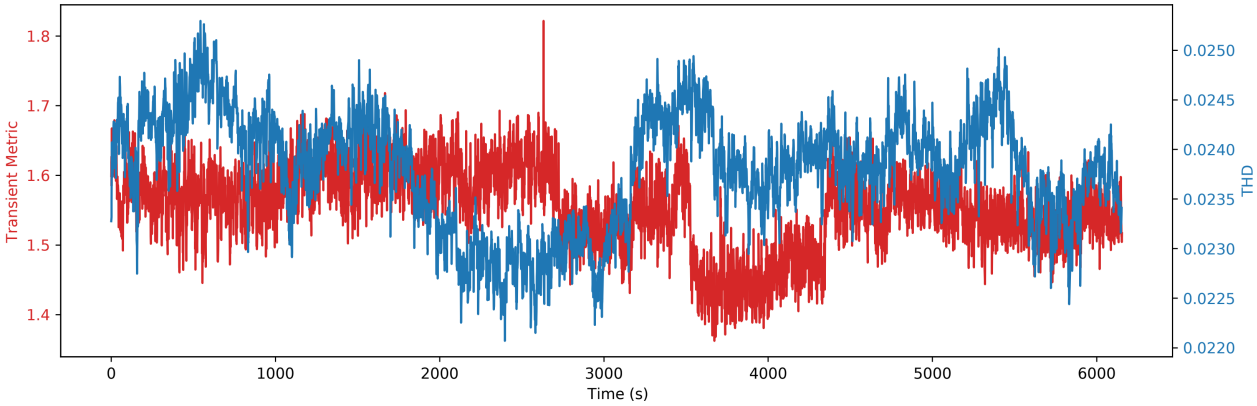


Figure 3.9: THD and Transient detection metric.

3.1.7 Network Communication

Computed fundamental frequency and V_{rms} are transmitted to the Makai service for aggregation. Data transmission is handled using ZeroMq software stack with Curve25519 elliptic curve encryption. Each device holds a unique private and public key, as well as the servers' public key, allowing both the Makai service and the OPQ Box 2 to verify its peer. Internally metrics transmission uses ZeroMq's PUB/SUB protocol. This protocol is a publish subscribe topology, with each message containing the topic, and a payload. Additionally ZeroMq pub-sub topology allows for multiple sub peers with subscriptions forwarded to the publisher automatically via a side channel. This allows for the aggregation service to be spread across multiple nodes, with minimal network overhead.

If the aggregation service determines that an anomaly has occurred, it is able to request raw waveform from the OPQ Box 2 RDRB via a separate ZeroMq pub sub channel. If the RDRB buffer contains data for the requested temporal range, OPQ Box 2 transmits the available data to the aggregation service via a push pull ZeroMq channel. Protobuf message serialization is used to encode messages across the OPQ ecosystem.

In order to make a distributed measurement, all of the OPQ Boxes on the OPQ network need to maintain an accurate time reference. Time synchronization across multiple OPQ Boxes is accomplished using the Network Time Protocol. The expansion port of the OPQ Box 2 supports a GPS receiver. However, since GPS receivers require line of sight to the sky it was not used for

deployment. NTP performance has been verified against GPS resulting in time error of $8ms \pm 5ms$ which is typical for NTP running over the Internet with a close by NTP server. This error is visualized in a Figure 3.6b. With a large coincidental V_{pp} drop across two devices, a 7ms phase error is clearly visible.

3.2 OPQ Makai

OPQ Makai implements the Napali Framework requirements for a "sink". It is a distributed extensible microservice framework responsible for receiving the triggering stream from the OPQ Boxes, locating anomalous temporal regions and requesting raw waveform for the anomalous time ranges. As evident from the block diagram shown in Figure 3.10, Makai consists of four major components: Acquisition Broker, Triggering Broker, Event Service and the Acquisition Service.

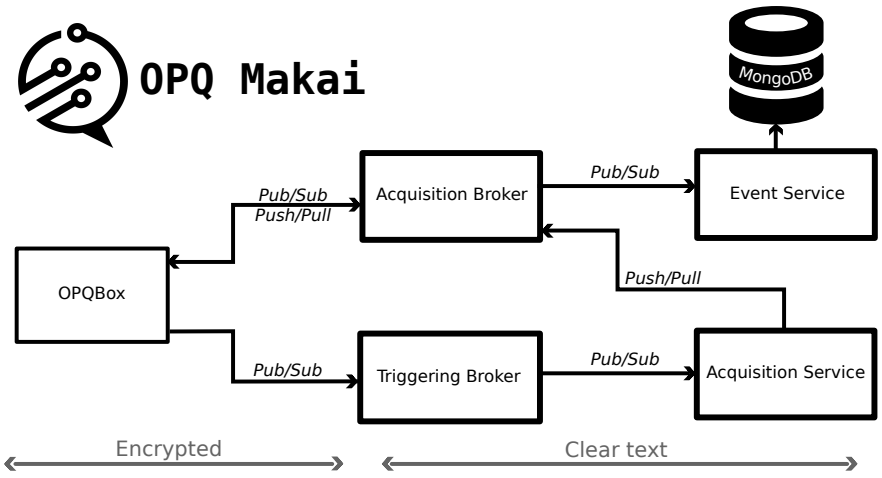


Figure 3.10: Block diagram of the OPQ Makai.

3.2.1 Triggering Broker

The triggering Broker is perhaps the simplest component of the OPQ Makai system. The triggering stream generated by the OPQ Boxes is encrypted to preserve users privacy. In order to minimize the CPU time spent decrypting the data across multiple OPQ services, the Triggering Broker is used to decrypt the data and send clear text measurements across the rest of the OPQ ecosystem. Triggering Broker uses the ZeroMq subscribe socket to receive data form OPQ Boxes, and sends it via a publish socket to any connected client. Each publish message is published to a topic which corresponds to the ASCII representation of the originating OPQ Box ID. This allows services which utilize the Triggering broker to select a subset of IDs to operate on. This is useful for load balancing the backend services, or dividing the OPQ network into separate regions with

no electrical connections between them.

OPQ Box transmits metrics which make up the triggering stream in form of Protobuf encoded messages. The structure of triggering messages is shown in Table 3.1.

Table 3.1: Triggering message structure.

Triggering Message		
Field	**Type**	**Description**
box_id	*uint32*	Device ID this message originated from.
timestamp_ms	*uint64*	Timestamp corresponding to the first cycle in the temporal window.
metrics	*map<string, Metric>*	A map of metric names to corresponding values.

Metric		
Field	**Type**	**Description**
min	*f32*	Minimum observed in this window.
max	*f32*	Maximum observed in this window.
average	*f32*	Average observed in this window.

3.2.2 Acquisition Broker

The Acquisiton Broker manages the two-way communication between the OPQ Boxes and the rest of the cloud infrastructure. Unlike the triggering stream which originates from the OPQ Box, two way communication is always initiated by the cloud services. Two way communication is realized via a command response interface, where the OPQ service initiates the communication by sending a clear text command to the Acquisition Broker, which then forwards it in the encrypted form to the appropriate OPQ Boxes.

As mention in the OPQ Box section, all communication between the OPQ Box and Makai is serialized using Google Protobuf. Protobuf allows for polylingual communication between software components across network boundaries. This is particularly important in case of the Acquisition and Triggering brokers, since they facilitate all inter-network communication. All commands between services downstream of the Acquisition Broker and OPQ Boxes take on the form shown in Table 3.2.

In order to send a message to an OPQ Box, a client generates a **Makai → OPQ Box** command containing its identity, serializes it, and forwards it to the Acquisition broker's push port. The acquisition broker assigns a unique sequence number to the request and stores it in an internal

Table 3.2: Command/Response message structure.

Makai → OPQ Box		
Field	**Type**	**Description**
seq	*uint32*	Sequence number for the command.
box_id	*uint32*	Device ID this command will be routed to.
timestamp_ms	*uint64*	Millisecond timestamp, created on command issue.
identity	*string*	Identity of the sender.
command	*Command*	Command payload.

OPQ Box → Makai		
Field	**Type**	**Description**
box_id	*uint32*	Device ID this command is routed from.
seq	*uint32*	Sequence number of the response
timestamp_ms	*uint64*	Millisecond timestamp, created on response issue.
response	*Response*	Response payload.

synchronized key value store along with the sender identity. The message is re-serialized and sent out to the desired OPQ Box via the broker's encrypted pub interface. Each message sent to the OPQ Box generates a response in form of **OPQ Box → Makai** message. This response is received by the Acquisition broker via encrypted pull interface, deserialized and validated. Response sequence number is used to determine the recipient's identity, and the message is forwarded to the correct recipient via the broker's cleartext publish interface. It is important to note that the service which generates a command and the service which receives the response need not be the same, and a command originator may place the box response into the subscribe stream of a different service. This detail becomes important during the discussion of the Event service. Furthermore, sending and receiving messages occurs asynchronously, thus multiple clients can communicate with multiple Boxes at the same time. Finally, a garbage collector cleans the synchronised key value store every hour, removing sequence numbers which did not receive a response for an OPQ Box. These sequence numbers and identities are logged, since a lack of a response indicates a device or a network fault, or a buggy client.

There are four unique payload commands/responses which an OPQ Box understands. These payloads are ignored by the Acquisition Broker and are simply forwarded to the recipient. Below is the list of commands/responses:

- **(HEALTH) Health:** The health command is broadcast periodically across all of the OPQ Boxes, in order to collect diagnostics from all OPQ devices. The OPQ Box response to this command contains diagnostic information, such as the timestamp of the last event requested, ip address and the name and strength of the WIFI network the OPQ Box is connected to.

- **(CMW) Change measurement window:** This command allows downstream services to vary how often a triggering stream message is generated and delivered to the triggering broker. This is accomplished by varying the length of the temporal window used to derive the triggering metrics.
- **(RD) Send raw data:** This command instructs the OPQ Box to send data from the its raw data buffer to the cloud.
- **(PLG) Plugin:** Send a binary payload directly to a OPQ Box Plugin.

Data fields for each command are described in the Tables 3.3 and 3.4. Notice that an **Err** response does not correspond to any commands described above. OPQBox generates this response under following conditions:

- Command payload could not be parsed.
- $start\_ms > end\_ms$ in case of the RD command.
- box_id field does not match the destination OPQ Box.
- PLG command is attempting to route to a OPQ Box plugin that is not loaded by the recipient.

In the current implementation, OPQ Makai protocol does not support multicast addressing of OPQ Boxes. Furthermore, due to the limitations of the ZeroMq API, there is no stable method of querying the clients connected to a particular port. Utilities which query all available devices, such as a periodic health check service, query the MongoDB database for a list of devices registered to the OPQ network, and send their queries to each box individually.

3.2.3 Acquisition Service

The Acquisition Service middleware resides between the Triggering and Acquisition Brokers. The Acquisition Service is responsible for the following tasks:

1. Computation of statistics of the incoming triggering stream.
2. Hosting plugins for triggering stream analysis.
3. Generating data event requests for OPQ Boxes.

A block diagram of the Acquisition Service is shown in Figure 3.11.

Table 3.3: Command Payloads

HEALTH		
N/A		This command has no fields.

CMW		
Field	**Type**	**Description**
measurement_rate	*uint32*	Number of cycles per triggering message.

RD		
Field	**Type**	**Description**
start_ms	*uint64*	Timestamp defining the beginning of the region of interest.
end_ms	*uint64*	Timestamp defining the end of the region of interest.
wait	*bool*	A flag indicating that a Box must wait if **end_ms** is in the future.

PLG		
Field	**Type**	**Description**
plugin_name	*string*	Name of the plugin.
message	*bytes*	Binary payload for the plugin.

Triggering Stream Statistics.

The Acquisition Service accesses the triggering stream by connecting to the publish socket of the Triggering broker. Since the connection is managed through the ZeroMq publish-subscribe socket, several Acquisition Services can be connected to a single Triggering broker endpoint each servicing a subset of OPQ Boxes by subscribing to only specific devices. This functionality is used to manage devices that the OPQ project sends for evaluation by organisations interested in the project. These loner devices participate in their own triggering schemes, and do not contaminate the triggering stream with messages about an unconnected grid.

Triggering messages received by the Acquisition service are buffered and stored in the MongoDB database. There are two types of statistics accumulated by the Acquisition Service:

1. *Measurements:* Every triggering stream message gets converted to a set measurement.

2. *Trends:* One minute window of triggering stream messages get feature extracted and stored.

Measurements have a one-to-one relationship with a triggering messages and take form of MongoDB documents described in Table 3.5. Measurement documents have an expire_at field instructing MongoDB to clean up the measurement collection after after a predetermined time period. Currently measurement documents persist for 24 hours.

Table 3.4: Response Payloads

HEALTH

Field	Type	Description
mac_addr	*string*	Mac address of the box.
wifi_network	*string*	SSID of connected Wifi network.
ip	*string*	Ip address of the wlan0 interface.
uptime	*uint64*	Device uptime in ms.
calibration_constant	*f32*	Calibration constant for the device.
pub_key	*string*	Byte64 encoded public key of the device.
measurement_rate	*string*	How often a triggering message is produced.

CMW

Field	Type	Description
measurement_rate	*uint32*	Old measurement rate.

RD

Field	Type	Description
start_ms	*uint64*	Timestamp defining the beginning of the region of interest.
end_ms	*uint64*	Timestamp defining the end of the region of interest.
data	*Cycle[..]*	Data divided into single cycle windows structures.

Cycle

Field	Type	Description
datapoints	*int16[200]*	Raw ADC samples for one cycle window.
timestamp_ms	*uint64*	Timestamp of the last sample.

PLG

Field	Type	Description
OK	*bool*	Flag indicating that the command was successfully routed to a plugin.

Err

Field	Type	Description
code	*uint32*	Error Code.
Error	*string*	Human readable error string.

Trends consist of feature extracted statistics accumulated from 1 minute of triggering measurements. The statistics computed are minimum, maximum and average for each box metric.

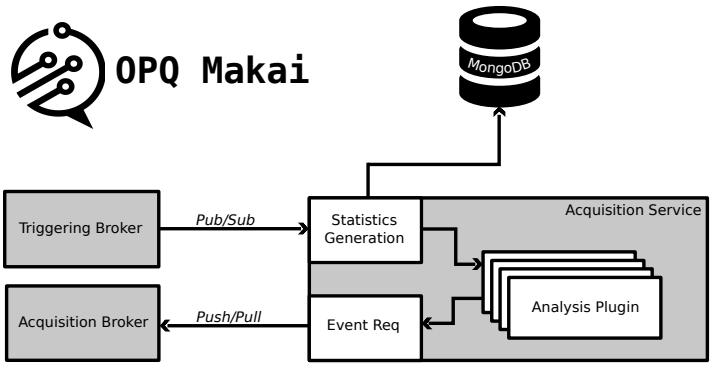


Figure 3.11: Block diagram of the Acquisition Service.

Table 3.5: Measurement Document.

Field	Type	Description
box_id	*string*	OPQ Box ID.
timestamp_ms	*uint64*	Millisecond timestamp.
voltage	*f32*	V_{rms} average value.
thd	*f32*	THD average value
frequency	*f32*	Frequency average value
expire_at	*f32*	Deletion time for the measurement.

Similarly to the Measurement documents, Trend documents expire after a predetermined period of time. Currently trends are retained for 7 days. Structure of a Trend document is shown in Table 3.6.

Triggering Analysis infrastructure.

The Acquisition Service does not include any analysis capabilities by default. Instead analysis is performed by shared library loadable plugins. These plugins can be loaded and unloaded at runtime, thus allowing live upgrading and testing of new analysis methods. Plugin are loaded using libdl by searching for C function hooks which produce a Rust structure which implements a plugin interface. The interface as well as the prototype C hook function are shown below:

```
pub trait MakaiPlugin: Any {
  fn name(&self) -> &'static str;
  fn process_measurement(&mut self, msg: Arc<Measurement>)->Option<Vec<Command>>;
  fn on_plugin_load(&mut self, json: String);
  fn on_plugin_unload(&mut self);
```

Table 3.6: Trend Document.

Trend Document		
Field	**Type**	**Description**
box_id	*string*	OPQ Box ID.
timestamp_ms	*uint64*	Millisecond timestamp.
voltage	*Statistics*	V_{rms} average value.
thd	*Statistics*	THD average value
frequency	*Statistics*	Frequency average value
expire_at	*u64*	Deletion time for the measurement.

Statistics		
Field	**Type**	**Description**
Min	*f32*	Minimum observed during the trend window.
Max	*f32*	Maximum observed during the trend window.
Average	*f32*	Average computed over the trend window.

```
}
#[no_mangle]
pub extern "C" fn _plugin_create() -> *mut MakaiPlugin;

#[no_mangle]
pub extern "C" fn _plugin_destroy(plg : *mut MakaiPlugin);
```

The *_plugin_create* function is marked as *no_mangle* which instructs the compiler to retain the function name so it can be located by the plugin loader. It returns a mutable pointer to a boxed type which implements the MakaiPlugin interface. Since Rust is a explicit memory management language, and the runtime is not aware of the internal structure of the Boxed type beyond its interface, a helper function *_plugin_destroy* is used to cleanup the plugin object during the unloading process. Each plugin runs in a separate thread with a synchronized queue from which it draws measurement messages. If a plugins is not able to keep up with the triggering stream, The Acquisition service will start dropping messages from its input queue, and a message indicating this fault will be stored in its log. Once a plugin is loaded it will receive its settings in the form of a JSON encoded string using the *on_plugin_load* method. This allows it to initialize all of the internal data structures to a known state. Similarly *on_plugin_unload* method is used to inform the plugin that it is about to be unloaded.

The processing of the triggering stream occurs in the *process_measurement* method. This methods takes a reference counted Measurement message in form shown in Table 3.1. Return type of this method is an optional list of commands to forward to the Acquisition broker in form shown in Table 3.2. All of the Commands in the list are treated to be pertaining to the same event.

Event Request.

Once a plugin determines that an event of interest has taken place, it will emit a list of **RD** commands with structure shown in Table 3.3. Event requester will fill in the identity field with the following string:

$$identity = \text{“}data\_\text{”} + EVENT\_TOKEN + \text{”}\_\text{”} + ACQ\_UUID$$

The “*data_*” prefix is used to route the messages responses from the OPQ Box to the Event storage service. As mentioned previously, responses from device command need not return to the same service that sent the command. Instead, the responses will be routed to a service which is subscribed to the identity used in the request. The $EVENT\_TOKEN$ root is a 16 character random hexadecimal string, which identifies data request corresponding to the same event. Since a single event may contain data from multiple OPQ Boxes, and each event waveform will share a unique $EVENT\_TOKEN$ root. The ACQ_UUID postfix is used to identify the Acquisition service which triggered the data request. Multiple Acquisition Services can be used to service the OPQ network, and each one is identified using its own UUID identifier. This identifier may be assigned to each service via the configuration file, or autogenerated at startup. Once the identity of each command is filled in, each command is forwarded to the Acquisition broker for routing to the OPQ Box experiencing a power quality disturbance.

3.2.4 Event Service.

The Event service is a microservice which stores raw data generated by OPQ Boxes in the MongoDB Database. On initialization, the Event service queries MongoDB database for the highest event number recorded so far, connects to the Acquisition Broker's publish port, and subscribes to all messages that start with the prefix “*data_*”. This allows the Event service to capture every response from OPQ Boxes generated from commands issued by the Acquisition service plugins. Once the Event service receives a data response with an identity containing an $EVENT\_TOKEN$ it hasn't seen before it will increment the event number, and store it in an internal key value store. This way every future data message which contains that $EVENT\_TOKEN$, will be appended to the same event. ER diagram of the MongoDB event storage is shown in Figure 3.12. The Events collection contains metadata for individual events. Box_events collection contains metadata for event information from individual OPQ devices. Finally gridfs, MongoDB's internal file storage, is used for storing event data.

3.2.5 Acquisition Service Plugins

The Acquisition service plugins are responsible for monitoring the OPQ Box triggering streams and determining when to request raw data. As such, the Acquisition service plugins constitute the

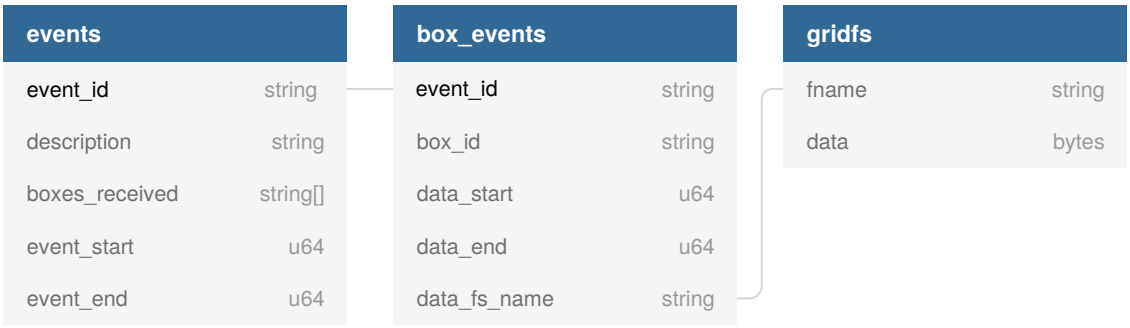


Figure 3.12: ER diagram of event storage in MongoDB

majority of business logic for the OPQ Makai system. There are currently four plugins developed for this system:

1. Health Plugin.
2. Debug Plugin.
3. Threshold Trigger.
4. Napali Trigger.

Health Plugin

The health plugin is responsible for monitoring the triggering messages passing through the Acquisition Service. It maintains a REST endpoint for reporting the OPQ Network status. Furthermore the REST endpoint can be used to trigger a device for debugging purposes.

Debug Plugin

Debug plugin consists of a Lisp interpreter which can manipulate the triggering stream, and emit triggering commands for devices. It can be used to quickly and interactively test new analysis methods, and debug the OPQ network.

Threshold Trigger

Threshold trigger plugin is used for in-situ comparison of a naive triggering method to Napali. Every time the Threshold plugin receives a triggering message, it will check it against global thresholds. If a metric value is outside of the predetermined thresholds the Triggering plugin will latch the timestamp and the id of the affected device. Once the metric returns to normal, Threshold plugin will emit a request for raw data to the device which is experiencing an event. If an event

is longer than 50% of the RDRB of an OPQ Box buffer an event will be broken up into several shorter events. Threshold values for each metric are shown in Table 3.7

Table 3.7: Threshold values for each metric

Metric	**Min**	**Max**
V_{rms}	$115V_{rms}$	$125V_{rms}$
$f_{fundamental}$	$59.9Hz$	$60.1Hz$
THD	N/A	5%
$Trans$	N/A	$8V$

The Threshold Trigger plugin functions in the same way as a self-triggering device. Since this method of event detection does not rely on intra-box information sharing, it functions in the same manner as if the OPQ Box itself is performing event detection. This allows for experimental comparison of the Napali method efficiency with the self-triggered methodology.

Napali Trigger

Napali Trigger plugin implements the Napali event detection method. Napali Trigger state machine is kicked off the same way as the Threshold trigger. If a received metric value is outside of the predetermined thresholds Napali plugin will latch the timestamp and the id of the affected device. Similarly, once the offending metric from all devices return to normal, Napali plugin will emit a request for raw data to the device which is experiencing an event. The main difference between the Napali trigger and the Threshold trigger are two-fold:

1. An event is not considered finished until every device's metrics return to the nominal threshold range.

2. Metric value which does not surpass the threshold but is placed 3σ of the mean, will mark the device as an event participant.

The first difference allows Napali to track a propagation of a fault throughout the power delivery network. Furthermore, during a lifetime of a power quality event, it may shift from one metric to another. As the disturbance travels through the power-grid, Napali will aggregate all of the temporally localized disturbances into one event. The second disturbance allows for recording of sub-threshold data to be acquired from devices which did not pass the threshold, but exhibited anomalous behaviour.

The mean and standard deviation are calculated using two first order IIR filter loops as shown in Equation 3.5:

$$\begin{aligned} \mu_{n+1} &= (1-\alpha) * \mu_n + \alpha * m \\ \mu_{n+1}^2 &= (1-\alpha) * \mu_n^2 + \alpha * m^2 \\ \sigma_n^2 &= \mu_n^2 - (\mu_n)^2 \\ \sigma_n &= \sqrt{\sigma_n^2} \end{aligned} \tag{3.5}$$

Where μ is the mean, σ is the standard deviation, m is the new sample and α is the decay parameter. Notice that $(\mu)^2$ is the square of the mean, and μ^2 is the mean of the squares. The α parameter determines the memory of the filter, and can be computed as:

$$\alpha = 1 - e^{2\pi \frac{T_{sample}}{T_{memory}}} \tag{3.6}$$

Where T_{sample} is the sampling rate of the system, 1s for OPQ, and T_{memory} is the desired response of the system.

Using an IIR filter for mean and standard deviation calculation is computationally cheaper than using a FIR window. Furthermore, using the IIR filter makes it trivial to tune the response of the system during runtime.

3.3 OPQ Mauka

The OPQ Mauka service is responsible for higher level classification and filtering of the anomalies generated by the OPQ Makai.[5] Since anomalies generation only relies on the triggering stream features and not raw data, the OPQ Makai is not able to ascertain if the anomaly is an actual power quality event, event type, or its severity. The OPQ Mauka on the other hand operates on the raw data, thus it is able to perform high level analysis to meet industry standards for vent classification. The block diagram of the OPQ Mauka is shown in 3.13.

Currently OPQ Mauka supports the following classification strategies:

- **ITIC** Power acceptability curve used to classify short term voltage sags.
- **IEEE 1159 Voltage** Voltage classification based on the IEEE 1159 power quality standard.
- **Brownout Detection** Classification of medium to long term voltage sags.
- **Total Harmonic distortion** Classification of events via harmonic analysis.

Once the anomaly is classified by OPQ Mauka, and the power quality characteristics are confirmed, it may be aggregated with other anomalies to form a disturbance. Disturbances are composed of raw box data, analysis results as well as expert annotations and other metadata.

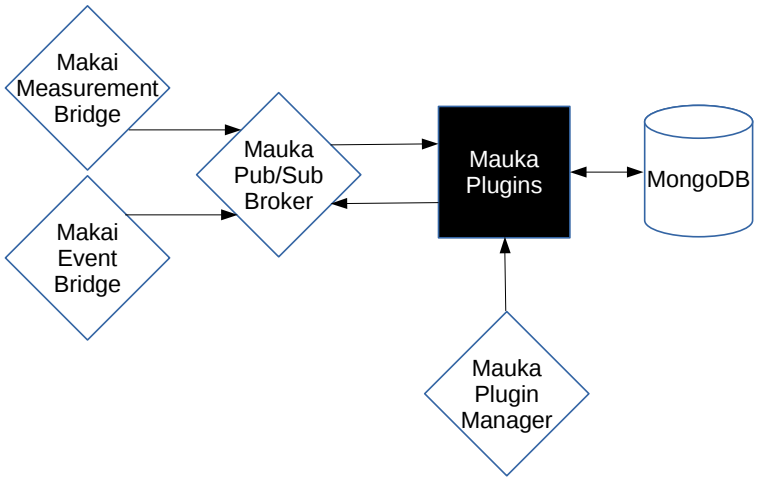


Figure 3.13: Block diagram of the OPQ Mauka.

3.4 OPQ View

OPQ View is the primary user interface to the OPQ ecosystem. View is written in JavaScript using the Meteor framework, and provides a robust and easy to use interface to the OPQ Box triggering stream, Makai triggering anomalies, and to the Mauka PQ disturbances. Furthermore, View provides an administration interface for initial setup and maintenance of the OPQ devices, and services. Finally OPQ View monitors the health of the OPQ components, keeping track of the individual box uptimes, and component failures. A screenshot of the recent OPQ View build is shown in Figure 3.14

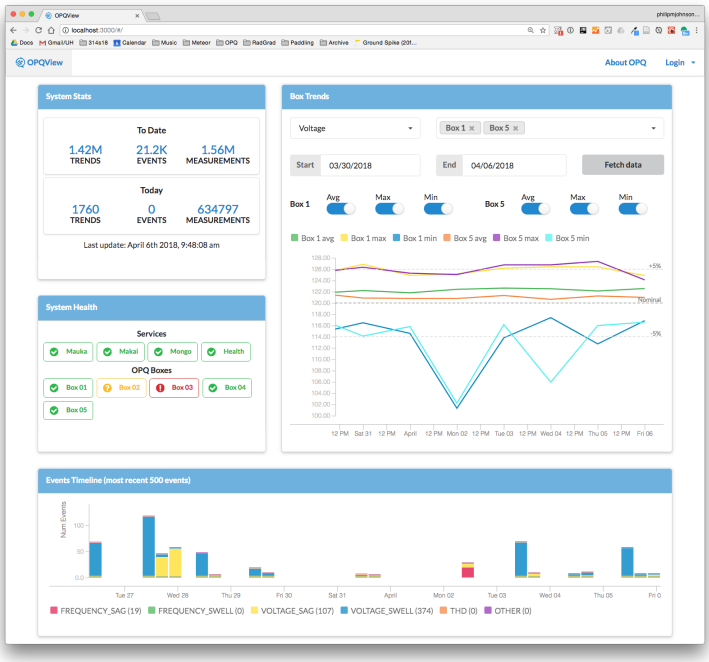


Figure 3.14: Screenshot of a recent OPQ View build.

CHAPTER 4
EXPERIMENTAL EVALUATION

Validation of the Napali approach is intrinsically linked to validation of Makai, both with synthetic benchmarks and in-situ. Since OPQ utilizes a custom power quality measurement device, its performance needed to be characterized prior to Makai evaluation.

4.1 OPQ Box Synthetic Evaluation.

A mentioned in Section 3.1.2, OPQ Box generates 4 metrics in order to enable event detection. In order to evaluate the limits of detection capabilities for each one of these metrics, an OPQBox was fed with synthetic waveform generated by the SDG1025 function generator. By utilizing a function generator the entire device including the hardware analog front end could be evaluated. Since the function generator is not capable of supplying $120V_{rms}$ signal, a $120mV_{rms}$ signal was fed on a low side of the OPQBox resistor divider, while the device was powered via an external 5V power supply.

4.1.1 Fundamental Frequency

Fundamental Frequency computation was evaluated by generating a 60Hz sine wave via the SDG1025 and supplying it to the OPQBox. Calculated frequency was accumulated by analyzing the device triggering stream. The resulting histogram is shown in Figure 4.1 As shown, the resulting distribution collected frequencies acquired over 2000s has a $\sigma = 420uHz$. In other words 99.7% of the collected measurements were between 59.998Hz and 60.001Hz.

4.1.2 Root Mean Square Voltage

Similarly to the fundamental frequency characterization, V_{rms} calculation was evaluated by supplying the OPQBox with a 60Hz, $120mV_{rms}$ sine wave via the SDG1025. Calculated RMS was accumulated by analyzing the device triggering stream. The resulting histogram is shown in Figure 4.2

As shown, the resulting distribution of collected V_{rms} measurements, acquired over 2000s has a $\sigma = 9.34mV$.

4.1.3 Total Harmonic Distortion

THD performance of the OPQBox was validated by injecting a various harmonics of 60Hz superimposed onto the 60Hz, $120mV_{rms}$ sine wave into the device via the SDG1025 arbitrary waveform generation capability. THD calculation results were acquired from the OPQBox triggering stream

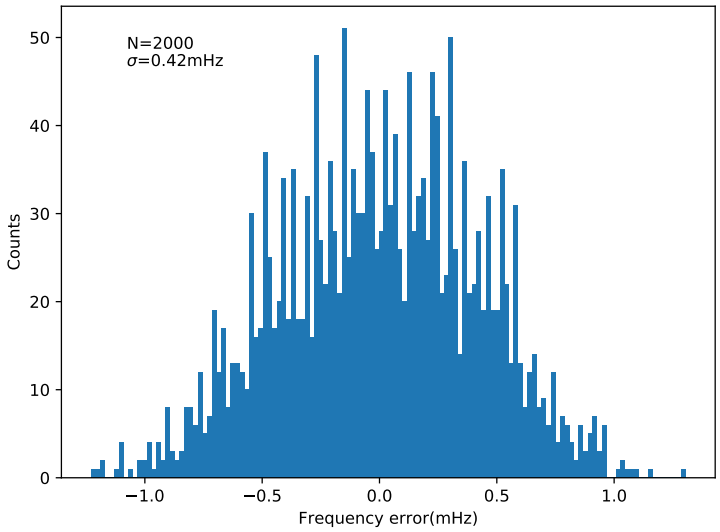


Figure 4.1: OPQBox frequency response.

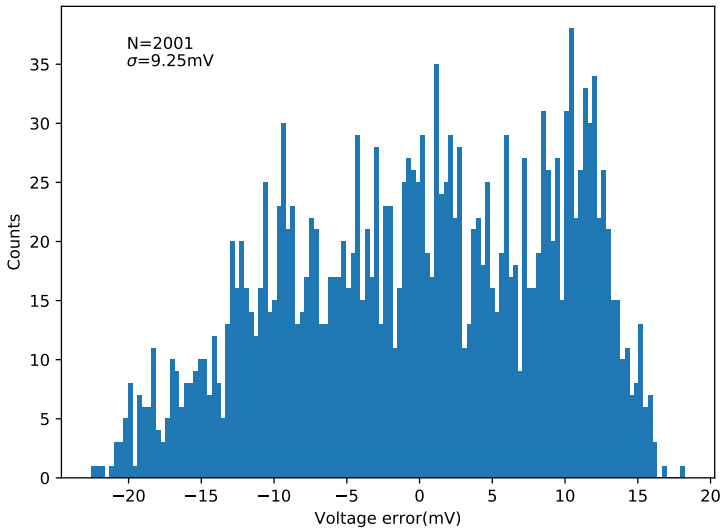


Figure 4.2: OPQBox V_{rms} response.

for analysis. As expected, the resulting detection efficiency remained self-consistent across all harmonics. Figure 4.3 shows a histogram of the error in THD values computed from a 60Hz, $120mV_{rms}$ sinewave superimposed with a 240Hz $1.2mV_{rms}$ sine wave. This measurement is equivalent to 1% THD at the 4^{th} harmonic.

As shown, the resulting distribution of collected THD measurements, acquired over 2000s, has a $\sigma = 0.001\%$.

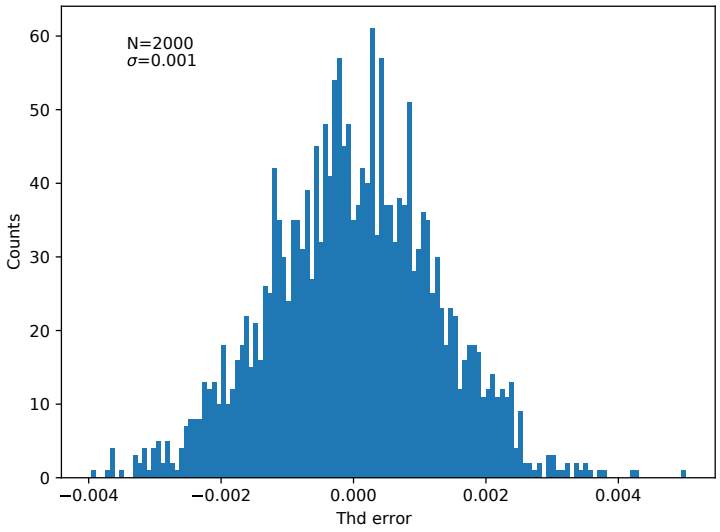
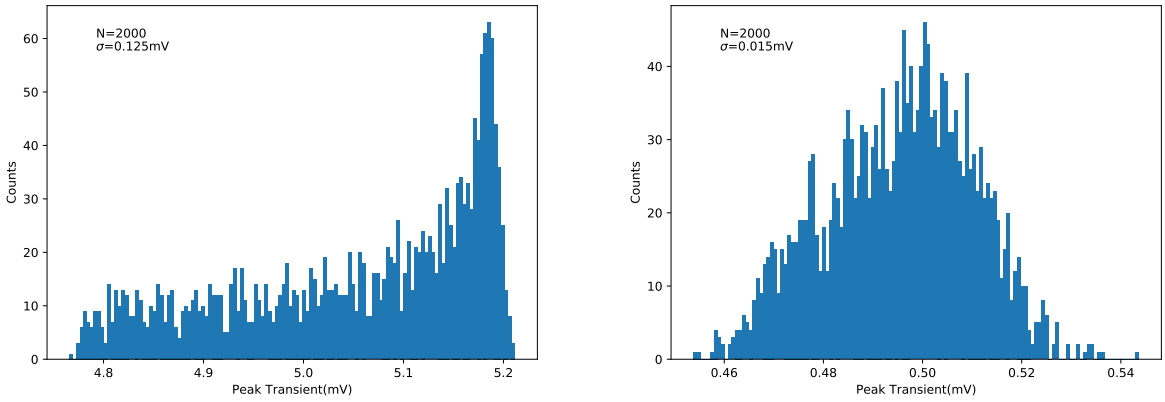


Figure 4.3: OPQBox THD response.

4.1.4 Transient Detection

Transient detection performance was evaluated by injecting a transient superimposed onto the 60Hz, 120mV_{rms} sine wave into the device via the SDG1025 arbitrary function generation capability. Transient detection results were acquired by capturing and analyzing the device triggering stream. Transients of various shapes and magnitudes were tested.



(a) (b)

Figure 4.4: Transient detection metric with a 5V transient(a), and 0.5V transient(b)

Figure 4.4 shows the resultant transient detection metric for two transients. The shape of the transient is the same and is shown in Figure 3.8c. Interestingly, in case of a 0.5V transient the

metric results in a much tighter distribution with $\sigma = 0.015V$, while in the case of a 5V transient the distribution exhibits a lower sideband tail. Since the transient is injected in a random position in the cycle, and the sampling rate of the DG1025 is significantly higher then sampling rate of the OPQBox(25Msps vs 12Ksps), the peak of the transient will sometimes fall in between the consecutive samples of the OPQBox. In the $0.5V$ transient case this effect is alleviated, since the transient is so small. Regardless, the result shown in Figure 4.4b is presented only as a synthetic benchmark, since OPQBox is expected to operate in an environment with THD larger then 0.4% at $> 400Hz$ required to detect a $0.5V$ transient. As such, the figure of $\sigma = 0.125V$ should be considered valid for the OPQBox transient detection capability. Since this metric is only used in transient detection and not characterization, it was found to be sufficient.

4.2 Real-world evaluation: The University of Hawaii deployment.

As part of the Napali evaluation, the OPQ system was deployed across the University of Hawaii Manoa campus (UH). This location was advantageous because it is an isolated microgrid connected to the Oahu powergrid only via a single 46kV feeder as shown in Figure 4.5. Another advantage of the UH campus is the high number of smart meters deployed across various levels of the power delivery infrastructure. While the purpose of these meters is to monitor the power consumption, they do include some rudimentary power quality monitoring capabilities. Data from the campus deployed meters was used as ground truth for comparison against the measurements, and for analysis performed by the OPQ project. The location of smart meters in the grid topology is shown in Figure 4.5 as the M nodes. As evident by the meter location none of them were monitoring the consumer level power and mainly focused on the higher voltage power delivery. This placement was a consequence of the smart meters' role as a consumption monitor, and thus the deployment of the OPQ Boxes at the residential level complemented UH power quality monitoring capabilities without introducing redundancies.

The University of Hawaii power grid supplies a highly diverse infrastructure. Beyond traditional residential equipment such as computers and consumer grade electronics, the UH power grid powers scientific and laboratory equipment, machine shops, and server farms. All of these elements have varying requirements/tolerances for power quality anomalies as well as different levels of power quality "pollution". Furthermore, some of the electricity consumers in the UH campus are entirely unique. For example, the free electron laser located in the Watanabe Hall is one of the only free electron lasers in the world, and the impact/sensitivity of power quality on the instrument are completely unstudied.

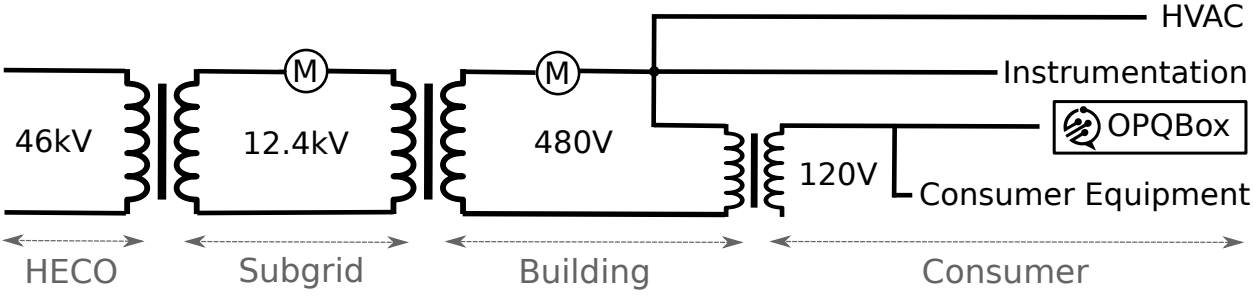


Figure 4.5: University of Hawaii at Manoa power delivery infrastructure.

There are 74 smart meters deployed across the UH campus. These meters measured the fundamental frequency V_{rms}, power consumption, reactive power, and power factor. Data from these meters was cross-referenced with the Napali detection system in order to ascertain its benefits.

OPQ Box placement was specifically selected to cover as much of the University of Hawaii power delivery infrastructure as possible. The OPQ Box deployment is shown in Figure 4.7. By spreading out devices across the entire power grid, OPQ system is able to monitor the propagation of power

quality disturbances throughout the UH power grid. Consider the event shown in Figure 4.13. Figure 4.6 shows the same event with the fundamental and harmonics suppressed using a notch filter bank. Furthermore, a location annotation is added to indicate the device deployment. The most affected devices were located at the Physical Plant and Hamilton Library, recording a $60V_{pp}$ transient. Incidentally, both of these devices are monitoring a subgrid rooted at transformer(MA4). Another device that recorded this event was located in Watanabe Hall. This device recorded a $30V_{pp}$ transient, still above the threshold for detection. This device was monitoring the the subgrid rooted at the transformer LA4. The final device was located at the parking structure entirely across campus. This device recorded a $15V_{pp}$ transient, about 1V below the required magnitude for the threhold based detection. However, Napali was able to determine that the parking structure OPQ Box was affected by the disturbance, and requested the raw data regardless.

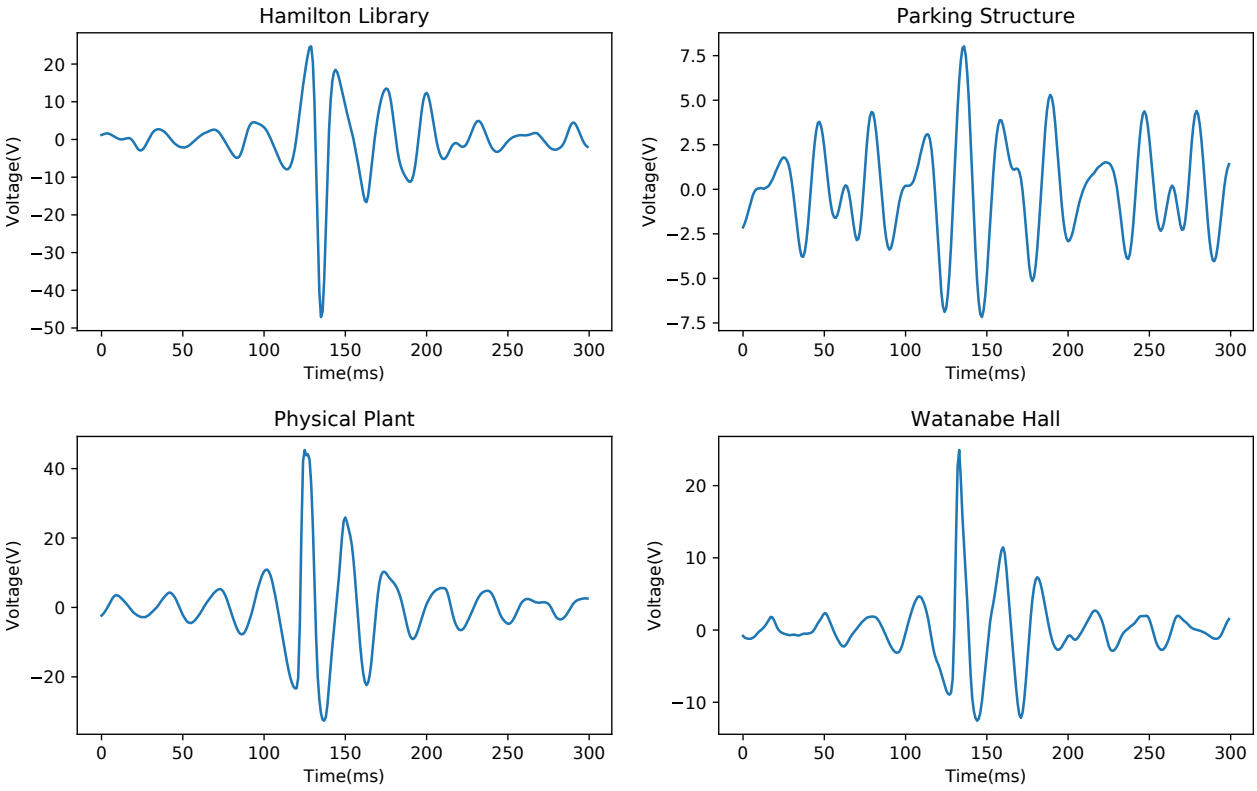


Figure 4.6: Filtered Transient from event shown in Figure 4.13

From the data gathered by Napali as shown in Figure 4.6, it seems apparent that the disturbance originated at the subgrid rooted at the transformer MA4. The Watanabe device was affected due to the short geographic and electrical distance to the MA4 subgrid. By the time the transient reached the parking structure, it was significantly attenuated by the transformers and transmission lines. It was only detected due to the sub-threshold detection ability of the Napali framework.

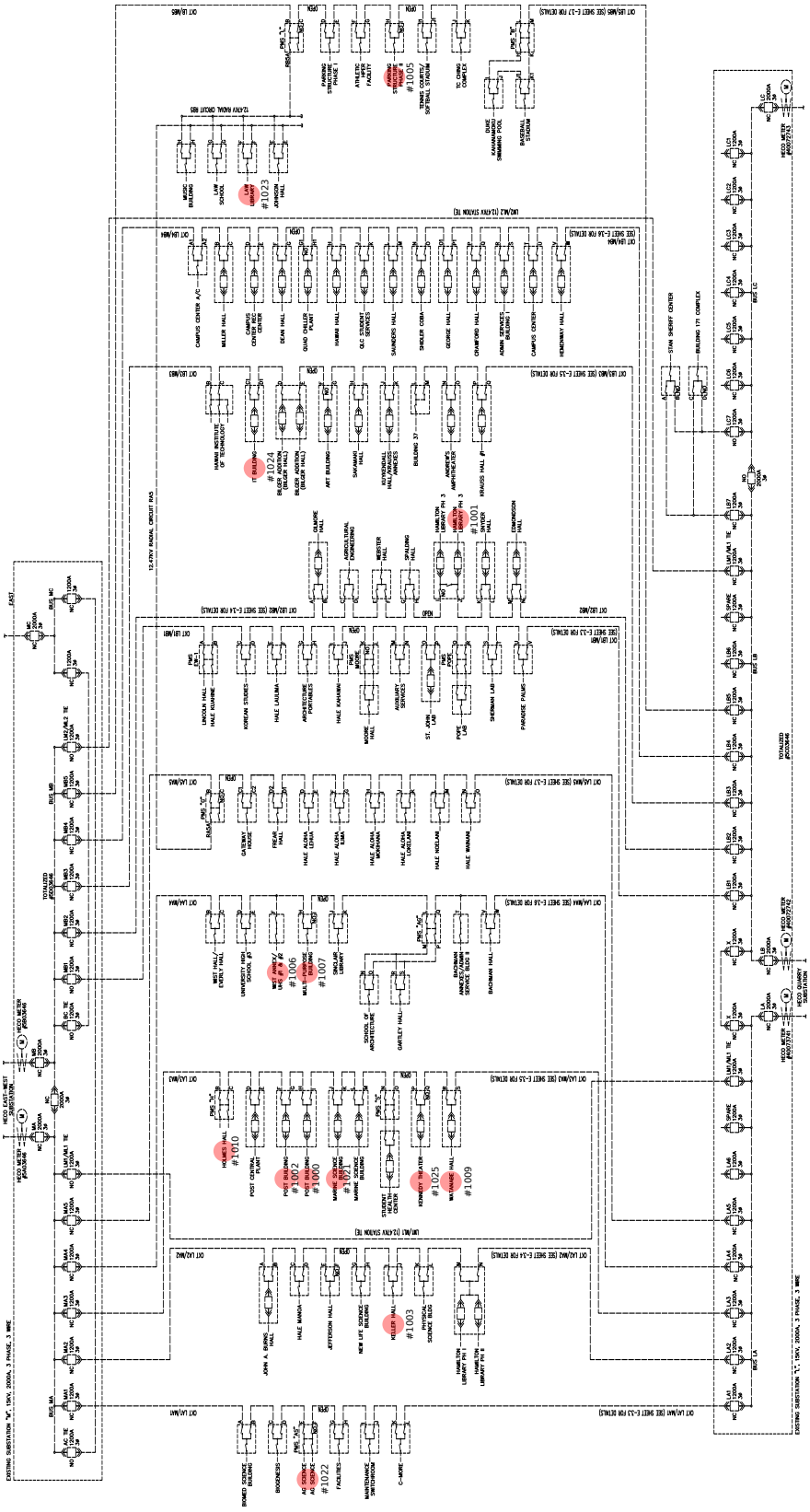


Figure 4.7: OPQ Box locations and device IDs across University of Hawaii.

4.2.1 Selection of α parameter

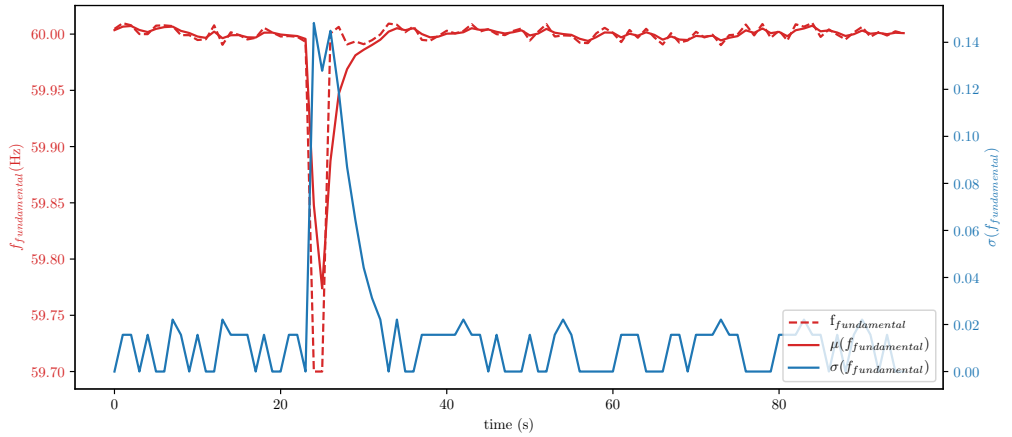
Napali maintains a mean and a standard deviation for each metric generated by the OPQ Box. Instead of maintaining a window of N latest datapoints a lossy mean algorithms is used to derive the statistical representation of the data as shown in Equation 3.5. This allows for a low overhead metric computation regardless of the sliding window length. Nonetheless, Napali maintains 1 hour of captured metrics in memory along with 1 day of metrics in the database.

The main tunable parameter in Napali is the α coefficient used in the low pass filter as shown in Equation 3.5. This parameter determines the memory of the lowpass filter used in the calculation of the mean and the standard deviation of metrics from OPQBox data stream. These statistics are in turn used during the Napali triggering process to locate sub-threshold gridwide events.

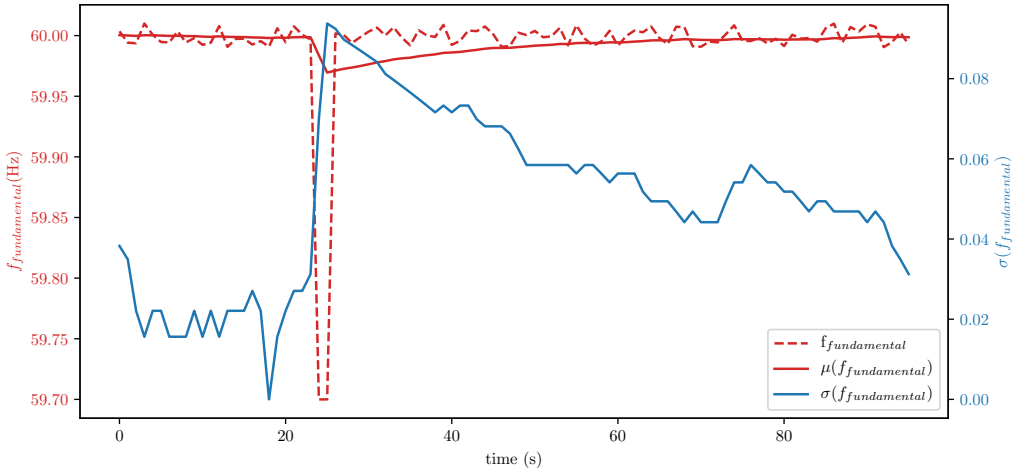
A smaller α parameter corresponds to a longer memory in the low pass filter as shown in Equation 3.6. This is further visualized in Figure 4.8. In particular this Figures 4.8a and 4.8b show the response of the IIR low pass filter to the simulated frequency measurements. The dashed red line represents the frequency measurement, solid red line represents the filtered mean, and the blue line represents the standard deviation. Figures 4.8a shows the filter response for $\alpha = 0.5$ or $T_{memory} \approx 10s$. As evident from the plot, the mean and the standard deviation quickly recover from the transient and return to their nominal values. Furthermore, the mean is closely tracking the random fluctuations present in the measurement. Figures 4.8a shows the filter response for $\alpha = 0.05$ or $T_{memory} \approx 123s$. While the stimuli remains the same, it takes significantly longer for the statistics to recover. Additionally, the mean no longer tracks the frequency fluctuations present in the simulated data.

Picking the α parameter for Napali is extremely domain specific, as it depends on the frequency content of the triggering stream. Intuitively, the T_{memory} parameter needs to be long enough to adjust to gradual changes in the triggering stream for the mean calculation, and dampen the standard deviation for detection of multiple consecutive anomalies. In addition, it needs to be short enough to converge on the mean and the standard deviation during a step-like transition in the triggering stream.

Luckily, in the Power Quality domain the Napali α selection is fairly forgiving. This is demonstrated in Figure 4.9. This graph represents the amount of time that Napali considered one of the metrics to be outside of the 3σ of the mean for various values of α. The triggering stream used to generate these values was captured over 24 hours by one of the OPQBoxes deployed on the University of Hawaii campus. All devices deployed thus far have followed a similar pattern. With $20s < T_{memory} < 2Hr$ the triggering stream resulted in similar behaviour, with the system correctly marking all potential sub-threshold events. At $T_{memory} \approx 20s$, system quickly recovered from large jumps in the triggering stream, however it marked a significant number of small anomalies ($\Delta_f > 0.01Hz$, $\Delta_v > 0.1V \ldots$ etc) as outside 3σ, and thus candidates for sub-threshold events. At $T_{memory} \approx 2Hr$, system took significant amount of time to recover from large jumps in triggering



(a)



(b)

Figure 4.8: μ and σ behaviour with a)$\alpha = 0.5$ and b)$\alpha = 0.05$

metrics, thus marking the metric as outside of 3σ for many tens of minutes. Furthermore, some of the larger anomalous measurements ($\Delta_f > 0.05Hz$, $\Delta_v > 2V \ldots$ etc) were no longer flagged as sub-threshold candidates. Outside of the two extremes, the system behaviour was quite similar. During all of the deployments the OPQ system was operating with:

$$\alpha = 0.05 \tag{4.1}$$

Which corresponds to the $T_{memory} \approx 2$ minutes. Thresholds which initiate the Napali event detection state machine are shown in Table 3.7.

The effect of the of selecting alpha as shown in Equation 4.1 can be observed in Figure 4.10. This figure shows interesting features from the same dataset that was used to produce Figure 4.9.

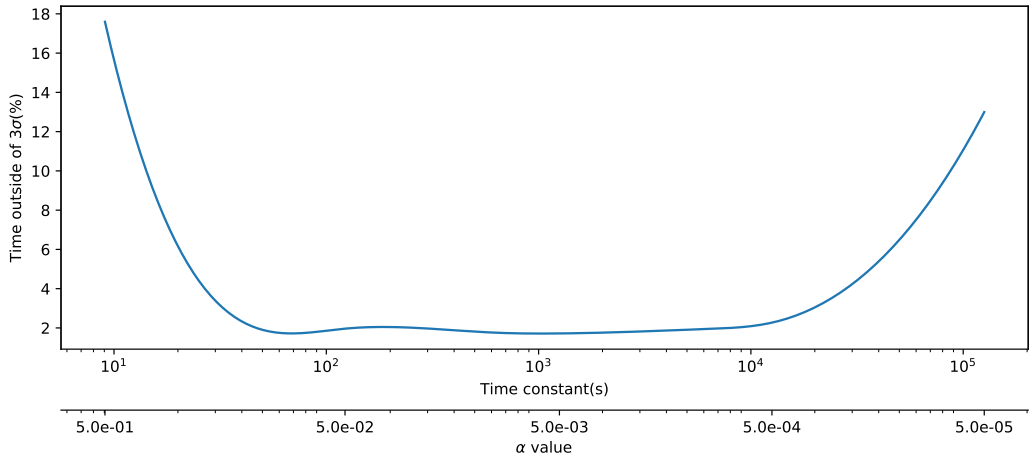


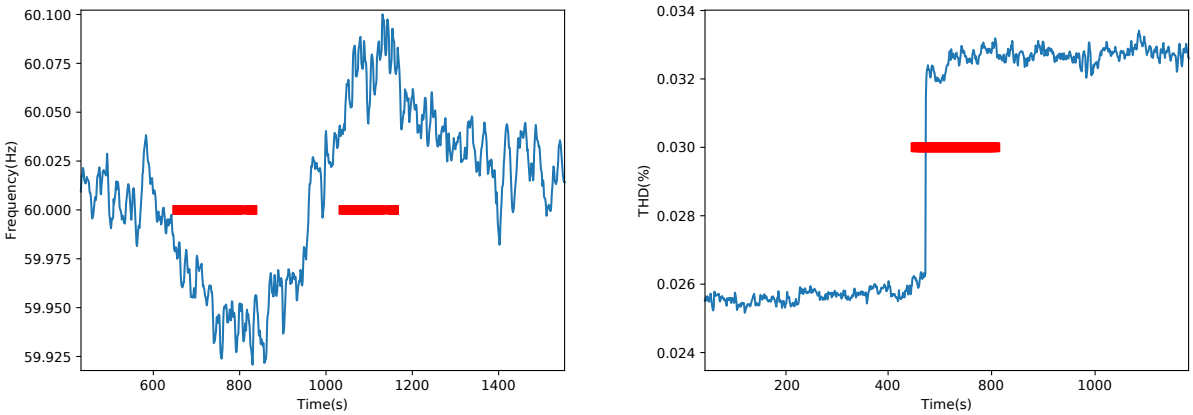
Figure 4.9: Amount of time a metric spends outside of the 3σ for various values of α

Blue traces show metrics that exhibit anomalous behaviour, while red indicates that Napali has flagged this temporal region as a sub-threshold event candidate. Figure 4.10a shows a frequency fluctuation which nearly passes the threshold of $60.1Hz$ which would mark it as a full fledged event. Instead, Napali marked almost the entirety of the fluctuation as a potential sub-threshold event, as shown by the red trace. Figure 4.10b shows a step in the total harmonic distortion metric, similar to the one shown in Figure 3.7 at the 6am mark. In this case the metric in question abruptly changed to a new mean, requiring a fairly slow α coefficient to catch up over 3 minutes. While it may seem wasteful to mark large temporal regions following an abrupt jump as candidates for sub-threshold event, it is important to note that:

1. Making the α parameter smaller does not benefit the false positive rate as shown in Figure 4.9.

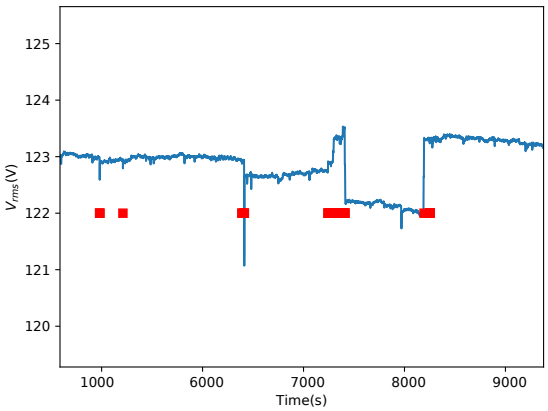
2. In-situ there is no way to tell if an abrupt shift is a switch to a new steady state, or if the metric will recover to a previous mean.

It is important to remember than Napali is not meant to have a low false positive rate. Instead, a system like OPQ Mauka can use all available information, including the raw data, to determine if an event is true gridwide event with much higher confidence. The main goal of Napali is to have an extremely low rate of false negatives. Figure 4.10c is on a different timescale from Figures 4.10b and 4.10a. This is done in order to include several potential sub-threshold events into a common chart. Five temporal regions during the the 3 Hrs are marked by Napali as potential sub-threshold events.



(a)

(b)



(c)

Figure 4.10: Potential sub-threshold events for a) $f_{fundamental}$, b)THD, and c)V_{rms} Red boxes indicate that Napali picked these temporal windows as a potential sub-threshold event.

4.2.2 Deployment nomenclature

In this section I describe the nomenclature used throughout Napali evaluation. Since this deployment is unique, naming convention of particular deployment characteristics and events is unique to this paper.

All meters, both Utility and OPQ Box fall into three categories shown in Figure 4.11. Category 1 are collocated devices. In this situation the Utility meter is monitoring the the $480V_{rms}$ three phase line going into the building. A transformer converts the $480V_{rms}$ to the household $120V_{rms}$. A deployed OPQ Box monitors the $120V_{rms}$ line. This category of devices is particularly important because they provide a baseline for comparison between the OPQ Box performance and a commercially installed system. The second category of deployed devices are the non-collocated OPQ

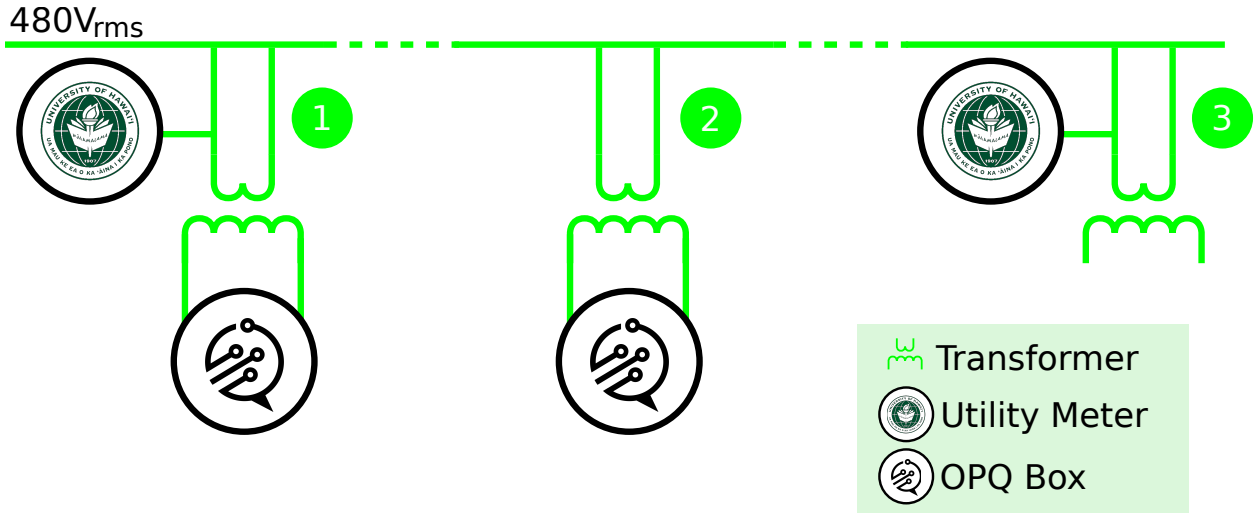


Figure 4.11: Three categories of device deployments.

Boxes. These devices are deployed in buildings that lack smart meters. Thus, we lack ground truth data for these devices. However, these devices are still useful for subthreshold triggering studies. Finally, category 3 consists of non-collocated Utility meters. These meters monitor locations without a deployed OPQ device. Similarly to locations in category 2, data from these devices proved useful in subthreshold event detection evaluation.

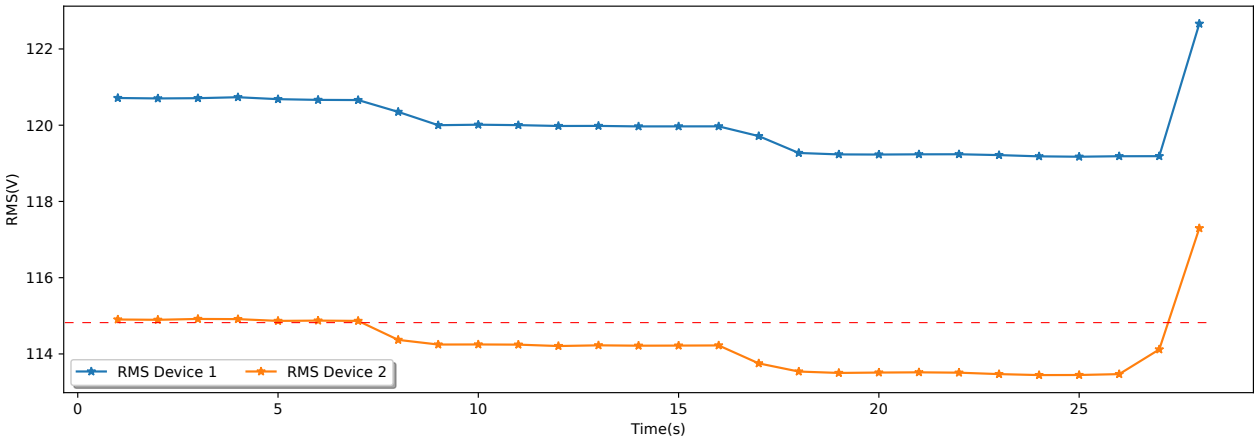
There are several types of events that Napali is able to detect. The first category are gridwide events. Gridwide events affect every device on the OPQ network, with each device passing one of the designated threshold. Gridwide events inherently lack a subthreshold component since every device captures over-threshold data. Partial gridwide events are events that affect only a subset of devices enough to pass the threshold and trigger data collection. One of the main goals of Napali is to utilize metric extraction in order to detect sub-threshold events. During the deployment, two types of sub-threshold events were identified:

- Partial sub-threshold events.
- Full sub-threshold events.

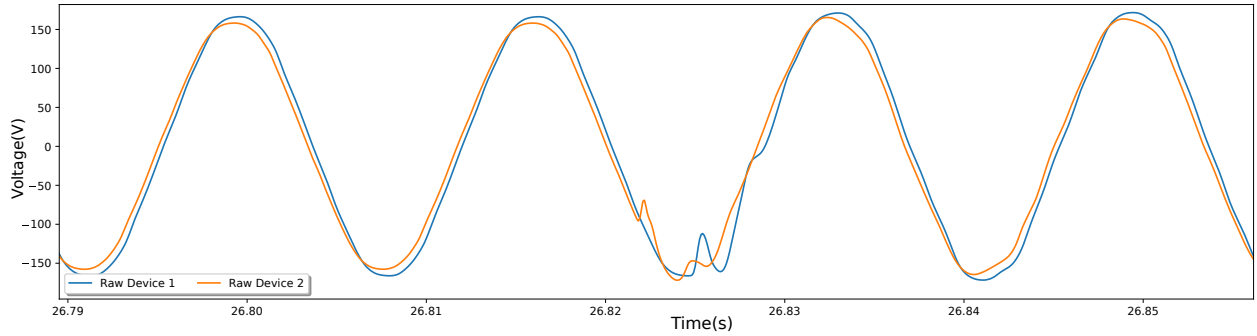
Full sub-threshold events consist of one or several devices passing the threshold described in Table 3.7, as well as one or several devices marked as sub-threshold by Napali. Partial sub-threshold events consist of devices which all passed the threshold described in Table 3.7, however some of the devices triggered on a different metric with a much shorter temporal window. The important distinction between partial sub-threshold events and full sub-threshold events is that if triggered using the Self-Triggering method, the majority of the sub-threshold data would be lost.

One would expect that partial gridwide events would be divided into those with and without a subthreshold component, however, none of the events captured by Napali during the deployment

consist of only over-threshold waveforms. As such, all of the Napali partial gridwide events are also subthreshold events, and the term is used interchangeably.



(a)



(b)

Figure 4.12: Partial sub-threshold event a) the sub-threshold component of the event, b) above threshold component of the event

An example of a partial sub-threshold event is shown in Figure 4.12. Figure 4.12a shows the sub-threshold component of the event. In this event device 2 passed the threshold on V_{rms} metric, initiating Napali to look for sub-threshold events across other devices. It is important to note, that at the event start device 1 was considered to be a sub-threshold candidate, however, at $t \approx 26.8s$ device 1 produced a transient metric which was above Napali threshold. As such Napali requested raw data from both device 1 and device 2, creating a partial sub-threshold event containing both a voltage sag shown in Figure 4.12a and a transient shown in Figure 4.12b.

An example of a full sub-threshold event is shown in Figure 4.13. This is a short-lived transient event observed by four devices on September 5^{th}. Devices 1, 3, and 4 generated a transient metric higher then the Napali threshold. Device 2 transient metric did not pass threshold, yet nonetheless produced a severe enough deviation from the mean for Napali to consider it a part of the event. This is particularly evident in the mild transient observed in Figure 4.13b.

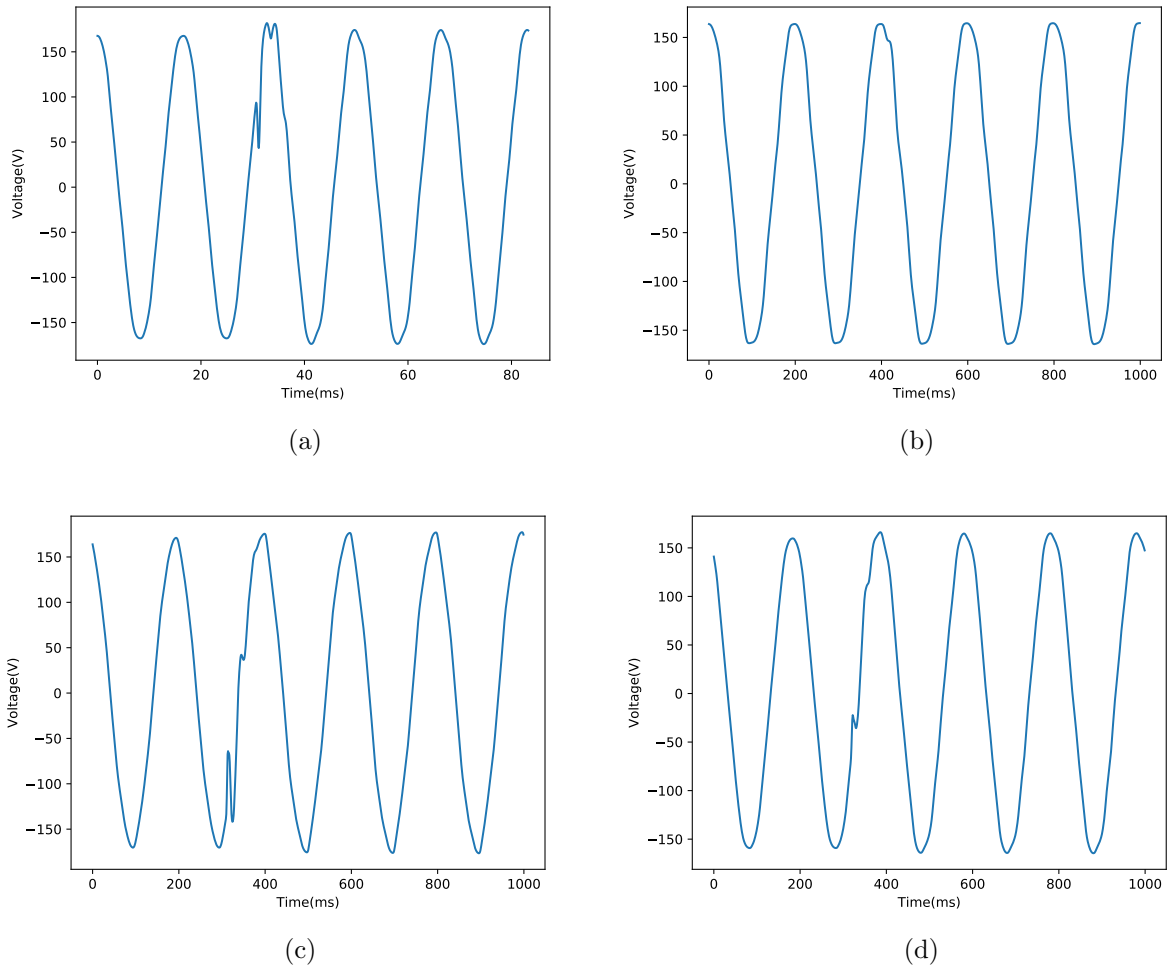


Figure 4.13: Full sub-threshold event across 4 devices. a) Device 1: above threshold b) Device 2: sub-threshold c) Device 3: above threshold d) Device 4: above threshold.

4.2.3 Event Dataset

Event dataset used in this analysis consists of data from 15 devices collected between November 14th and January 1st 2019. There are 2163 events in total, with 1378 V_{rms} events and 737 transient events. 6 Events were related solely to frequency and 83 related solely to total harmonic disturbance. During the deployment, OPQ network achieved 97% availability with only downtime relating to software updates and a power outage November 25h power outage.

4.3 Results of experimental evaluation

Napali was validated using simulation, synthetic data with the device-in-loop, and in-situ during the deployment. In this section I present the results of the evaluation of the claims of my dissertation as discussed in Section 1.6.

4.3.1 Napali Bandwidth usage

During the OPQ deployment it was found that Napali significantly outperformed both the Self-Triggered and the Naive event detection methods. In order to evaluate the bandwidth performance of Napali, a Self-Triggered plugin ran along-side it inside the Makai host. This plugin utilized the same thresholds as Napali as described in Table 3.7. However, the Self-Triggered plugin did not take into the account any inter-device signatures. This method is equivalent to each device performing non-collaborative triggering.

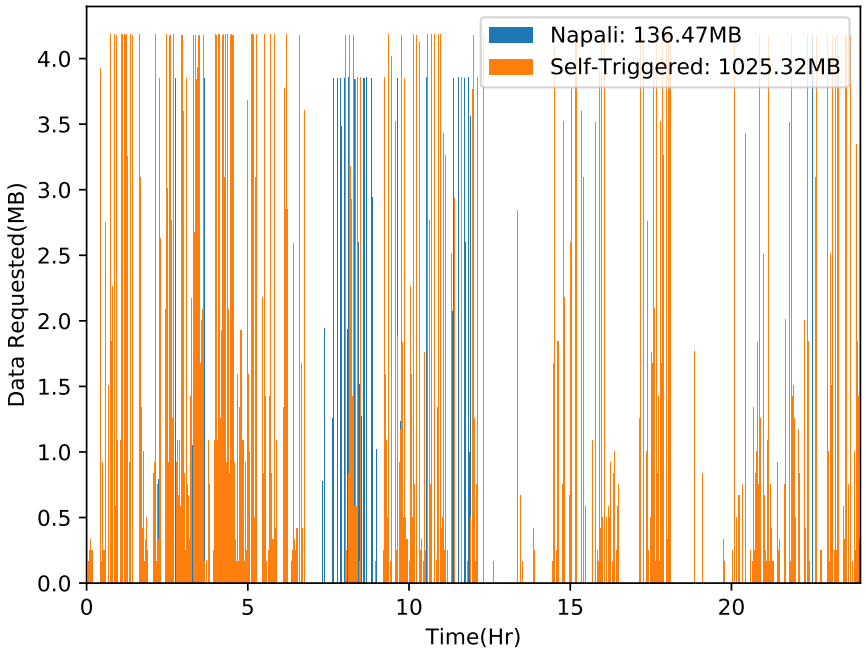
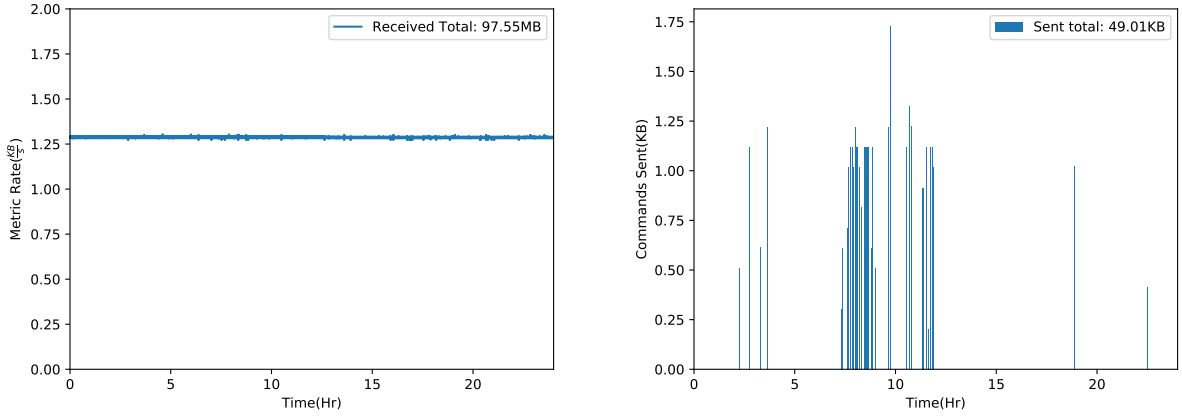


Figure 4.14: Amount of data requested from 10 OPQ Boxes via the Self-Triggered and Napali methods.

Figure 4.14 shows the amount of data requested from 10 devices by the Self-Triggered and Napali plugins over 24 Hours. It is evident that the majority of data requests for the Self-Triggered method resulted in local noise, and did not contribute to the grid measurements. Napali on the other hand, ignored anomalies which did not affect more then a single device, while requesting

sub-threshold data during a gridwide PQ event.



(a) (b)

Figure 4.15: Penalties incurred by the Napali framework. a) Metrics received from 10 OPQ Boxes. b) Commands sent to 10 OPQ Box

It should be noted that the Self-Triggered method does not incur the penalty of having to constantly transmit the device metrics to the sink, since all the event detection is performed on the device. Figure 4.15a shows the amount of data received via metrics from 10 OPQ Boxes during the same 24 hours as the Figure 4.14. As expected the bandwidth requirement for metric transmission remains constant, since all OPQ Boxes send the metrics at fixed intervals. While this penalty is significant as it constitutes 41% of the total bandwidth used by Napali, the aggregate bandwidth is still shows a 440% improvement over the Self-Triggered method. Another penalty incurred by Napali is the two way communication requirement. Each device which participated the event detection needed to receive a command with the temporal range with anomalous data. Neither the Naive nor Self-Triggered methods require two-way communication, and as such there is no direct comparison to Napali. Figure 4.15b shows the command bandwidth consumption for 10 OPQ Boxes across 24 hours. The total consumption was 50kB, which is quite trivial for any modern sensor network.

During the 24 hours shown in Figures 4.15 and 4.14, Napali captured 60 events, while the Self-Triggered method captured 878. The average length of the Napali Event was 10s to the Self-Triggered 3s. Of 60 Napali events, all 60 contained sub-threshold data.

Comparison of Napali with the Naive method was performed analytically. Since the sampling rate of the OPQ Box is well characterized, and the number of OPQ Boxes is fixed, it is trivial to calculate the amount of raw data generated by the OPQ network during any time period. In order to make this comparison fair, the raw data bandwidth will be scaled by the compression ratio of the state of the art compression algorithm specifically designed for power quality measurements.[43]

Operating at 12kSps, OPQ Box produces raw data at 24KB/s. With state of the art compression operating at 90% compression ratio and 5% overhead of meta-data, one can expect a 3KB/s stream of raw data for each OPQ box if it were to send the entirety of it to the sink. For 10 OPQ Boxes we would expect the aggregate bandwidth of 30kB/s, and as such the bandwidth consumption of 24.7GB/day. During a 24 hour period, as shown in Figures 4.15 and 4.14, Napali used 234MB of bandwidth. This corresponds to an over 100x improvement over the Naive method.

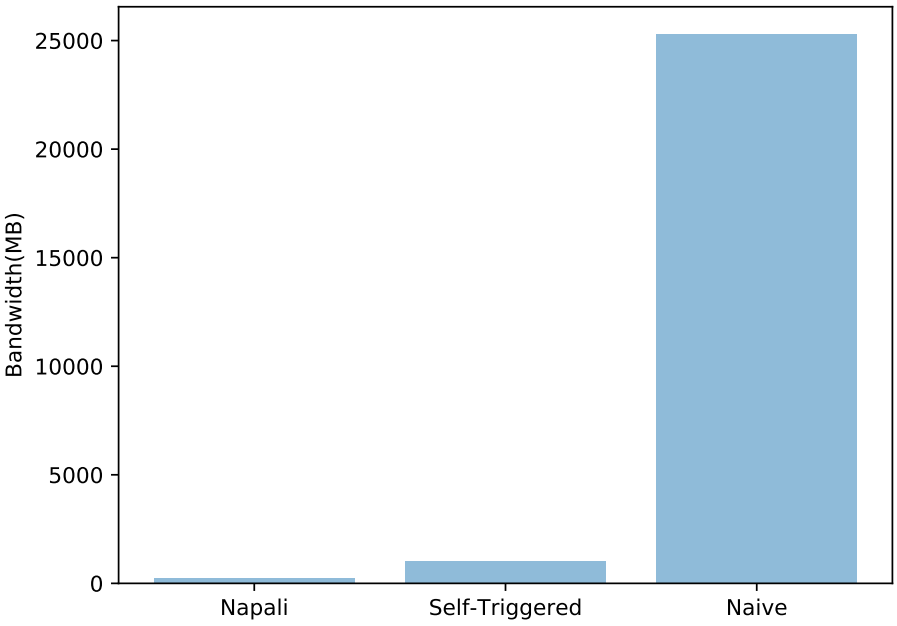


Figure 4.16: Bandwidth requirement comparison between three event detection methods.

Figure 4.16 shows the comparison between Napali, the Naive, and Self-Triggered methods. While Napali incurs additional costs described in Figure 4.15, it outperforms the comparable methods. Finally, the cost of the two way communication as shown in Figure 4.15b is greatly outweighed by the bandwidth savings in the raw data reception. Modern sensor networks greatly benefit from two way communication, as it allows on-demand health monitoring, and software updates. With addition of Napali, two way communication allows for significant bandwidth requirement reduction in for the sensor network as a whole.

This result validates the claim described in Section 1.6.1. The Napali event detection method significantly reduces the bandwidth cost associated with power quality event detection.

4.3.2 Sink processing requirement under the Napali Framework

Sink processing requirements for event detection between Self-Triggered, Naive and Napali are quite different. In general the processing requirement can be described as follows:

$$C_{total} = C_{metric\_extraction} + C_{detection} \tag{4.2}$$

In the Equation 4.2 the C_{total} is the total cost, $C_{metric\_extraction}$ is the cost of extracting metrics and $C_{detection}$ is the cost of event detection.[11] Each of the three methods, Napali, Self-Triggered, and Naive, have different sink costs associated with each parameter.

Sink processing: Naive Method

First, let's consider the Naive method. In this case, all of the metrics need to be extracted at the sink. Disregarding the processing power required to keep up with the data rate described in Section 4.3.1, the $C_{feature\_extraction}$ can be measured empirically. In order perform this measurement, OPQ Box software was built for an x86 architecture and stress-tested. Instead of acquiring data from a device driver, the feature extraction stack was supplied with synthetic data. Finally, the ZMQ communication was removed and replaced with the performance analysis code. Stress test was performed on an Intel Core i9-8950HK CPU with thermal management disabled running at 2.9GHz. Under such conditions, the metric extraction stack was able to extract features from 1s worth of raw data in $800us$ running on a single core. Since metric extraction has no inter-device data dependencies, a modern 8 core CPU can expect to keep up with feature extraction from 1000 devices. If an OPQ Box sensor is used with 16 bit samples and 12kSps ADC, aggregate bandwidth for such system is 10.8Gbps, which is well within the realm of a collocated server with dual 10Gbps network interfaces. $C_{detection}$ cost can be made linear with the number of devices. If a rolling window is applied to metrics as they are generated, raw data from all devices contained in the window with an offending threshold metric can be retained for later analysis. While simple, this method will collect all of the gridwide events along with a large number of false positives. In synthetic benchmarks, the $C_{detection}$ made up less then 0.01% of the computational cost when compared to $C_{feature\_extraction}$ and does not significantly contribute to the C_{total}.

Sink Processing: Naive Method

In contrast to the Naive method, the Self-Triggered method has no sink processing requirements, since all of the feature-extraction is performed on the edge device. Thus, Self-Triggered method event detection is only limited by the available network bandwidth.

Sink Processing: Napali Method

Napali, being a hybrid of Naive and Self-Triggered methods, moves the $C_{metric\_extraction}$ cost to the edge devices, while retaining the $C_{detection}$ at the sink. Unlike the Naive case, Napali performs additional computations on the features in order to detect sub-threshold events while excluding local noise. However, even with additional metric analysis the Napali stack was able to process synthetic data from 100000 devices on a single core of an a Intel Core i9-8950HK CPU. This would allow a single server running Makai to provide 50% coverage of households in the city of Honolulu.

Sink Processing: Event Classification

The final step of any power quality analysis stack is event classification. Every event collected by an event detection system must be analyzed and classified according to their severity and type. While Makai/Napali are not responsible for event classification, it is important to consider event classification cost when discussing sink processing requirements. In the case of the Naive method, events which are detected are a mix of local and global events. However, every event will contain a waveform from every device on the network. For the Self-Triggered method, only the events which cross the threshold will be considered for classification. While Self-Triggered events will not contain false positives and sub-threshold events, vast majority of acquired waveforms are comprised of local disturbances. Finally, Napali produces high quality events which only contain high fidelity sub and over threshold events, while ignoring local disturbances. During the campus deployment Napali detected 302 Events comprised of 1561 individual device waveforms a week on average. The Self-Triggered method detected 26520 offending waveforms. If we assume unitary classification cost, classification computational requirements for one week of data are shown in Figure 4.17.

This result validates the claim described in Section 1.6.3. Napali greatly reduces the sink processing requirements when compared to the Naive method. While no direct comparison can be made with the Self-Triggered method, the volume of data generated by it will have an adverse effect on event analysis resource consumption compared to Napali.

4.3.3 Effects of latency in the Napali framework

In order to understand the effects of latency on Makai in the OPQ deployment, we examined the event length and round trip latency for OPQ Boxes. These parameters are shown in Figure 4.18.

Figure 4.18a was generated from 1000 events or about a month of data. As depicted, the majority of events captured by Napali were less then 20s in length, with a few stragglers hitting a 40sec mark.

Figure 4.18b was generated by requesting a 20s event from all OPQ Box devices and timing the amount of time it takes them to respond. Since this measurement can be performed synthetically

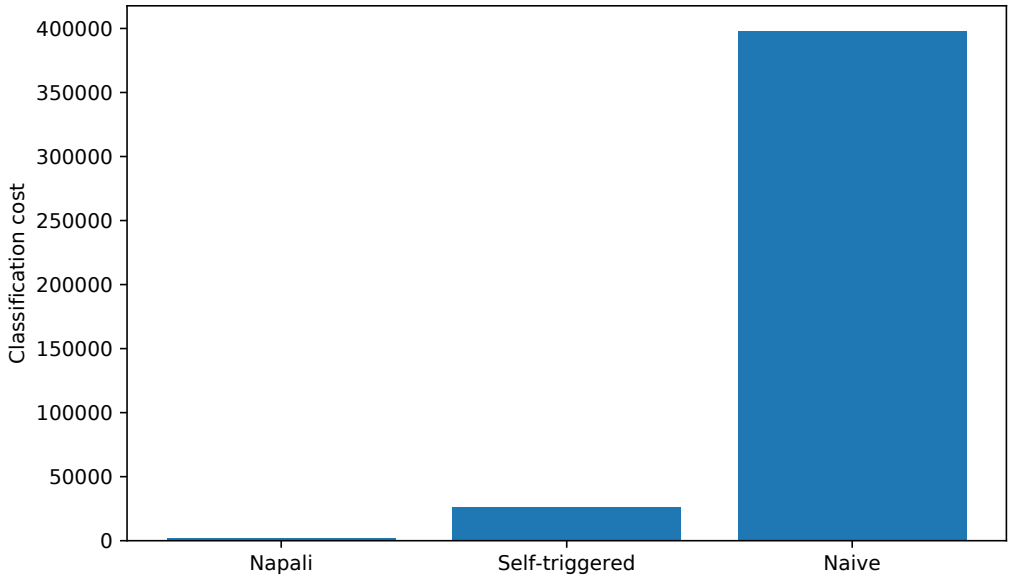
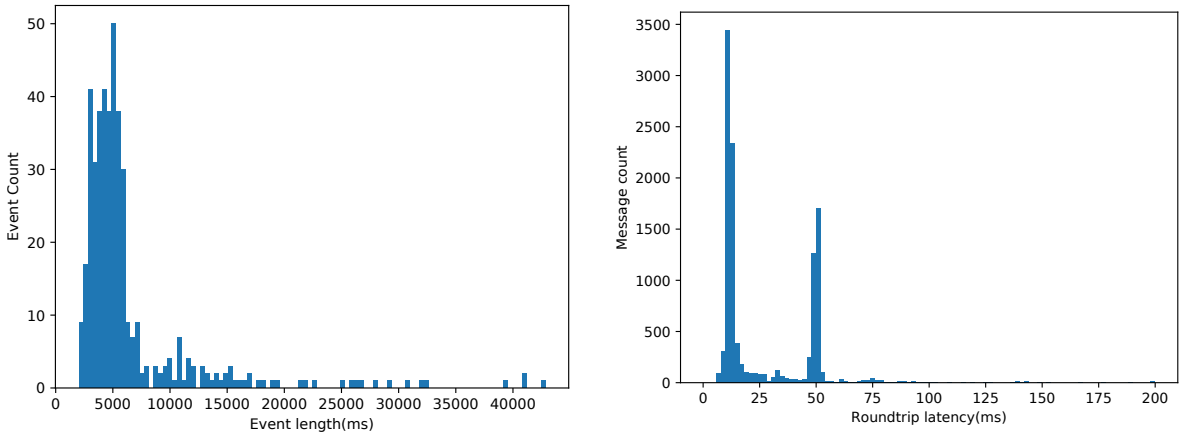


Figure 4.17: Classification cost based on the expected amount of waveforms for the three considered methods.



(a) (b)

Figure 4.18: Event length(a) and message latency(b) observed by the OPQ devices.

without waiting for an anomaly, 10000 samples were used, which equates to approximately ten months of real world measurements. Latencies clustered into two groups, likely based on the number of retransmission required for the senor to reach the router, with means of 12.5ms and 50ms. A few stragglers were observed at at 200ms and a single device had a latency of 1.2s, and was omitted from Figure 4.18b for clarity.

Since OPQ Boxes communicate via WiFi it is common for them to lose connections to the access point, and reconnect some time later. This behaviour can be quantified in the OPQ network by examining the trend dataset. Trends are generated for each device at 1 minute intervals. By

comparing the database insertion time and the timestamp reported by the device, it is possible to find trends which were delayed in transit. Out of two months of data only 73 delayed trends were reported, and are shown in Figure 4.19. While the majority of trends were delayed by less then 60 seconds, one was delayed by 3 minutes.

OPQ Box does not utilize compression in the RDRB. The maximum safe size of RDRB was found to be 100MB of 256MB of total system memory. This way there is enough memory left over on the device to prepare the data for transmission in the extreme case where Makai requests the entire buffer. Given that the sampling rate and data size is fixed, 100MB of RDRB can store about 1.1 hours of raw waveform. As evident from Figure 4.18, under normal operating conditions, even with degraded latency, no event will overrun the RDRB. Even with the longest observed event/worst observed latency, the maximum delay is on the order of 50s, or slightly longer then 1% of the maximum tolerated latency. Under abnormal conditions, such as a WIFI drop, the maximum delay may be on the order of 5 minnutes, which is on the order of 8% of maximum tolerated latency.

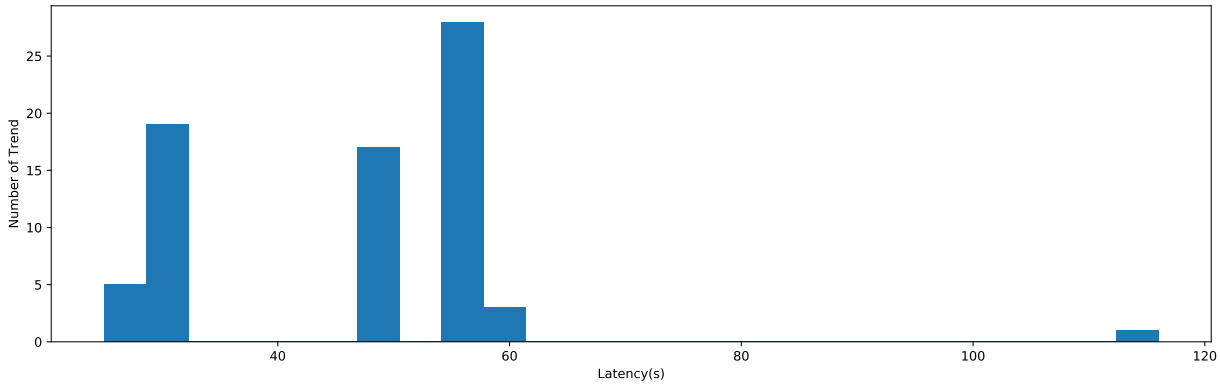


Figure 4.19: Delay in trend creation, indicating a network failure.

Device latency has a different effect on Napali, Naive and Self-Triggered methods. The Self-Triggered method is completely unaffected by latency, since event detection is performed entirely on the device. The Naive method is affected the most, since the sink performs all of the event detection computations. In this case, if latency is asymmetric across the devices, and the device with the highest latency observes an over-threshold condition, there is potential for waveforms from the rest of the network to get falsely discarded. Napali only requires metrics for anomaly detection, and as such, the sink does not need to buffer raw data in anticipation of a potential event.

The Naive method must buffer enough raw data to be able to handle network failures and interruptions. If a device with an over the threshold metric is delayed, the Naive method must keep raw waveforms for all devices in order to store them for later analysis. In the case of OPQ, as described in Figures 4.19 and 4.18, the maximum observed latency may be on the order of 5 minutes. This implies, for each device Naive method must buffer 7MB of raw waveform per device. In reality, for large deployment both the rate of network failure, and latency is expected to be higher

then those reported on the UH network.

In the case of Napali, only the metrics need to be buffered for event detection. An optimal amount of metrics storage is determined by the worst case latency and the maximum expected event size However, since the overhead of buffering metrics is so small there is no reason to not maintain a measurement buffer appropriate to accumulate the maximum delay that the OPQ Box can tolerate. This limit is governed by the amount of data that the RDRB can store, thus even if an over the threshold metrics are transmitted the delay slightly less then the RDRB capacity, Napali will still acquire the raw waveform. The amount of memory required for each device can be derived by analysing the metric size and the frequency of metric transmission. In the case of OPQ, each measurement is 24bytes in size and they are transmitted at the rate of 1 measurement per second resulting in the total memory consumption of 84kB for 1 hour of metric storage. A modest server can support Napali/Makai with 100000 devices while utilising 8GB of memory for 1 hour of metric storage, a 100 times improvement over the Naive method only buffering 5 minutes of raw waveforms.

This result validates the claim described in 1.6.2. Napali is able to mitigate device latencies of up to the device buffer size with no adverse effects, while the Naive method struggled to cope with short disturbances. No direct comparison to the Self-Triggered method is made, since latency is not a factor of Self-Triggered event detection.

4.3.4 Summary of Computational and Network Resource Utilisation

Sections 4.3.1, 4.3.2 and 4.3.3 describe the computational utility of the Napali framework, when compared to the Naive and Self-Triggered methods. Using the insights described above we can compare the number of devices that the three methods in question can support on a modest collocated server. Our hypothetical server is a 4 core machine, with each core equivalent to an Intel i9-8950HK we have been using for benchmarking. It is further equipped with 8GB of random access memory and a single symmetric 1Gb uplink to OPQ devices. For comparison we consider the three metrics described above: bandwidth, computational cost and memory utilization. When comparing network computational cost, the classification cost is not considered, since these results illustrate the cost of the event detection only. Notice, that we use linear scaling for Napali event rate. Napali uses a statistical approach, in order to locate subthreshold events. Intuitively, this statistical approach would not scale linearly, since as the number of devices in the network increases, the probability of multiple devices observing unrelated subthreshold event increases as well. However this can be mitigated using device clustering as shown in Section 4.4.1, which results in linear scalability.

Table 4.1: Method comparison for a typical collocated server: Bandwidth

	Napali	**Self-Triggered**	**Naive**
# of devices	44000	10000	800

Table 4.1 describes the bandwidth limitation across the three methods, assuming no overhead and 100% utilization on the 1Gb link. Napali is a clear winner in this case, since it is able to to support significantly more devices before becoming bandwidth limited.

Table 4.2: Method comparison for a typical collocated server: CPU

	Napali	**Self-Triggered**	**Naive**
# of devices	400000	∞	1000

Table 4.2 describes the CPU limitation across three methods, assuming 100% CPU utilization on all 4 cores. In the Self-Triggered method, event detection is performed entirely on the device, so it incurs no CPU cost. However, the cost of event detection with Napali is so low, that it hardly affects it's utility. Naive method fairs the worst between the three since it relegates all of the event detection cost to the sink.

Table 4.3: Method comparison for a typical collocated server: Memory

	Napali	**Self-Triggered**	**Naive**
# of devices	100000	∞	1000

Table 4.3 describes the memory limitations across the three methods, assuming 100% memory utilization. Again, with no sink requirements, the Self-Triggered method requires no memory buffer for event detection. Napali buffers 1Hr of metrics in order to accommodate on devices with excessive latency and handle network faults. Naive method must maintain a buffer of raw waveforms on the sink, which leads to a memory bottleneck while only maintaining a 5 minute buffer.

Table 4.4: Method comparison for a typical collocated server: Worst of all metrics

	Napali	**Self-Triggered**	**Naive**
# of devices	44000	10000	800

Table 4.4 describes the final tally across the three methods. It illustrates how many devices our hypothetical collocated server can handle before becoming limited in one of three metrics. All three

methods are limited by the bandwidth, however, even with additional bandwidth the Naive method would quickly run out of computational resources to keep up with the data stream. As such, when it comes to efficiency of gridwide monitoring Napali is a clear winner suitable for deployment across a large portion of grid endpoints. Self-Triggered method is close second, but as we will see in the next two sections, it performes poorly when it comes to detection efficiency. Finally, Naive method is the most computationally expensive at the sink. However, it is also the most robust when it comes to event detection, both local, gridwide, and subthreshold. As such, the Naive method is best left for monitoring high value infrastructure such as substations and production centers which can afford the additional computational and bandwidth cost.

4.3.5 Temporal locality triggering of the Napali framework

As mentioned previously, during the University of Hawaii deployment the OPQ network worked alongside the utility grade meters embedded in the University power delivery infrastructure. Data from these devices provided ground truth when it came to evaluating OPQ power quality anomaly detection. Before we delve deeper into that, I will compare the OPQ Box and utility meter metric extraction capabilities.

Utility meter metric extraction comparison

Eight OPQ devices were collocated along with smart utility meters capable of logging power quality metrics. This analysis focuses on Device 1000 along with a smart utility meter labeled POST_MAIN_2 which is located upstream of Device 1000. POST_MAIN_2 is monitoring a 480V line going into the Pacific Ocean and Science Building prior to its conditioning and step down into the domestic 120V.

Frequency: A plot of 1 week of frequency measurements collected by the OPQ Box 1000 and POST_MAIN_2 is shown in Figure 4.20.

OPQ Box and the utility meter track the fundamental frequency across all devices throughout the entire deployment. This is expected since the fundamental frequency must be stable across the entire grid in order for it to operate properly. This parity is further demonstrated in Figure 4.21 where the frequency recorded by the two devices was subtracted from one another. These differences were histogramed and statistically analyzed.

Synthetic benchmarks showed the OPQBox frequency measurement capability on the order of 200x better then the error reported in Figure 4.21. This is likely due to the fact that both comparisons are done over 1 minute averages of frequency, and the offset of where the minute average starts is not well described in the utility data. Regardless, with the power quality threshold of $0.1Hz$ an error of two devices with $\sigma = 8mHz$ is perfectly acceptable to draw our conclusions regarding the Napali temporal trigger efficiency.

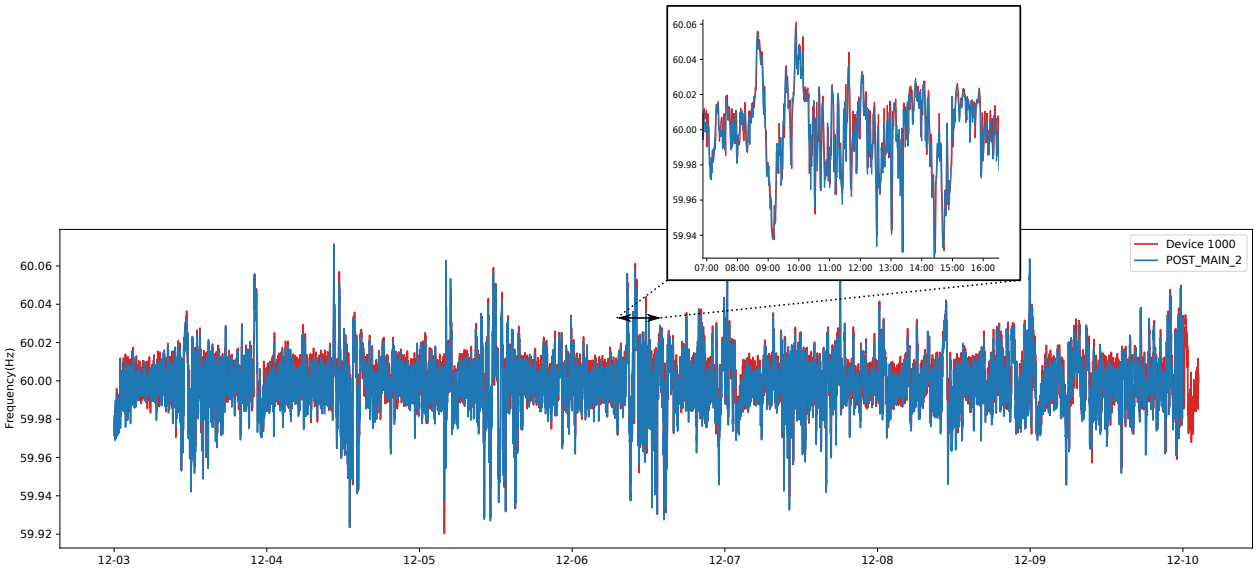


Figure 4.20: Frequency metric for the POST_MAIN_2 utility meter and OPQ Box 1000.

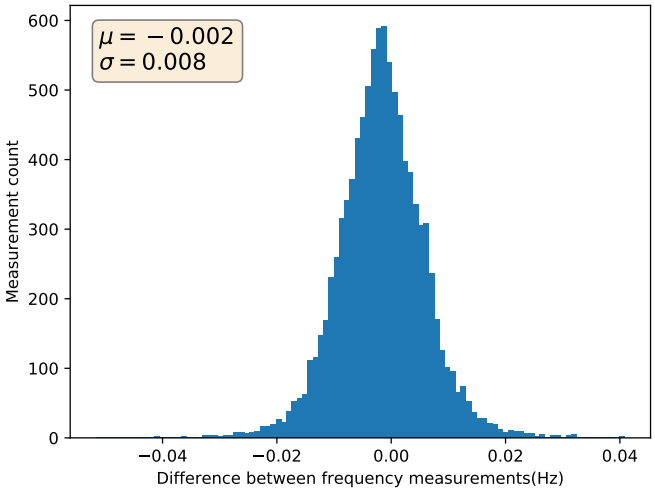


Figure 4.21: Difference in the frequency metric between POST_MAIN_2 utility meter and OPQ Box 1000.

THD: A plot of 1 week of THD measurements collected by the OPQ Box 1000 and POST_MAIN_2 is shown in Figure 4.22.

Unlike the frequency measurements, THD shows anomalous behaviour during the daylight hours. Particularly, from 6:00-18:00 daily the THD measurement diverges by a fraction of a percent between the devices. Since the POST_MAIN_2 meter is located upstream of Device 1000 on the power grid hierarchy, there is likely a reactive power compensation or a THD filter system posi-

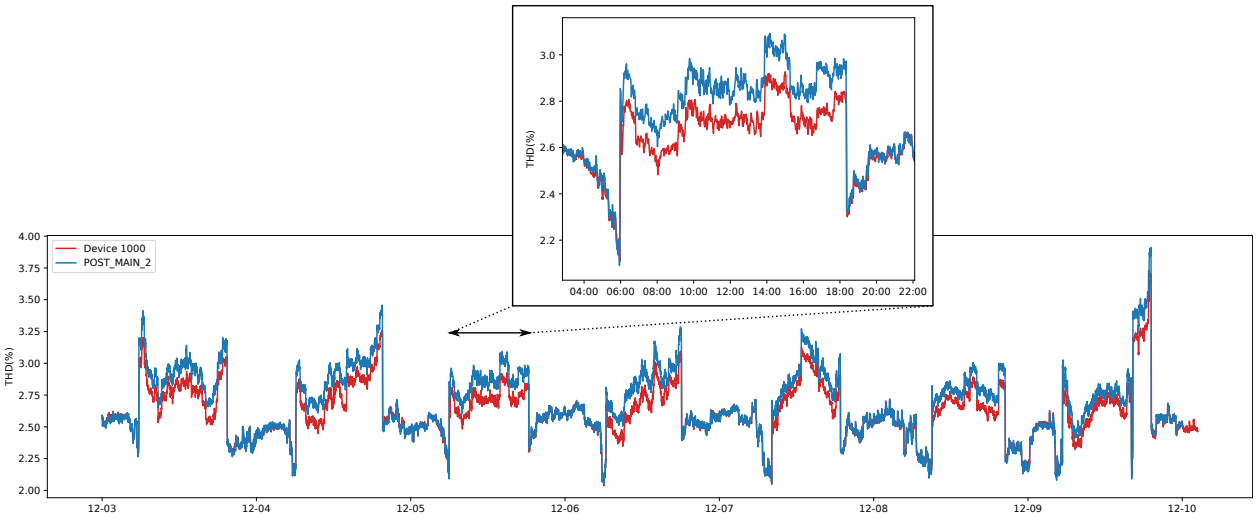


Figure 4.22: THD metric for the POST_MAIN_2 utility meter and OPQ Box 1000.

tioned between the two devices. The fact the the minute differences in THD are tracked equally well, albeit with an aforementioned offset is particularly interesting. This fact is further illustrated in Figure 4.23.

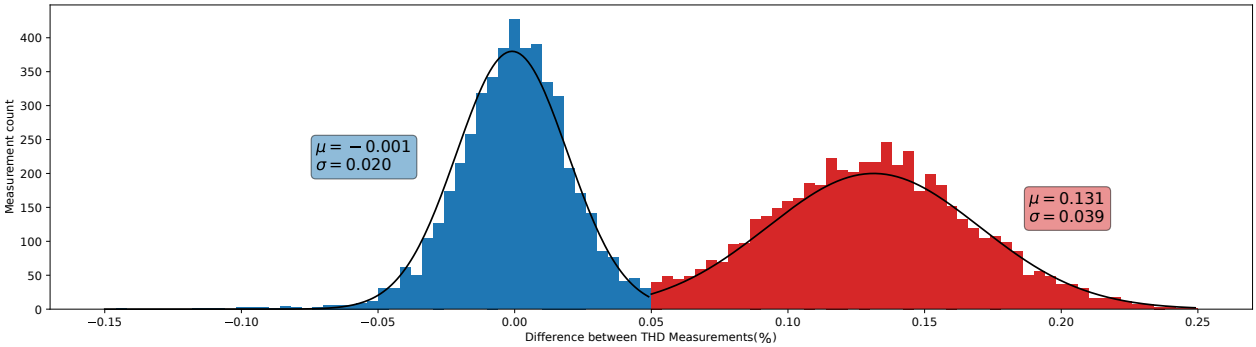


Figure 4.23: Difference in the THD metric between POST_MAIN_2 utility meter and OPQ Box 1000.

Figure 4.23 shows the histograms of the difference in measurement between the OPQ device and the utility meter. In blue are the samples occurring during the seemingly coincidental regions of 18:00-6:00. These regions are characterized by an excellent agreement between the two devices, with difference given by $sigma = 0.02\%$. This level of agreement is comparable to the synthetic benchmarks of the OPQ box with a well calibrated source. The red region is the anomalous region, where the two sets of measurements are offset by the $\mu = 0.13\%$. Even though the accuracy suffered, measurement remained precise down to $\sigma = 0.04\%$.

RMS Voltage: A plot of 1 week of normalized voltage measurements collected by the OPQ Box 1000 and POST_MAIN_2 meter is shown in Figure 4.24.

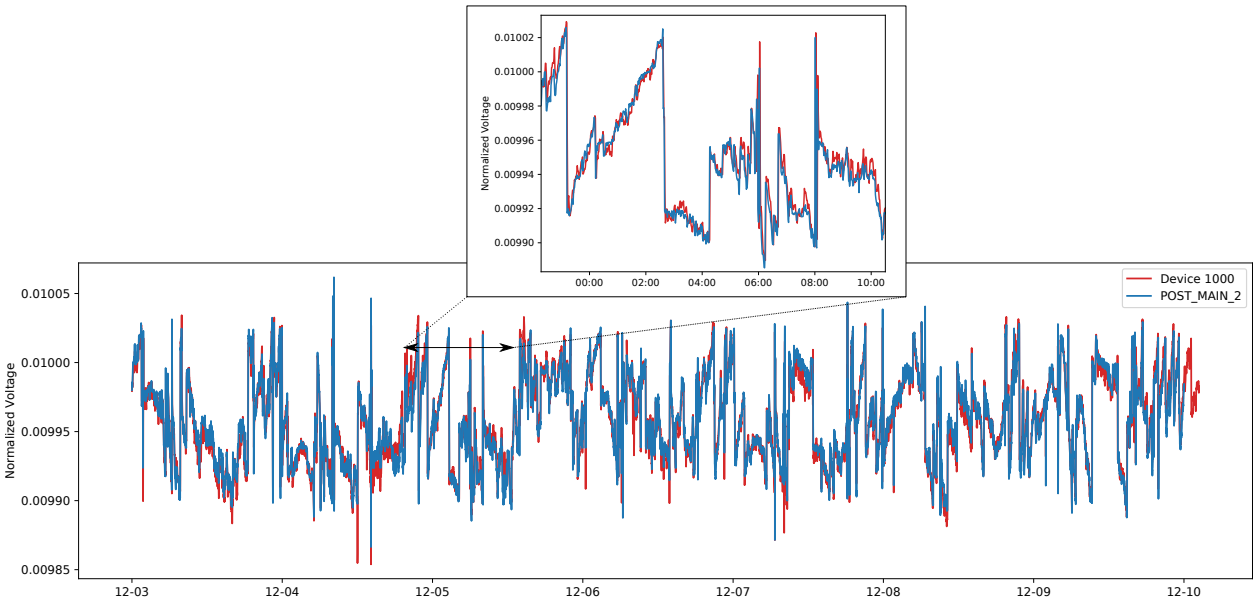


Figure 4.24: RMS metric for the POST_MAIN_2 utility meter and OPQ Box 1000.

It should be noted that the V_{rms} for POST_MAIN_2 shown in Figure 4.24 is not the raw waveform acquired from the meter. The raw data consists of the inter-phase across between each leg of the 3 phase system. Typically, in order to step down a 3 phase 480V system into a 120V single phase a star or a delta transformer is employed. The voltage generated from this transform configuration is a quadrature combination of the three phases [16], thus the voltage displayed in Figure 4.24 was in fact:

$$V_{rms} = \frac{1}{\sqrt{3}C}\sqrt{V_{ab}^2 + V_{bc}^2 + V_{ca}^2} \tag{4.3}$$

where the V_{ab}, V_{bc} and V_{ca} are the inter-phase voltages reported by the meter, and C is a constant dependent on the transformer configuration and the final step down voltage. In order to make the differences stand out across the two scales Figure 4.24 was generated with both data sets normalized. A typical step down turn ratio of 480V to 120V is naturally 4:1, which matches quite well with the measured step down factor extracted from the OPQ data and result of Equation 4.3 of 3.9985:1.

Just as with Frequency and THD metrics, we compared the difference between the reported voltage by the two devices. Equation 4.3 was used along with the empirically measured $C = 3.9985$ in order to preprocess the POST_MAIN_2 dataset. The resulting histogram is shown in Figure 4.25

There were a few outliers, resulting in a poor fit. These outliers were likely due to the timing misalignment in the device data. While overall timing agrees quite well, the meters report their

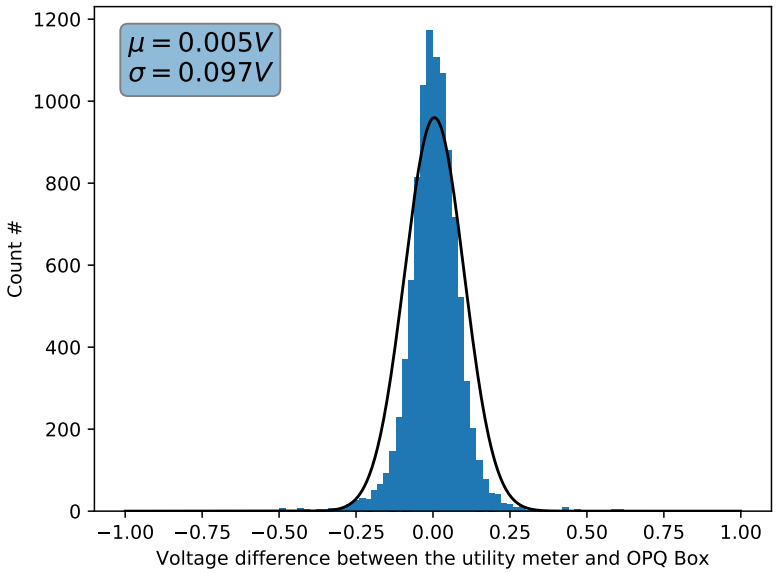


Figure 4.25: Difference in the V_{rms} metric between POST_MAIN_2 utility meter and OPQ Box 1000.

metric at one minute averages, similar to the OPQ Box. However, when the minute begins and ends is quite arbitrary for the OPQ Box, thus resulting in some overlap. Regardless, even with the timing discrepancy the voltage metrics agree to an excellent degree across both devices with a $\sigma = 0.1V$. With the event threshold of $\pm 5V$ this level of agreement was found to be acceptable.

Overall metric comparison: Metrics between the utility meter and the OPQ device were found to correlate to a great degree of accuracy. Frequency measurements in particular was well formed, with a well characterized gaussian with minimal offset and a $\sigma = 8mHz$. Total harmonic distortion tracked very well across the devices, with the difference $\sigma < 0.04\%$ in all cases. However, additional offset was regularly introduced by some form of harmonic filtering equipment. V_{rms} voltage agreement across the two devices suffered the most as expected. There is no way to analytically calculate the step down factor between the 480V three phase system and 120V system without taking apart and measuring the power transformer which powers the POST building. However the empirically measured turn ratio matched closely to the expected value. Furthermore, two meters were in close agreement at all times with $\sigma = 0.1V$ between the two devices. This discrepancy was expected, since both devices are influenced by a different noise sources, with the POST_MAIN_2 meter noise being a superset of that observed by the OPQ Box 1000. The performance of the metric extraction, compared to the utility meter, was found to be satisfactory across the board, making the OPQ Box well suited for grid edge power quality monitoring.

Gridwide event extraction from utility meters

Table 4.5: OPQ Box and utility meter collocation.

OPQ Box ID	Utility Meter
1000, 1002	POST_MAIN_1
	POST_MAIN_2
1001	HAMILTON_LIB_PH_III_CH_1_MTR
	HAMILTON_LIB_PH_III_CH_2_MTR
	HAMILTON_LIB_PH_III_CH_3_MTR
	HAMILTON_LIB_PH_III_MAIN_1_MTR
	HAMILTON_LIB_PH_III_MAIN_2_MTR
	HAMILTON_LIB_PH_III_MCC_AC1_MTR
	HAMILTON_LIB_PH_III_MCC_AC2_MTR
1003	KELLER_HALL_MAIN_MTR
1021	MARINE_SCIENCE_MAIN_A_MTR
	MARINE_SCIENCE_MAIN_B_MTR
	MARINE_SCIENCE_MCC_MTR
1022	AG_ENGINEERING_MAIN_MTR
	AG_ENGINEERING_MCC_MTR
1023	LAW_LIB_MAIN_MTR
1025	KENNEDY_THEATRE_MAIN_MTR

In addition to POST_MAIN_2, several other meters were collocated in the same building as an OPQ Box. Details of this collocation are shown in Table 4.5. Data from these utility meters was used as a basis for evaluation of the temporal locality claim of the Napali framework. Napali temporal locality states that by monitoring the edge nodes of the power grid for temporally related disturbances, the state of the power grid can be determined. In the next section I show this claim to be true, however, prior to that I demonstrate how gridwide events were extracted from the utility meter records.

As shown in Table 4.5 the POST building has two OPQ Devices and two utility meters. Furthermore, several OPQ devices are located in buildings with more then one meter. Unfortunately, no detailed building schematics were made available to us, thus it is not clear which meter was a parent node of the OPQ Box in the power grid hierarchy.

In order to extract anomalies which affected the collocated meters, utility meters from Table 4.5 were queried for metrics from November 15th 2019 to December 19th 2019. These metrics contained minute windowed average, minimum and maximum values for the following power quality measurements:

1. $V_{rms}(AB)$, $V_{rms}(BC)$, $V_{rms}(CA)$: RMS value for the inter-phase voltage.

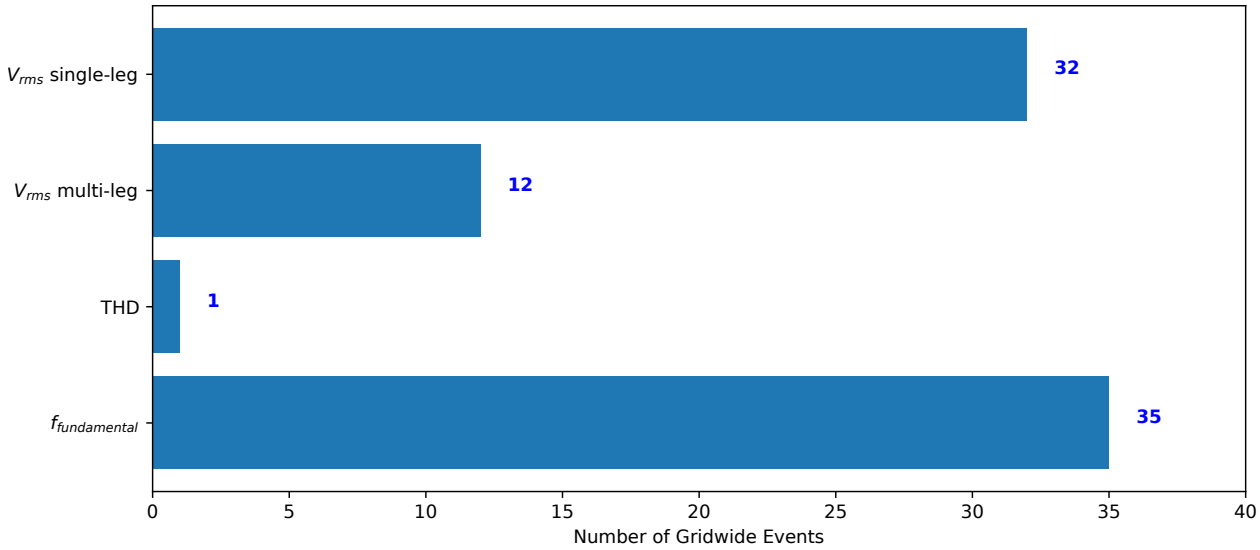


Figure 4.26: Total number of gridwide and subthreshold events extracted from the utility power meters from November 15th to December 19th.

2. $f_{fundamental}$: Fundamental frequency.

3. THD : Total harmonic distortion.

It is unclear which leg of the three phase system the $f_{fundamental}$ measurement was performed on. The THD measurement was likely performed on all three legs and combined together, since it matches closely to the box measurement as shown in Figure 4.22 and 4.23. Finally, in a few meters lacked the $V_{rms}(AB)$, $V_{rms}(BC)$ and $V_{rms}(AB)$ and instead delivered $V_{rms}(AN)$, $V_{rms}(BN)$, $V_{rms}(CN)$ measurements. These measurements are line-to-neutral instead of line to line. In these caseses, instead of using equation 4.3, Equation 4.4 to calculate observed 120V line voltage. [16]

$$V_{rms} = \frac{\sqrt{3}}{4}\sqrt{V_{ab}^2 + V_{bc}^2 + V_{ca}^2} \tag{4.4}$$

Extraction for $f_{fundamental}$ and THD metrics were processed in a similar manner to the OPQ Box events. If the minimum or maximum metric exceeded the threshold defined in Table 3.7 were surpassed, the temporal region was marked as anomalous. If anomalous regions across multiple utility meters were marked, that temporal region was reported as a utility gridwide event. V_{rms} measurements were processed in a slightly different manner. First, the V_{rms} value as observed by the OPQ devices was calculated from both minimum and maximum of the utility metrics using Equation 4.4 or 4.3. These values were then treated in the same manner as $f_{fundamental}$ and THD. Next the individual phases $V_{rms}(AB)$, $V_{rms}(BC)$, $V_{rms}(CA)$ or $V_{rms}(AN)$, $V_{rms}(BN)$, $V_{rms}(CN)$ were thresholded using the same $8\frac{1}{3}\%$ threshold used for the OPQ devices. If more then one utility meter observed a single phase sag or swell it was marked as a utility gridwide event.

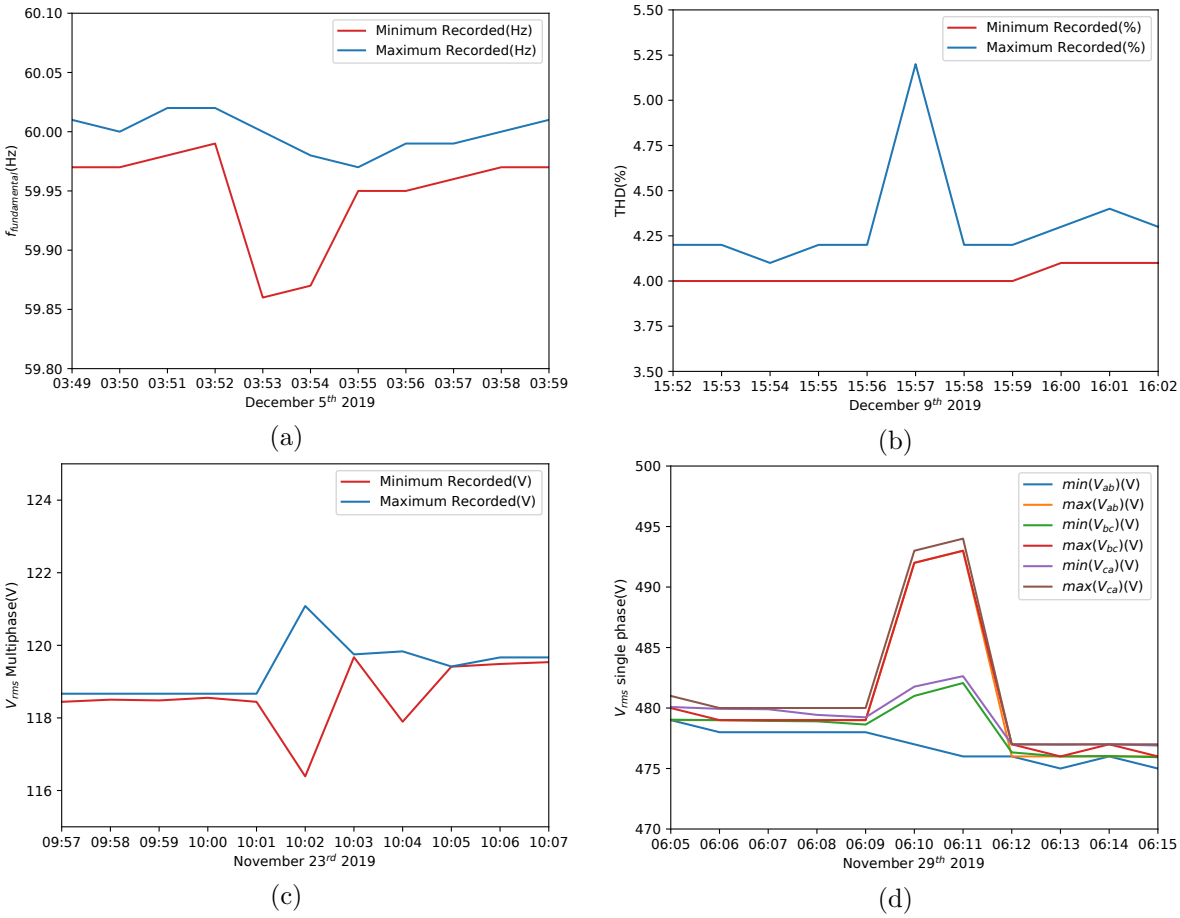


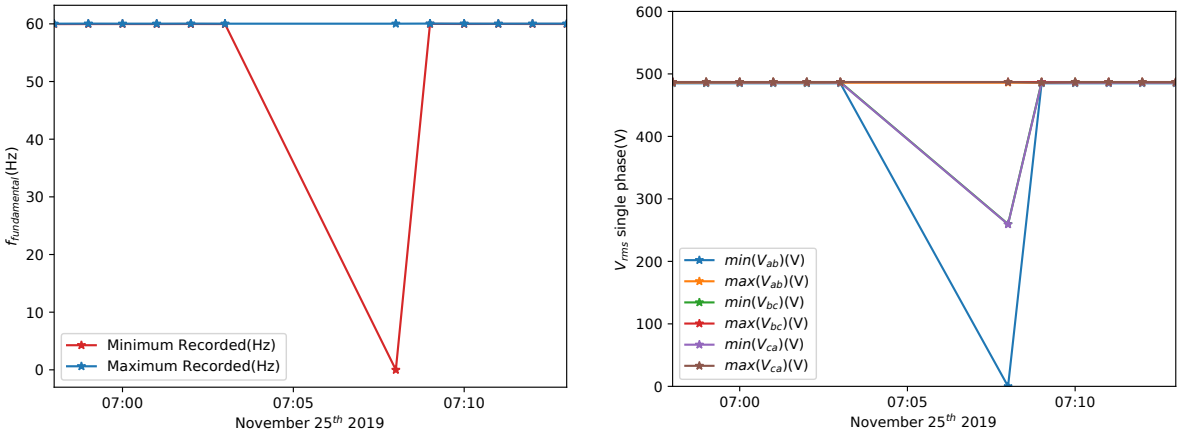
Figure 4.27: Example of the four types of events extracted from utility meter data. a)Fundamental frequency event recorded by POST_MAIN_2. b) THD event recorded by HAMILTON_LIB_PH_III_MAIN_1. c) Multiphase V_{rms} event as recorded by HAMILTON_LIB_PH_III_CH_2. d) Single phase V_{rms} event as recorded by HAMILTON_LIB_PH_III_CH_3.

Total number of events of each type extracted from the utility meters is shown in Figure 4.26. Examples of recorded gridwide events as observed by various meters is shown in Figure 4.27.

Evaluation of detection capabilities for the Napali framework

With the gridwide events extracted from the utility meters in the previous section, a detailed evaluation of Napali detection capabilities was carried out. Events timestamps which corresponded to the utility meter grid wide events were queried against OPQ event database and analyzed. Primarily, if an event was located it was reanalyzed to validate that the metric which triggered the utility meter was the same as the one which triggered the OPQ system.

Frequency Gridwide Events: Out of 35 gridwide frequency events 34 were detected by Napali. As expected, since frequency is quite consistent across the entire power grid, all OPQ devices triggered on the 34 recorded events. The missed frequency event occurred on November 25th at 7:08:00 AM which corresponds to a complete 3 minute power outage as shown in Figure 4.28. Unfortunately, by the time the OPQ devices finished their initialization procedure, the grid conditions have returned to normal. It should be noted that the outage was recorded and reported by the OPQ system.



(a) (b)

Figure 4.28: November 25th outage as observed by the utility meter POST_MAIN_2. a) Observed frequency. b)Observed voltage.

THD gridwide event: A single THD event extracted from the utility meters was recorded and by the OPQ system with every device sending raw waveforms for the event.

V_{rms} **multiphase:** Out of 12 gridwide utility events only 7 were detected by the OPQ system. However, a closer examination of missed events revealed a pattern. The table of the missed events and their triggered meters is shown in Table 4.6

It seems that between 2pm November 26^{th} and 3:30pm November 27^{th} electrical equipment in Hamilton library was experiencing power quality issues, while the rest of the University power grid exhibited nominal behaviour. Since only a single OPQ Device is located in the Hamilton library, Napali rightfully rejected any power quality metrics observed there as local disturbances. Finally, November 25, 7:08 signifies the power interruption as discussed previously. As such the true utility gridwide event number is 8 with 7 detected using the Napali framework.

Table 4.6: Missed V_{rms} multiphase utility events.

Event Timestamp	**Triggered utility meter**
November 26 14:53	HAMILTON_LIB_PH_III_CH_1_MTR
	HAMILTON_LIB_PH_III_CH_3_MTR
	HAMILTON_LIB_PH_III_MCC_AC1_MTR
	HAMILTON_LIB_PH_III_MCC_AC2_MTR
November 27 11:15	HAMILTON_LIB_PH_III_CH_1_MTR
	HAMILTON_LIB_PH_III_MAIN_1_MTR
	HAMILTON_LIB_PH_III_CH_3_MTR
	HAMILTON_LIB_PH_III_MCC_AC1_MTR
	HAMILTON_LIB_PH_III_MCC_AC2_MTR
November 27 11:21	HAMILTON_LIB_PH_III_CH_1_MTR
	HAMILTON_LIB_PH_III_MAIN_1_MTR
	HAMILTON_LIB_PH_III_CH_3_MTR
	HAMILTON_LIB_PH_III_MCC_AC1_MTR
	HAMILTON_LIB_PH_III_MCC_AC2_MTR
November 27 15:18	HAMILTON_LIB_PH_III_CH_1_MTR
	HAMILTON_LIB_PH_III_MAIN_1_MTR
	HAMILTON_LIB_PH_III_CH_3_MTR
	HAMILTON_LIB_PH_III_MCC_AC1_MTR
	HAMILTON_LIB_PH_III_MCC_AC2_MTR
November 25, 7:08	All utility meters.

V_{rms} **single phase:** Out of 32 single phase gridwide events, 23 were detected by the Napali framework. Similarly to the V_{rms} multiphase events:

- 1 event was due to the Nov 25 outage.
- 6 events were incorrectly classified as gridwide, since they originated from the same building.

Two events occurring on November 26^{th} 14:53 and November 27^{th} 11:15 were missed by the OPQ system. These events triggered on all of the Hamilton library meters and the KELLER_HALL_MAIN_MTR located in Keller hall. While one of the phases did go bellow the $\approx 8\%$ threshold, it did not cause a large enough disturbance to appear as an anomaly to the keller hall device. Furthermore, while these events impacted multiple meters, they during the anomalous period experienced by the Hamilton library subgrid. It is unlikely that these events were true gridwide events, but instead the bleed over from the Hamilton library.

Detection capabilities for the Napali framework

The events missed due to the power outage could have been remedied by inclusion of a battery in the OPQ Box. However, even without the battery, OPQ system was able to correctly identify the power outage. Disregarding the multi-utility meter events which originated from the same building,

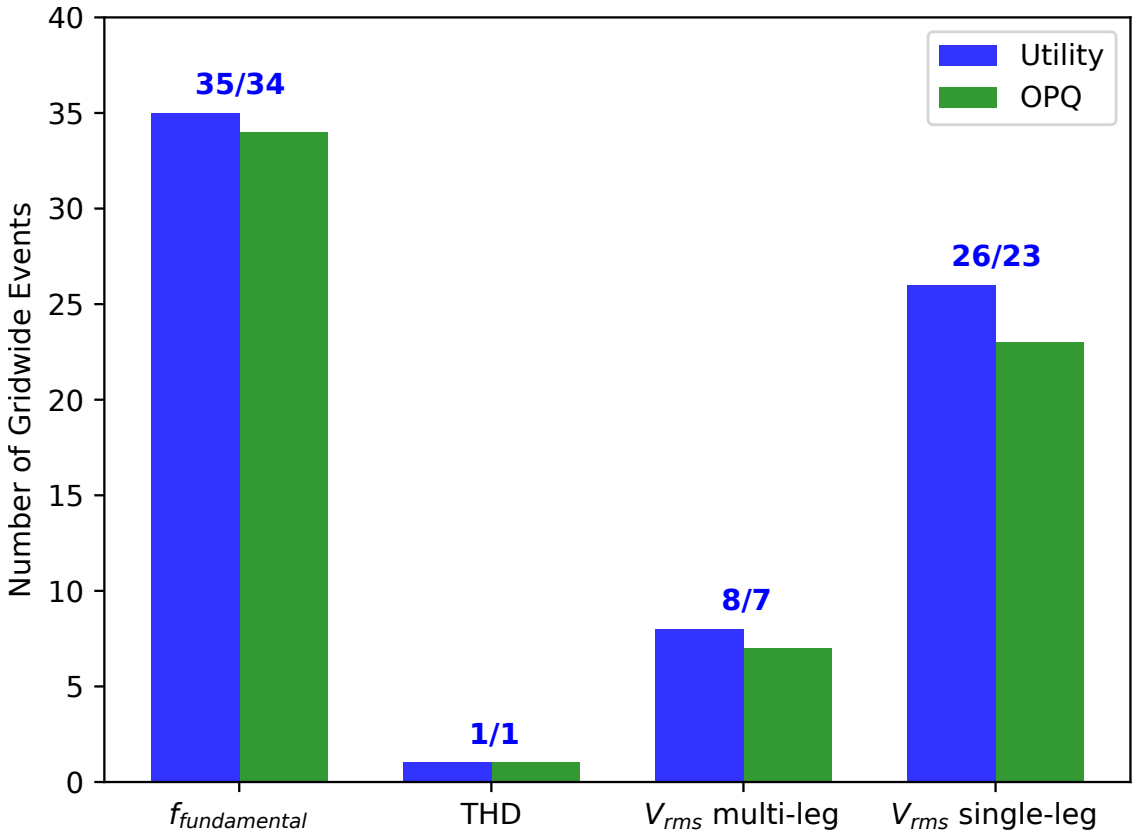


Figure 4.29: Comparison of the gridwide events detected by Napali and the utility meters.

detection performance of the Napali framework is outlined in Figure 4.29. This result validates the temporal locality hypothesis as presented in Section 4.3.5. Disregarding the event caused by the power outage, Napali showed an over 97% detection rate for grid wide events. The two events that were missed by the Napali framework, only affected a single phase of a neighbouring building, and did not cause significant drop in the observed line voltage.

4.3.6 Sub-threshold Data Acquisition

While the Naive method compares favorably to Napali when it comes to resource consumption as outlined in Section 4.3.4, it lacks the ability to detect portions of gridwide events which are below the device detection threshold. Unlike the Naive method, Napli cooperative edge-centric event detection attempts to capture both over-threshold and sub-threshold data as described in Section 3.2.5. Section 4.3.5 described the detection capabilities of the OPQ Network and Napali, when detecting partial gridwide events with consensus of 2 or more utility meters. These are the types of events that the Naive and Napali are equally capable of detecting. Indeed, the Naive plugin in Makai received data from devices which were collocated with utility meters which observed the disturbance. In contrast Napali captured additional waveforms which corresponded to the sub-

threshold portions of each event. More interesting however, were the gridwide events which were extracted from all of the utility meters, not just the ones collocated with an OPQ box. Particularly, gridwide events which corresponded to an over-threshold anomaly observed by a collocated utility meter as well as one or more of the non-collocated meters were of particular interest. These events are invariably gridwide, since more then one UH Utility meter was affected, however only a subset of OPQ devices would experience the disturbance with an over-threshold severity. This is a subtle difference from events used in the previous section. In order to perform this evaluation, a dataset of ground truth validated partial gridwide events was required, and the only way to obtain it was by filtering Utility meter data to temporal windows consisting of:

- **One over-threshold collocated utility meter:** This made sure that one of the metrics from a collocate OPQBox passed a threshold.
- **One or more over-threshold non-collocated utility meters:** This allowed us to conclude that the event was partial gridwide event.

These events create a basis for sub-threshold data acquisition false negative analysis. After all, it is clearly established that these events are gridwide, and affect only a portion of the UH power delivery infrastructure. If Napali is capable of locating and acquiring subthreshold portions of these events, it would prove the claim that the subthreshold event acquisition is both a possible and useful tool for power quality analysis.

Unfortunately, events which impact only a part of the UH power grid with the affected area limited to a region with both collocated and non-collocated meters are quite rare. From the data made available to us from the UH Smart meters only 3 such events were identified. These events are shown in Table 4.7.

It is unclear why these events all seem to have similar parent Utility meters. Nonetheless, two of the events have data from every device on the OPQ network. The other two events only impacted 9 and 10 out of 15 deployed devices respectively. Another peculiarity is that POST_MAIN_2 and POST_MAIN_1 utility meters service the same building, however, POST_MAIN_1 was not affected enough to qualify for these events. In the next sections these events are examined in more detail.

Event 1(Nov 24 7:57):

The V_{rms} for Utility meters registering this event is shown in Figure 4.30. This event was characterized as a large voltage swell, affecting a large portion of the campus. The voltage anomaly was significant enough to trigger OPQ Box devices 1006, 1007 and 1002 using the Self-Triggering method making it a partial gridwide event. In contrast the Napali acquired data from all of the UH

Table 4.7: Gridwide events with collocated and non-collocated meters which impacted only a portion of the power grid

Time	**Collocated**	**Non-collocated**	**OPQBox**
Nov 24 7:57	POST_MAIN_2	BUS_AD_SHIDLER_MAIN_MTR ST_JOHN_PLANT_SCIENCE_MAIN_MTR MOORE_HALL_MAIN_MTR HPER_KLUM_GYM_MTR MULTIPURPOSE_BLDG_MAIN_MTR	All
Nov 29 6:10	POST_MAIN_2	MULTIPURPOSE_BLDG_MAIN_MTR	1000 1002 1005 1006 1007 1010 1021 1022 1023 1024
Dec 14 13:14	POST_MAIN_2	BUS_AD_SHIDLER_MAIN_MTR ST_JOHN_PLANT_SCIENCE_MAIN_MTR MOORE_HALL_MAIN_MTR	1000 1005 1006 1007 1010 1021 1022 1023 1024

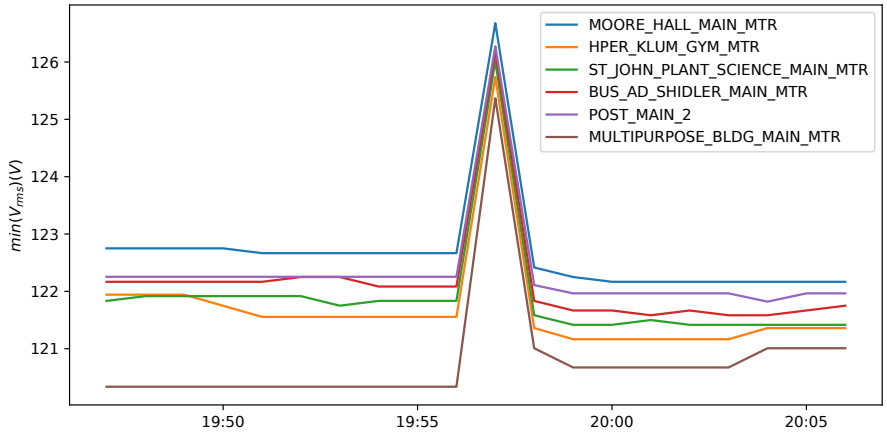


Figure 4.30: Utility meter data for Event 1

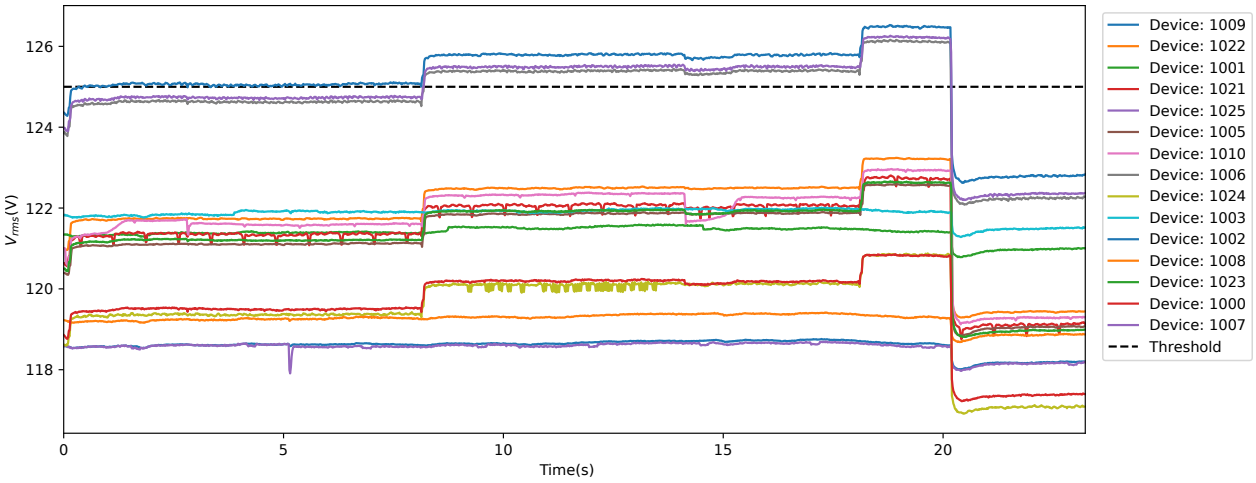


Figure 4.31: Napali Event Data for event 1.

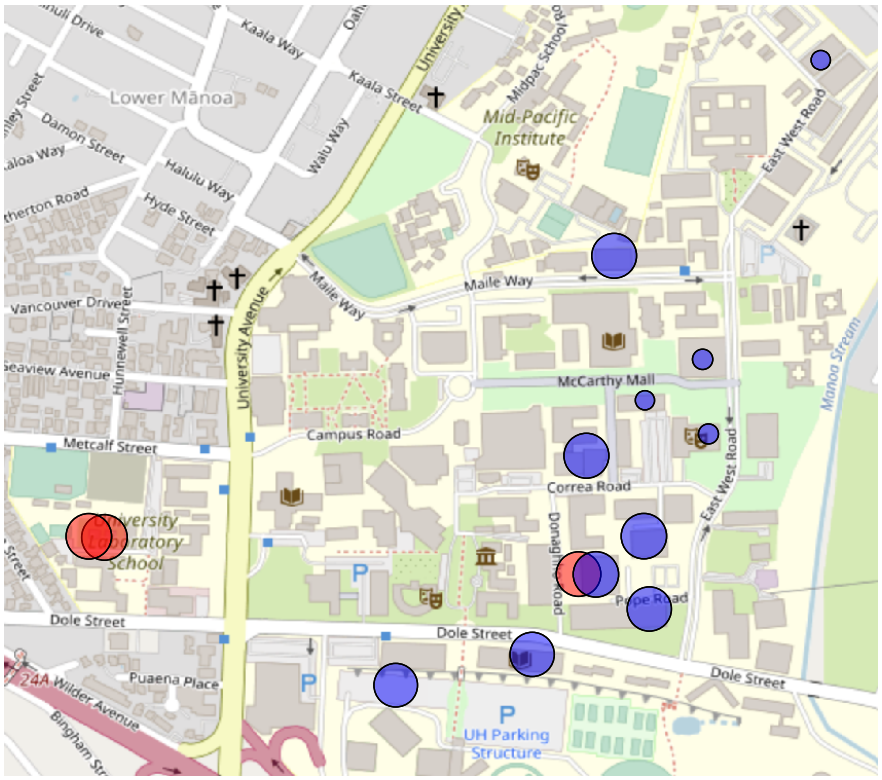


Figure 4.32: Geospatial representation of event 1. Self-Triggered detected events are shown in red.

devices. Figure 4.31 is a 1 cycle V_{rms} calculated from the raw data acquired from the OPQ Devices by Napali. The threshold displayed as a dashed line clearly outlines the 3 anomalous devices using the Self-Triggered method, however a similar trend is observed in all other devices.

Another representation of Event 1 is shown in Figure 4.32. Here the severity of the event is

categorized not by the value over threshold, but instead by the swing between the maximum and minimum of the V_{rms} waveform. Larger changes in the V_{rms} magnitude are used as a metric of the severity of the event at a particular location. The size of the circle in Figure 4.32 represents a larger impact. Whether the device passed the trigger threshold or not is represented in the color of the circle. Devices which were triggered via the Self-Triggered method in addition to Napali are shown in red. Clearly this event looks quite different when acquired via the two methodologies. From the point of view of the Self-Triggered method, the event only had a limited impact on the power grid with only a small portion of campus affected. On the other hand, Napali provides a much clearer picture of the event propagation and impact. In fact the magnitude of the disturbance is just as large in some of the over-threshold and sub-threshold waveforms.

Event 2(Nov 29 6:10):

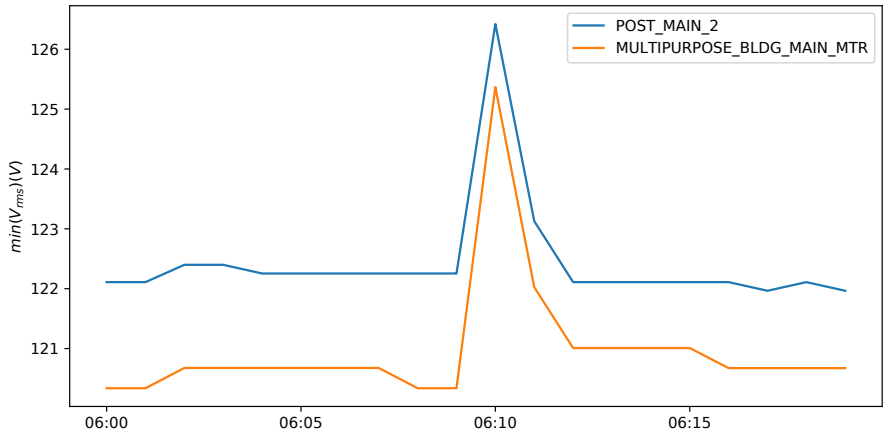


Figure 4.33: Utility meter data for Event 2.

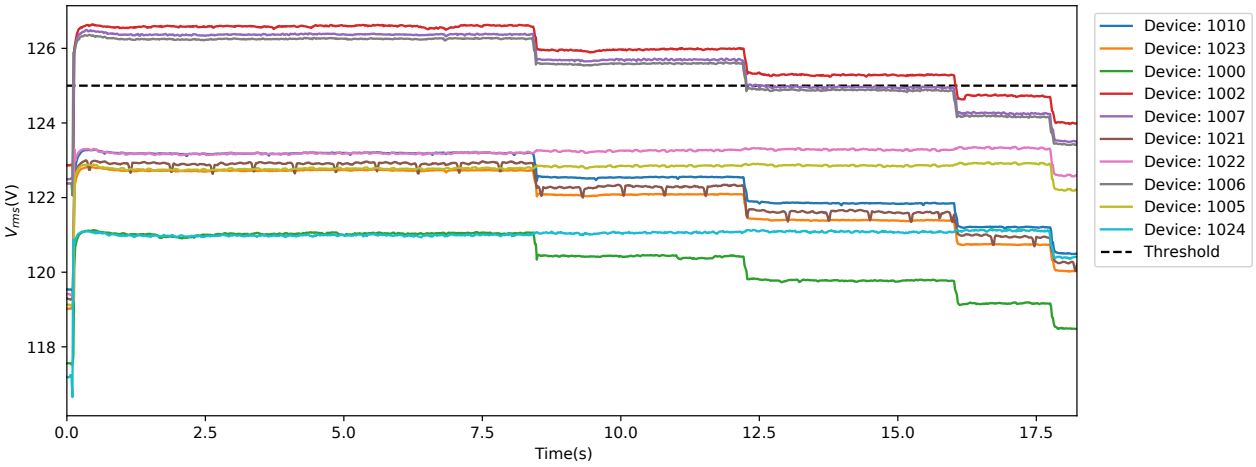


Figure 4.34: Napali Event Data for event 2.

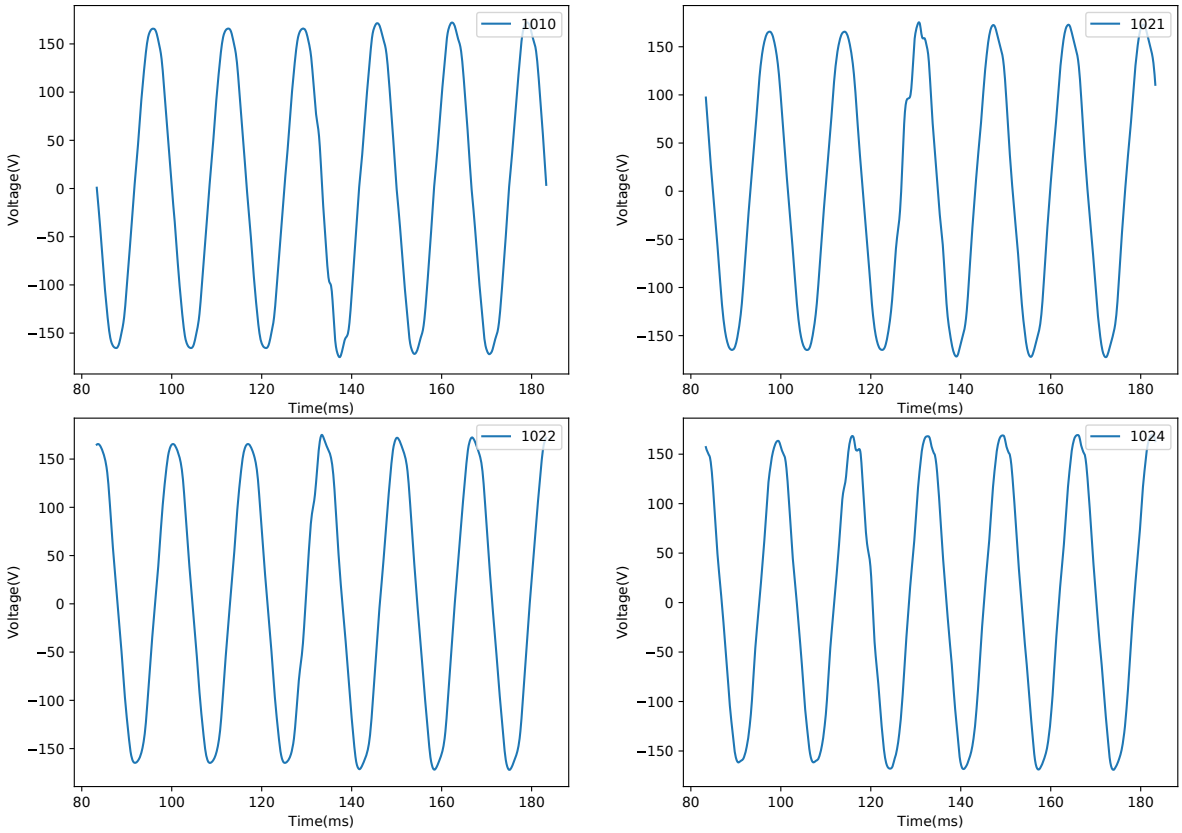


Figure 4.35: Transients recorded as part of event 2.

This event was observed by only two Utility Smart meters. The V_{rms} for Utility meters is shown in Figure 4.33. Similarly to event 1, event 2 was categorized by the Utility meters as a voltage swell. In continued similarity, same devices: 1006, 1007 and 1002 passed the V_{rms} threshold, and thus were triggered by the Self-Triggering event detection method. Napali however, triggered 7 additional devices. The cycle level V_{rms} for the devices detected by Napali is shown in Figure 4.34 While the utility meter data makes Event 1 and Event 2 look quite similar, OPQ data shows them to have an opposing chronological order. Instead of a slow V_{rms} rise and a fast return to nominal as observed in event 1, event 2 is characterized by a fast V_{rms} rise and a slow decay.

Further analysis indicates that the fast V_{rms} rise at $t = 130ms$ is further characterised by a transient observed by 4 OPQ Box devices. These transients are shown in Figure 4.36 The transients were small enough to be missed by the Self-Triggered event detection method, yet were captured by Napali. The geospatial representation of event 2 is shown in Figure 4.36. As with event 1, the magnitude of the event is taken as the swing between the minimum and maximum of the V_{rms} waveform. Event magnitude is represented as the size of the symbol on Figure 4.36. Red represents devices triggered by both Napali and the Self-Triggered method. Blue are events which were triggered only by Napali. Finally, green color symbols represent events which were triggered

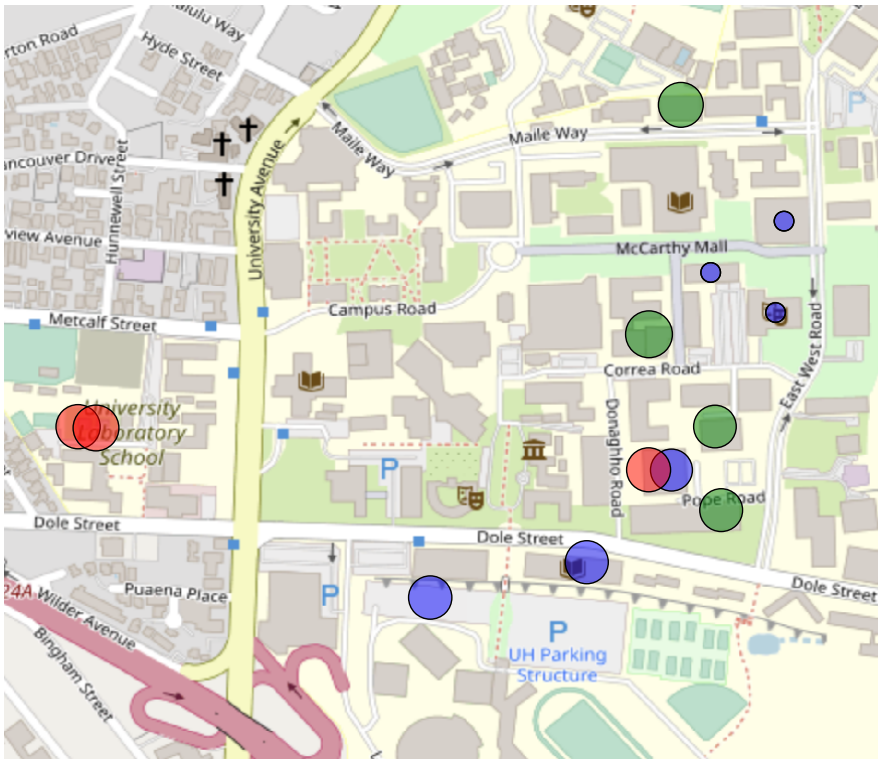


Figure 4.36: Geospatial representation of event 2. Self-Triggered detected events are shown in red. Waveforms containing transients are shown in green.

by Napali with contained transients.

Event 2 as observed by the Self-Triggered method contains only a small portion of the event. Waveforms from the same buildings as those affected by event 1 are captured, with no information available regarding the rest of the grid. The transients are completely missed, and so is the rest of the impacted locations.

This event looks significantly different when observed via Napali. The University power gid is divided into three distinct portions. West portion of campus experienced a V_{rms} swell which surpassed the OPQ imposed threshold. South portion experienced a V_{rms} swell of a similar magnitude, without passing the threshold. Finally, the North-East portion of the campus experienced a similar disturbance as the southern section with an addition of a transient.

Event 3(Dec 14 13:14):

Event 3 is very similar to event 1. It was characterized by a slow V_{rms} swell followed by a sharp return to nominal. Utility meter data for the triggered Utility meters is shown in Figure 4.37. No additional transient was observed as with event 1. Devices 1006, 1007 and 1002 were triggered via the Self-Triggered method, while Napali acquired data from every device on the network. Cycle

level V_{rms} of the recorded data is shown in Figure 4.38. Finally, the geographical representation of this event is shown in Figure 4.39.

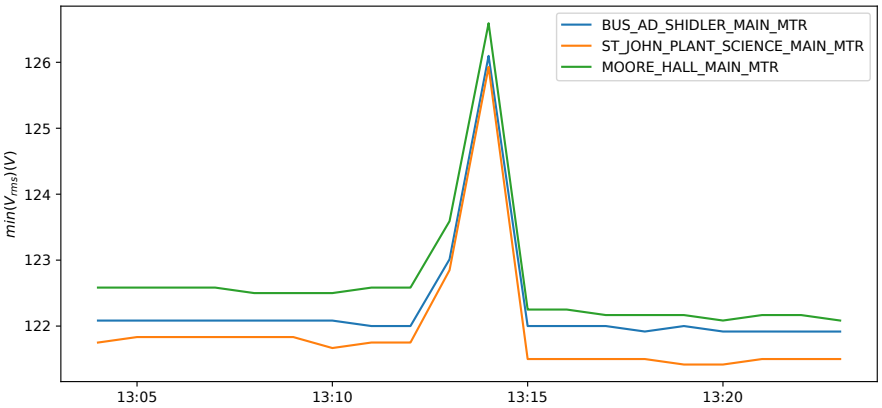


Figure 4.37: Utility meter data for Event 3.

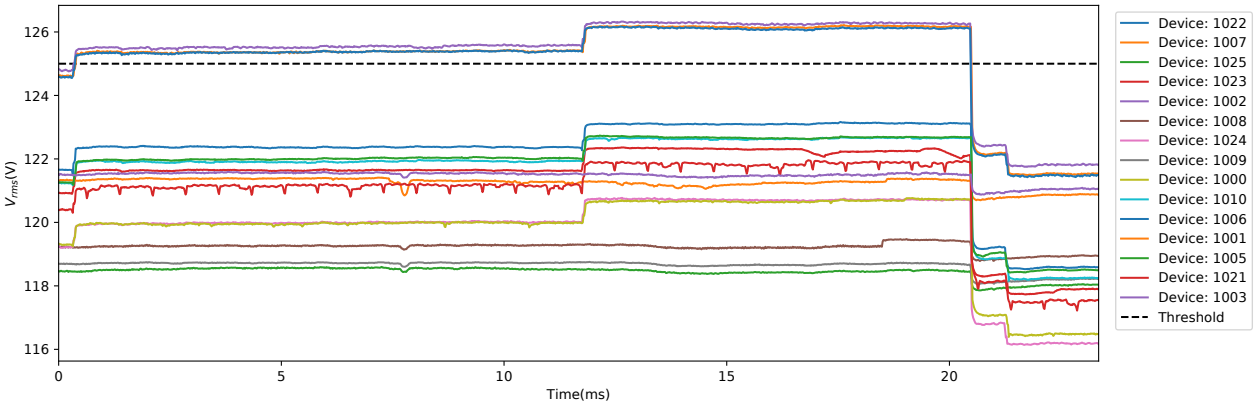


Figure 4.38: Napali Event Data for event 3.

False Negative Sub-Threshold Data Acquisition.

As mentioned previously, the goal of Napali is not a low false positive detection rate. Napali only serves as a gateway for data acquisition in a complex system designed for power quality analysis. Any false positive events acquired by Napali will be cross examined by algorithms far better suited for waveform analysis. Instead, the goal of Napali is to provide an extremely low false negative event detection rate, providing the downstream detection stack all of the available data from all potentially affected device. As shown in the evaluation with the ground truth data Napali has performed significantly better then the Self-Triggered method in event acquisition. False negative sub-threshold data acquisition rate on Napali is approximately 0% based on the following facts:

1. **Multiple triggered collocated meters:** As shown in Section 4.3.5 Napali was able to

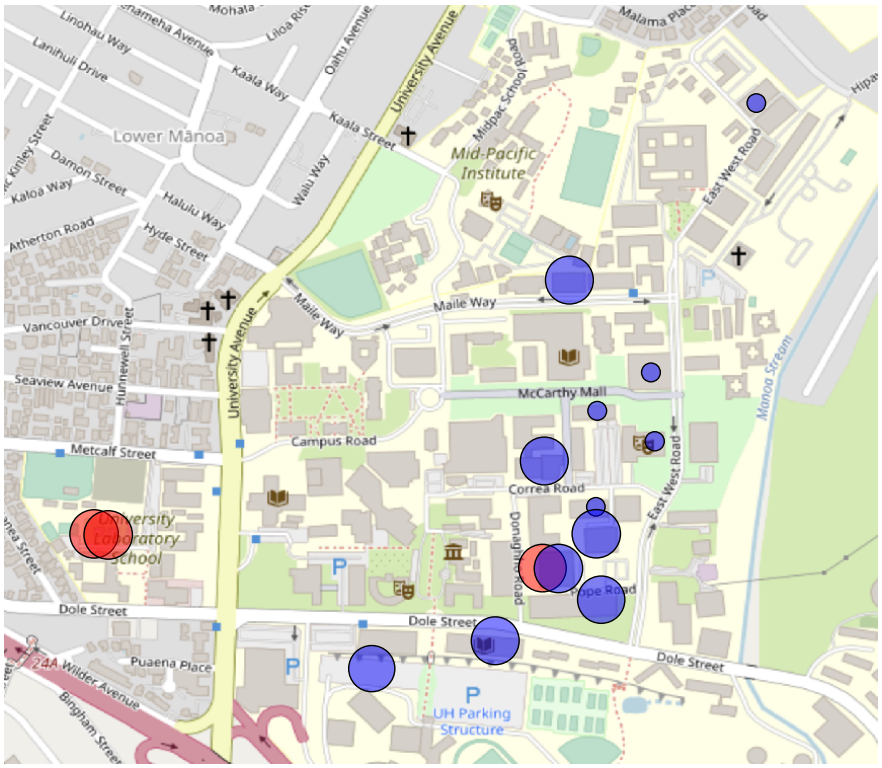


Figure 4.39: Geospatial representation of event 3. Self-Triggered detected events are shown in red. Waveforms containing transients are shown in green.

identify events which were marked as gridwide by the utility meters at the rate of 100% accuracy.

2. **Single triggered collocated meters:** Napali was able to capture both threshold and subthreshold data for events which were identified as gridwide by a subset of collocated and non-collocated Utility meters at a rate of 100% accuracy.

These results validate the claim described in Section 1.6.5. Sub-threshold triggering data compares favorably to the ground truth data acquired from the UH power meters. Furthermore, the next Section describes an additional benefit of the subthreshold triggering: power grid partitioning.

4.4 University Power Grid Partitioning via Power Quality Data

The previous section provides evidence in support of the claims for this dissertation, but the UH Microgrid study yielded an entirely new application of subthreshold triggering: power grid substructure detection. This section discusses this "emergent result" of the research project. In addition to the 3 events described in the previous section, Napali detected another 25 events triggered on devices 1006, 1007 and 1002, all with a large sub-threshold data pool. All 28 of

these events share the same structure with a voltage swell, sometimes accompanied with additional transients in the rest of the power grid. Besides devices 1006, 1007 and 1002, further clusters of of events were identified. There were 28 events with devices 1024 and 1000 observing a temporally correlated voltage sag. One of such events in temporal and spacial representation is shown in Figure 4.40.

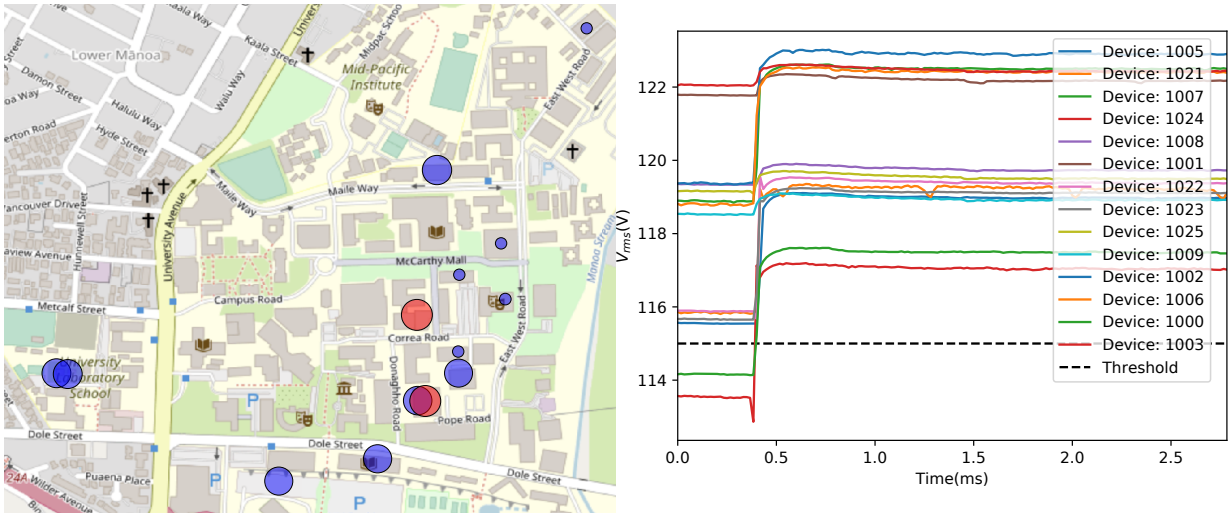


Figure 4.40: A voltage sag observed on Dec 4, 6:07. *Right:* Temporal representation *Left:* Spacial representation.

There were 15 events with devices 1003 and 1001 observing a common voltage sag beyond the threshold. A typical event involving these devices is shown in Figure 4.41.

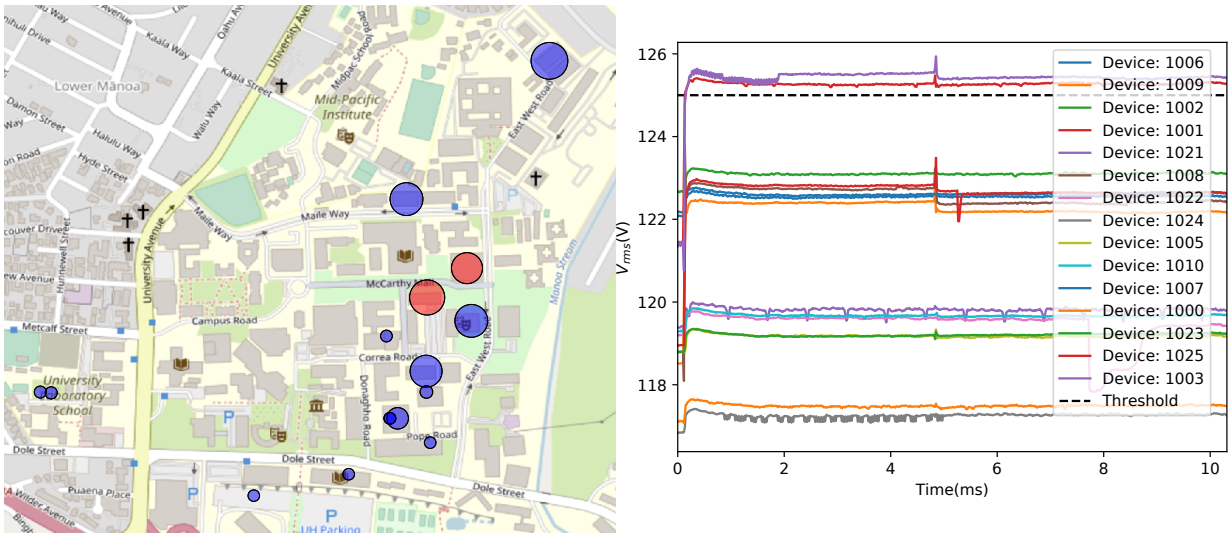


Figure 4.41: A voltage sag observed on Nov 21, 6:24. *Right:* Temporal representation *Left:* Spacial representation.

.

While the origin of these events is unclear, Napali data allows for creation of a propagation model for events originating from the triggered subset of the clustered devices to the rest of the network. For this model the $max(V_{rms}) - min(V_{rms})$ is used as the anomaly metric. We can compute a correlation metric for any devices d and p over n events as shown in Equation 4.5

$$W^n_{p\to d} = \frac{(max(V_{rms}) - min(V_{rms}))_p}{((max(V_{rms}) - min(V_{rms}))_d}$$
$$W_{p\to d} = \frac{1}{n}\sum_n (W^n_{p\to d}) \tag{4.5}$$

The result of Equation 4.5 for the cluster of events triggered by devices 1000 and 1024 is shown in Figure 4.42.

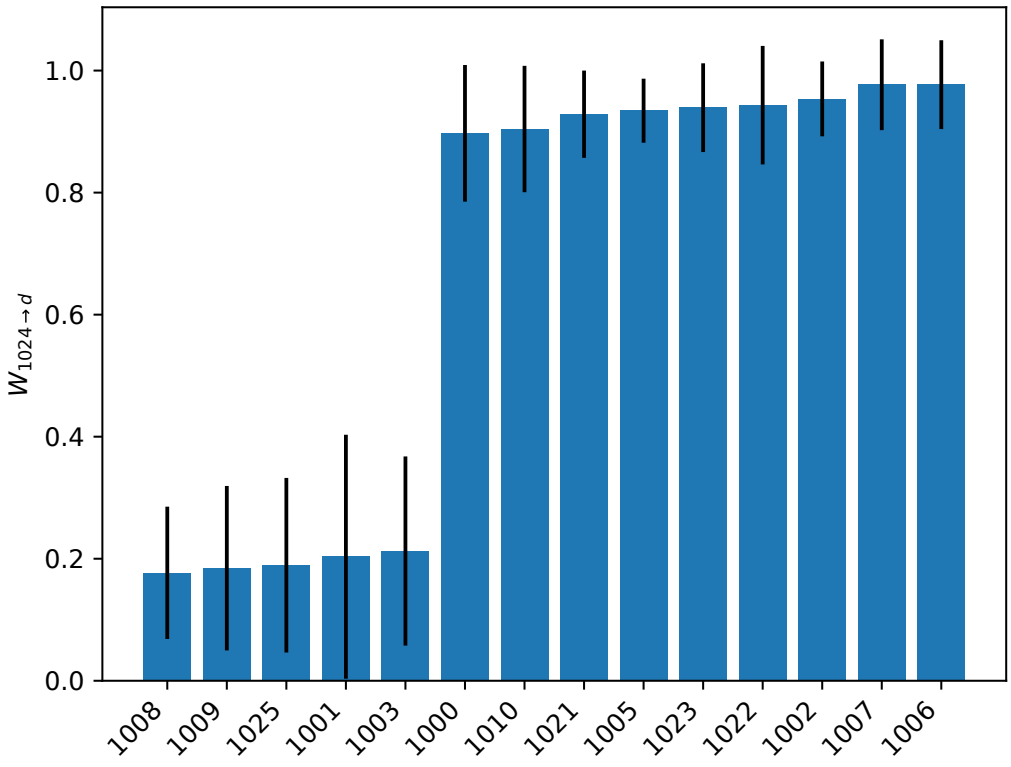


Figure 4.42: Blue bars represent the similarity coefficient $W_{p\to 1024}$. Black bars represent the σ for the ensemble of $W^n_{p\to d}$.

The error bars in Figure 4.42 represent the σ of the average between device 1024 and device in question across the 28 events in the cluster dataset. It is clear that the disturbances observed by OPQ Boxes 1010, 1005, 1021, 1022, 1023, 1002, 1007 and 1006 show are strongly interconnected with those observed by devices 1024 and 1000. On the other hand devices, 1008, 1009, 1025, 1001 and 1003 experienced a much more diminished disturbance.

A similar examination was performed on a cluster of events triggered by a V_{rms} disturbance observed by devices 1002, 1006 and 1007. This time device 1006 was selected as the device p.(Device p is selected arbitrarily) Result of Equation 4.5 with error bars representing σ of the fractions between device 1006 and device in question is shown in Figure 4.43.

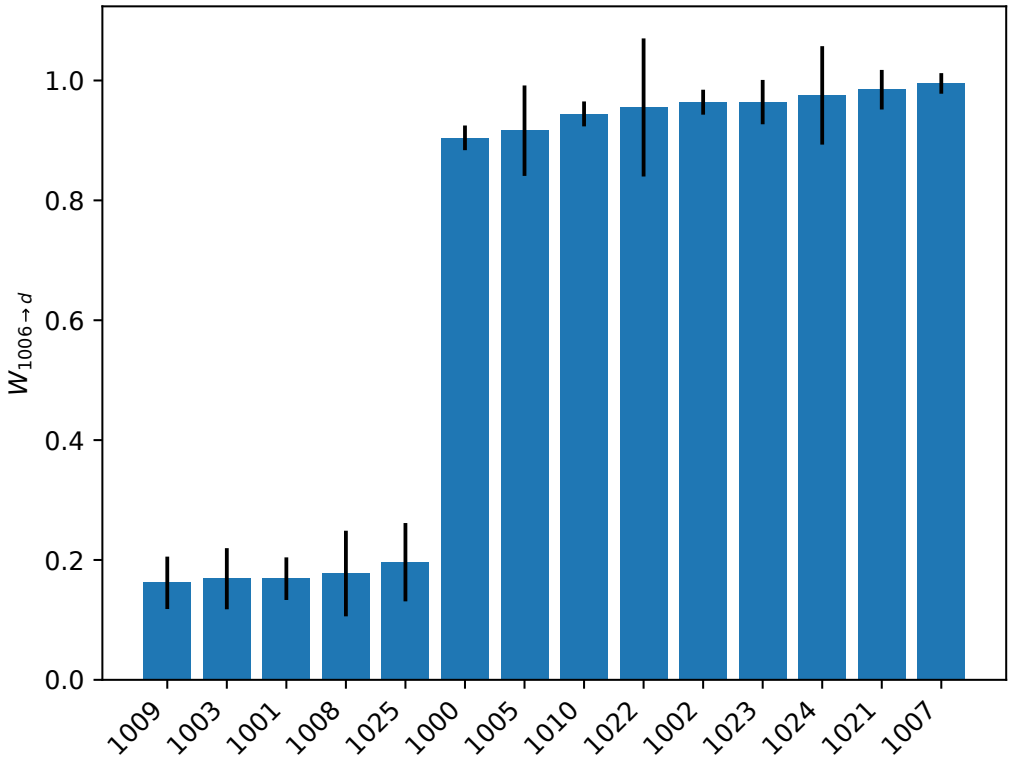


Figure 4.43: $W_{p\to d}$ metric computed for device p=1006

Figure 4.43 and 4.42 both show the same trend. Devices 1010, 1005, 1021, 1022, 1023, 1002, 1007, 1006, 1024 and 1000 observe similar magnitude V_{rms} swell while devices 1008, 1009, 1025, 1001 and 1003 are minimally affected.

Finally, we examined cluster of events triggered by devices 1003 and 1001 using Equation 4.5 with OPQ Box 1003 serving as device p. As previously, the error bars represent standard deviation of the fractions between device 1006 and device in question. The result of this analysis is shown in Figure 4.44. These $W_{p\to d}$ results shown here are the opposite of the results for the other two clusters. Devices 1008, 1009, 1025, 1001 and 1003 experience similar magnitude of the disturbance while devices 1010, 1005, 1021, 1022, 1023, 1002, 1007, 1006, 1024 and 1000 are far less impacted.

From these observations, we can conclude that at the top level, Universiy of Hawaii power grid is divided into two subgrids The East subgrid is monitored by OPQ Boxes 1010, 1005, 1021, 1022, 1023, 1002, 1007, 1006, 1024 and 1000. The West subgrid is monitored by OPQ Boxes 1008, 1009, 1025, 1001 and 1003. Each of the subgrids is likely further divided into sections. We can get a glimpse of this substructure in events such as the one shown in Figure 4.45.

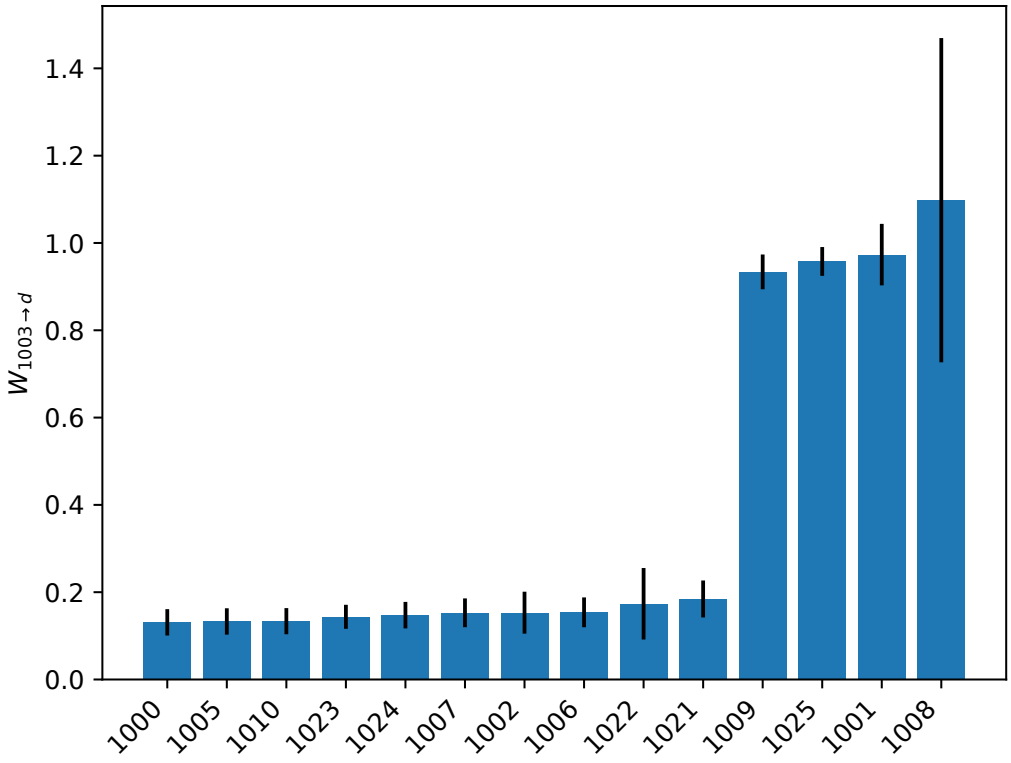


Figure 4.44: $W_{p\to d}$ metric computed for device p=1003

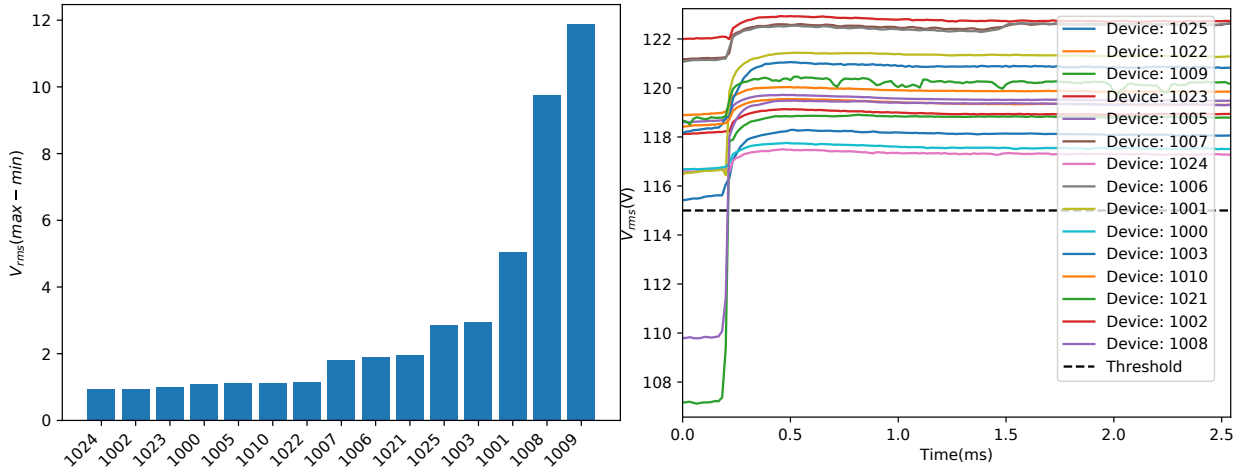


Figure 4.45: A voltage sag observed on Dec 16 14:43. *Right:* Temporal representation. *Left:* Difference between the minimum and maximum V_{rms}.

Figure 4.45 shows an event that was captured by Napali on Dec 16. Recording of this event was initiated by devices 1008 and 1009 V_{rms} measurement passing the voltage sag threshold, with additional sub-threshold data captured from the rest of the network. Figure 4.45 shows the further division of the West subgrid in what is potentially subgrids containing devices 1008 and 1009 as

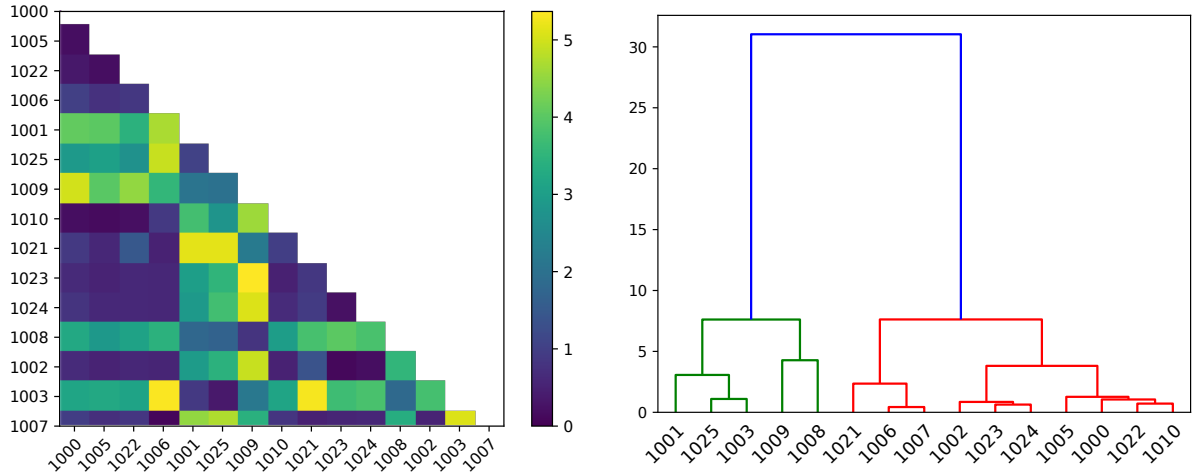
well as devices 1003, 1025 and 1001.

4.4.1 Clustering Napali events

In order to examine the UH power grid structure more closely, a further analysis was performed on all of the Napali collected data via hierarchical clustering. Out of 2163 events collected, 1378 events contained a voltage sag or swell in 2 or more devices at a magnitude $max(V_{rms}) - min(V_{rms}) > 1V$, as such these events formed the dataset for this analysis. Since the number of nodes we wish to cluster is quite small, agglomerative hierarchical clustering was used. Next, a pairwise dissimilarity metric had to be selected. Equation 4.5 could not be used as a pairwise dissimilarity, since $W_{p\to d}$ is at best a similarity coefficient. Instead we define a pairwise dissimilarity between two devices as the average of the L1 distance between the $max(V_{rms}) - max(V_{rms}^{min})$ for devices p and d as shown in Equation 4.6.

$$D_{p\to d}^{rms} = \frac{1}{n}\sum_{n} |(max(V_{rms}) - min(V_{rms}))_p - ((max(V_{rms}) - min(V_{rms}))_d| \tag{4.6}$$

The pairwise dissimilarity matrix calculated from the 1378 events in the dataset is shown in Figure 4.46a. Using this pairwise dissimilarity matrix the hierarchy problem was reduced to unidimensional hierarchical clustering. Clustering was performed using *sklearn* Python library, and the result is visualized in Figure 4.46b. Figure 4.46b shows a dendrogram representation of the extracted hierarchy. The height of each level represents the confidence level extracted by the clustering algorithm.



(a) (b)

Figure 4.46: *Left:* $D_{p\to d}^{rms}$ calculated for the Napali dataset containing V_{rms} anomalies. *Right:* Hierarchical clustering dendrogram.

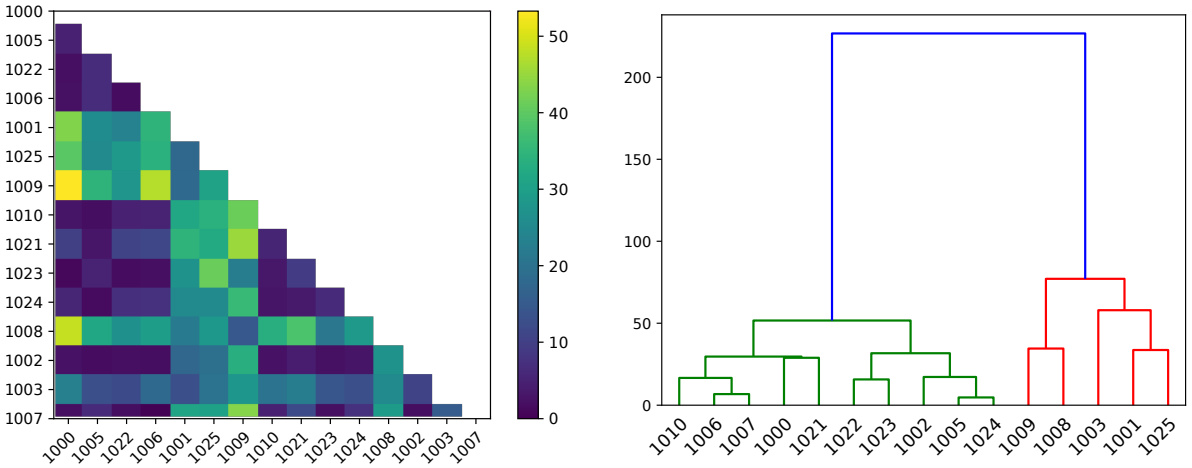
This result further validates our binomial top level hierarchy hypothesis. Top level clusters

extracted from 1378 events were divided in agreement with our previous findings. Furthermore, partitioning of the two clusters further matched our observation of events such as the one portrayed in Figure 4.45.

Next, the same partitioning strategy was applied to the transient data. Transient events, were filtered from all of the Napali collected waveforms by selecting events with more then one device observing a transient $> 5V$ in magnitude. This reduced the working dataset to 737 events. Each waveform was filtered via a 400Hz highpass filter, a region with the transient was identified and the maximum and minimum of the transient was extracted. The pairwise dissimilarity metric was selected as the absolute value of the difference in the transient magnitude as shown in Equation 4.7.

$$D_{p\to d}^{trans} = \frac{1}{n}\sum_{n} |(max(V) - min(V))_p - ((max(V) - min(V))_d| \tag{4.7}$$

The pairwise dissimilarity matrix is shown in Figure 4.47a. Unlike the pairwise dissimilarity matrix in Figure 4.46a device 1003 hierarchy is not as obvious. The result of the hierarchy algorithm for this dataset is shown in Figure 4.47b. As expected device 1003 is placed higher in the hierarchy, otherwise the top two levels of the grid structure remain unchanged from the V_{rms} result for the smaller top level cluster. The structure of the larger cluster remains undetermined with poor clustering performance with both V_{rms} and transient data sets.



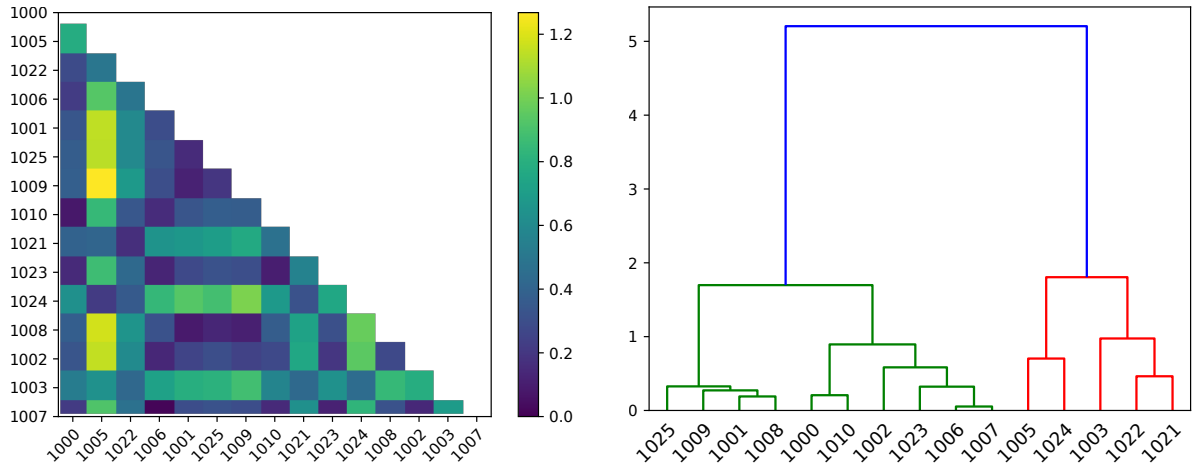
(a) (b)

Figure 4.47: *Left:* $D_{p\to d}^{trans}$ calculated for the Napali dataset containing transient anomalies. *Right:* Hierarchical clustering dendrogram.

Finally, clustering was performed by examining the THD across all of the Napali events. Only 2 events in the entire dataset triggered exclusively due to the THD metric surpassing the threshold. As such, in order to obtain enough data to calculate the pairwise dissimilarity matrix, THD was computed for every event collected by Napali. The pairwise dissimilarity function used for clustering

was the absolute value of the difference between the THD values between devices p and d as shown in Equation 4.8.

$$D^{thd}_{p \to d} = \frac{1}{n} \sum_{n} |THD_p - THD_d| \tag{4.8}$$



(a) (b)

Figure 4.48: *Left:* $D^{thd}_{p \to d}$ calculated for the entire Napali dataset. *Right:* Hierarchical clustering dendrogram.

The computed pairwise dissimilarity matrix is shown in Figure 4.48a. The pairwise dissimilarity matrix looks quite different from the ones derived with V_{rms} and transient events. Particularly, devices 1003, 1005, 1021, 1023 and 1024 exhibit, THD reading is significantly higher then the rest. Clustering algorithm result is shown in Figure 4.48b. As expected devices 1003, 1005, 1021, 1022 and 1023, are placed in a separate cluster resulting in a tertiary top level division. Otherwise the other two clusters remain unchanged from the previous results.

Event Clustering Evaluation

There are two substations which feed the UH power grid: substation M and substation L. Each substation has multiple circuits labeled alphabetical. Finally each circuit has a subcircuit labeled with a number. For example a building connection MA3 which powers the POST building is originating from substation M circuit A line 3. Using Figure 4.7, each OPQ device was paired a power line supplying it's location. This information along with the top level cluster ID generated from V_{rms} and transient clustering is shown in Table 4.8

Interestingly, the top level clusters extracted from both V_{rms} and transient metrics follows closely to the top level hierarchy of the power grid itself. This has several implications. First of all, Napali

Table 4.8: Gridwide events with collocated and non-collocated meters which impacted only a portion of the power grid

Device	Powerline	Cluster
1001	lb2	1
1008	lb2	1
1003	la2	1
1009	la3	1
1025	la3	1
1000	ma3	2
1002	ma3	2
1005	mb5	2
1006	ma4	2
1007	ma4	2
1010	ma3	2
1021	ma3	2
1022	ma1	2
1023	ma5	2
1024	mb3	2

statistical trigger could utilize this clustering in order to improve scalability. Once the clusters have been established, only neighboring clusters need to be explored for subthreshold events. Most importantly, the clusters themselves are can be established with no prior knowledge of the power grid topology. Secondly events can be localized to the individual clusters via subthreshold triggering, thus allowing for event epicenter location. Both of these features can be used to improve Napali.

THD clustering produced a result which is dramatically different from the other two metrics. While it is unclear why this is is the case, possible explanations are: local noise and building level filters. THD at any given location is a sum of THD originating from the the grid and THD originating from the local sources. It's possible that some buildings have much better THD mitigation techniques in place the others. Alternatively, it's possible that some locations house sources of significant THD noise. For example the highest average THD across any building originated from device 1005 located at the UH parking structure. A newly deployed 1MW solar farm located on the roof of this parking structure may be the cause of the additional THD noise. Device 1003 located in keller hall was housed in the Lava Lab, a location with a substantial amount of electrical equipment. IT Building, the location of the device 1024, is the hub of the UH IT infrastructure, and is likely a candidate for higher then normal local THD noise.

4.4.2 Subthreshold Triggering Advantage

The level of analysis shown in the previous section would be impossible with a self-triggered dataset. Without subthreshold data the top level clustering would consist of 4 individual clusters

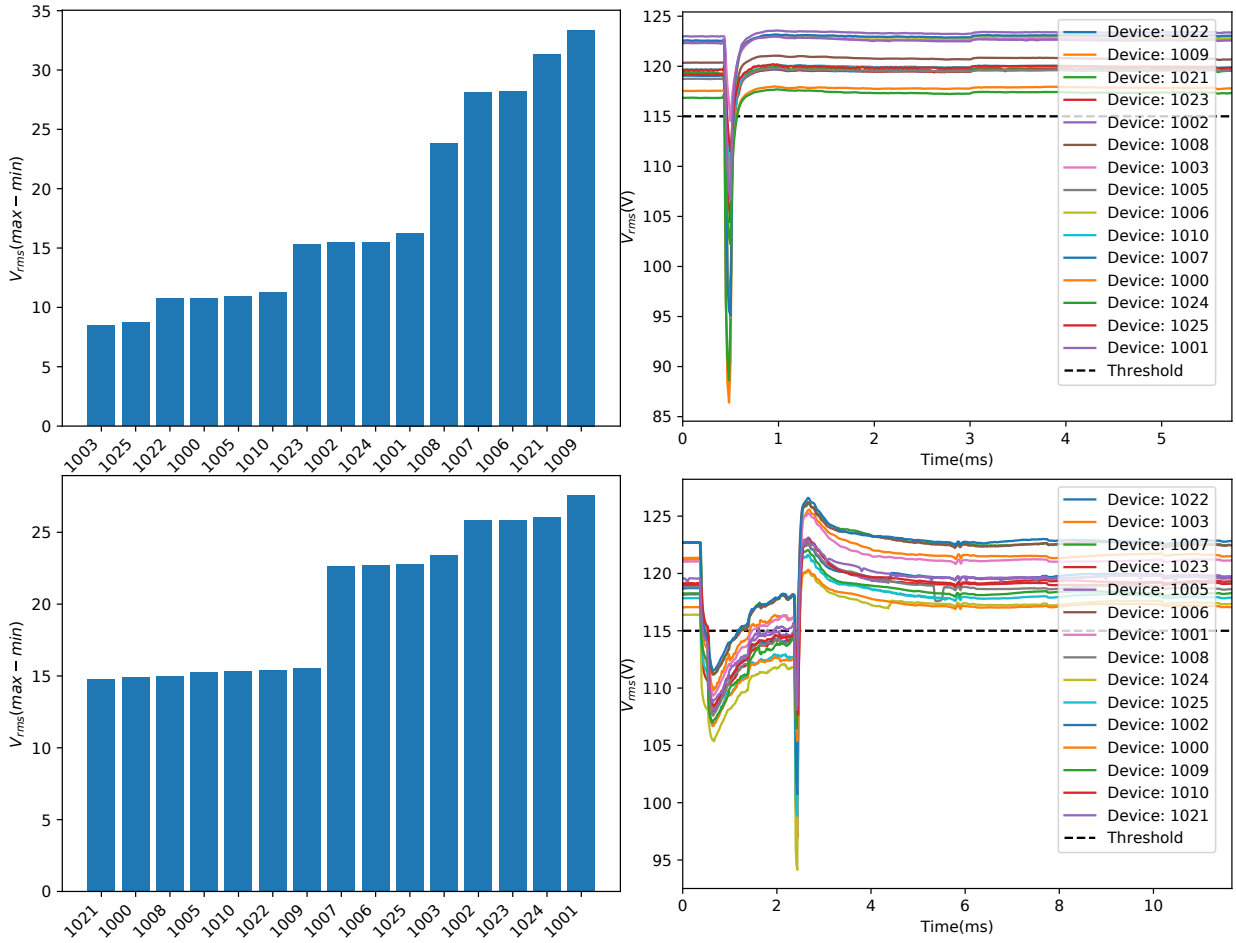


Figure 4.49: Two of the largest magnitude events recorded by the OPQ network. *Right:* Temporal representation. *Left:* Difference between the minimum and maximum V_{rms}.

instead of just 2. It should be noted that during this analysis we found it impossible to reconstruct lower sub-grid hierarchy from events that affected the entire higher level. For example, consider the two most severe events that OPQ observed during the UH deployment as shown in Figure 4.49. With all of the devices passing the voltage sag V_{rms} threshold, previous sub-grid structure is no longer visible. A simple explanation for loss of this hierarchy is propagation of the power quality events from the top of the hierarchy down operates differently from the events discussed previously. If an event occurs in a subgrid it must propagate up the power delivery infrastructure to affect a neighbouring subgrid. On the other hand, events that affect the entire network, it never needs perform lateral jumps in the hierarchy. As distributed power generation overtakes the large centralized power plants, this type of grid partitioning and monitoring will become increasingly more important. Instead of monitoring a single power producing entity such as a coal or natural gas plant, many distributed renewable and non-renewable power sources will need to be monitored for their power quality contribution. Power quality disturbances have the greatest effect downstream

of the power delivery hierarchy. Napali is well suited for monitoring power production at the leaf nodes of the electrical distribution system, and separate local noise from partial gridwide events. Self-Triggering systems are incapable of performing this task, while Naive methods do not scale well enough to support potentially thousands of monitoring sites. Napali on the other hand, has proven to operate efficiently, resiliently and accurately in the power quality domain.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

5.1 Application of Napali in Other domains

Sensor networks are prolific in today's world. Industrial process and environment monitoring is striving to make the world more efficient and productive. Medical and Personal sensing is a welcome addition to improving health care and quality of life. Many of these fields operate in regimes suitable for Napali. The results of my research provide evidence that Napali is well suited for sensor networks operating in domains with:

- Signal to noise ratio of > 1.
- Consensus based event detection.
- Two way communication between the device and sink.

Any monitoring situation which requires a consensus of multiple devices, with individual devices unable to ascertain the validity of an anomaly is well suited for Napali deployment. In this section I review potential applications of Napali to several domains intrinsically different from power quality monitoring, while still adhering to the constraints outlined above.

5.1.1 Earthquake detection

Detection of seismic phenomenon is a task well suited for sensor networks. Single location monitoring is unfeasible, due to a multitude of factors. Local noise from human activity results in a large number of false positives. Furthermore, single location monitoring is useless for development of an early warning system. Earthquake detection relies on prompt detection of P-waves, or pressure waves. These waves travel faster than their more destructive counterpart: S-waves. A prompt detection and characterization of a P-wave can provide an early warning of an impending catastrophe. A large number of sensor networks of varying complexity and sophistication have been deployed in order to monitor geographic areas for seismic activity.[8][42][20][12]

In their paper "Lessons Learned from Operating an On-site Earthquake Early Warning System"[42] authors Zaicenco and Weir-Jones describe the main challenges for designing and operating an earthquake detection system:

1. Unknown direction of a potential seismic event, since sources capable of generating the ground motion that exceeds design parameters are spread around the region.
2. Multiple sources of industrial noise at the site: highway traffic, railroad, fishery, heavy trucks driving several meters away from the instrumented area;

3. Requirement for the system to operate 24/7 in the autonomous mode for several years;

4. High cost of a potential false alarm, which might result in closing the traffic on the major highway.

Their experience comes from operation of borehole sensor arrays located along the highways of British Columbia. Each sensor array is connected via a fiber drop to the central computer. Every measurement performed by the sensor array is transmitted to a main processing unit for P-wave analysis. The triggering algorithm first precomputes a p-wave metric for each sensor, and uses a threshold based algorithm for earthquake detection. This is a well established system which was able to detect and provide early warning for multiple earthquakes during its operational phase 2009-2011. The design of this system is very similar to a Naive triggering method for Power Quality monitoring. All data is funneled to the central sink and processed on site.

The California Integrated Seismic Network is another seismograph sensor network consisting of over 400 high quality borehole sensors.[37] Similarly to the On-site Earthquake Early Warning System, the California Integrated Seismic Network transmits all of the sensor data to the central sink. The California Integrated Seismic Network is now a part of ShakeAlert, a US-based effort to integrate seismic prediction into actionable intelligence.

Another approach to earthquake monitoring comes from the newly emerged IOT domain. The cellphone based Earthquake Network [12] utilizes smartphone accelerometers in order to detect P-wave propagation throughout the world. As a part of the Earthquake Network, cellphones which are at rest and plugged into a power source will monitor the internal accelerometer for abnormalities. If the inertial tensor recorded by the cellphone passes a threshold a message will be transmitted to a central cloudbased sink. The cloudbased sink in turn uses device location and statistical clustering in order to determine if a P-wave has been detected. Another IOT sensor network designed for earthquake detection is called Earthcloud.[20] This sensor network utilizes dedicated low cost sensors which communicate via the Internet to the centralized sink. Similarly to the Earthquake Network, Earthcloud utilizes the number of "prewarnings"(devices which passed the local threhold) in order to determine if an earthquake is taking place.

Sensor networks described above fall into the two categories described in the previous chapters. The On-site Earthquake Early Warning System and California Integrated Seismic Network are the Naive approaches with all of the sensor data funneled to the sink. Earthcloud and Earthquake Network on the other hand are similar to the self-triggered system, with additional statistical analysis performed at the sink. All three networks maintain two way TCP/IP link between devices and the sink. With a relatively small number of nodes the On-site Earthquake Early Warning System and California Integrated Seismic Network are able to operate in the Naive mode due to relatively low bandwidth requirements. On the other hand, Earthquake Network app has 4 million downloads and 75000 active cellphones, and as such is forced to operate in the self triggered mode.

Napali methodology could enhance both of these event detection topologies. "High quality"

earthquake monitoring networks such as California Integrated Seismic Network serve a dual purpose. First and foremost, they are the nation's early warning system for disaster mitigation. Second, they are a research and analysis tool used by geophysicist to study earthquake propagation, localization and classification. Application of Napali to earthquake detection could:

1. Reduce the bandwidth requirement for operating a seismic sensor.
2. Preserve sub-threshold earthquake data for earthquake analysis.
3. Reject single point anomalies resulting from human activity.

Napali approach could aggregate multiple measurements in order to conserve bandwidth for below threshold events. If a local threshold is passed, the measurement would be forwarded to the sink immediately in order to provide a timely latency. If the sink determines that a P-wave has been detected, raw, high resolution data can be requested from the seismic sensor.

Currently IOT approaches are useless for scientific application, since only a "prewarning" is transmitted to the sink without the high resolution waveform. With Napali useful high resolution data can be transmitted for later scientific analysis.

5.1.2 Lightning Detection

United States National Lightning Detection Network operates over 100 sensors across the United states[9]. These lightning detectors create a sensor network used to record, detect, localize and classify lightning strikes. Data from this network is made available for utilities, National Weather Service and to meteorologists for analysis and study. Potential uses for this data include cooperation with the Forestry Service, space flight providers, air traffic control, and wind farm operators in order to mitigate lighting risk.[27] Lightning detectors are placed in remote areas and communicate via satellite in order to limit interference from anthropogenic sources. The operational cost of such network can be reduced by placing the lighting detectors in urban centers and other locations they are supposed to protect. The local man-made noise presents an issue for traditional Self-Triggered operation of such a network, however using Napali, this noise can be filtered out. Napali has the potential to provide lighting detection at a fraction of the cost of a satellite based system. Furthermore, sub-threshold data from Napali would allow for better event localization and classification than a conventional Self-Triggered system.

5.1.3 Gunshot detection acoustic sensor networks

Gunshot detection acoustic sensor networks are generally placed in urban areas. A multitude of these sensor networks have been proposed in the literature, however the only known operational systems seem to be single location multi-microphone devices. [6] [19] There are however patents relating to multi-sensor gunshot detection systems, so it is possible that such systems are operated

covertly. All of these systems rely on feature extraction and triggering performed on the device itself, making them Self-Triggered networks. Napali could enhance the detection capability of these networks through sub-threshold detection as well as on the fly localized false positive rejection. Once the waveforms from the sensors are acquired, more sophisticated algorithms can be utilized for event localization.

5.1.4 Neutrino Physics

Electron anti-neutrino($\bar{\nu}_e$) is a particularly interesting flavor of a neutrino which interacts with a very specific signature. An inverse beta decay process is characterized by an $\bar{\nu}_e$ interaction with a proton, resulting a creation of an electron and a positron:

$$\bar{\nu}_e + p \rightarrow e^+ + n \tag{5.1}$$

The positron is immediately captured resulting in production of two gammas:

$$e^+ + e^- \rightarrow \gamma + \gamma \tag{5.2}$$

This is known as the prompt event and it is well characterized in both energy and time. The neutron will travel through its medium until capture by a nucleus of an atom, resulting in additional gamma release. This is known as the delayed event, and it will occur within 200us of the initial interaction. The advantage of the inverse beta decay is that it allows for a well characterised and established detection mechanism for neutrino measurement. In fact it's the only mechanism we are aware of which uniquely identifies the neutrino.

Many neutrino experiments utilize inverse beta decay as a main physical mechanism.[4][23] Unfortunately the chance of the neutrino interaction is extremely small, with many experiments expecting only a few events per day. In order to maximize data-rate a larger observed volume and a larger number of optical detectors are required. As such waveform samplers are utilized instead of common ADCs for event capture.

Waveform samplers trade dead time for low power per channel. Intrinsically, waveform samplers are not ADC in and of itself, instead they are an analog storage medium. Modern waveform samplers can store $1ms$ or more of analog waveform sampled at $> 10Gsps$. Along with a slow ADC and a triggering system, a waveform sampler allows for extremely fast waveform extraction at the cost of some experiment dead time. The waveform sampler is filled in a round robin fashion, until the triggering systems determines that an event is occurring. Once the anomalous condition is identified, the sampler stops, and a slow ADC digitizes all or a portion of the waveform buffer, The digitized signal is then passed further up the triggering chain. A Waveform digitizer triggering system is generally a single discriminator per channel built into the the waveform sampler ASIC. At the lowest layer, the triggering systems count the number of channel discriminator hits and

converts them into a triggering metric. In some cases the geometry of trigger bits is also used in trigger determination. This layer of the triggering system is commonly referred to as L0.

Tuning the trigger discriminator threshold is a tedious task. The detectors which the ADCs service are generally optical photon counters sensitive to single photon hits. However, at this level of sensitivity single photon level noise is prevalent in all optical detector used to date. This noise, known as dark counts, is usually on the order of a single photon signal, which makes a single channel incapable of discriminating false positives. Only via the global triggering system can the validity of an event be ascertained. On the low event rate experiments, the trigger thresholds are tuned to detect a single photon hit, and thus are subject to a lot of noise.[23] Furthermore, the physics process of interest is usually buried in the common physical processes which occur at a much higher rate. This means that the triggering system must discriminate between interesting physics, common physical processes and detector noise. If the common physical processes are not filtered at the triggering level, the resulting digitization dead time results in an unusable detector.

Napali, in conjunction with additional hardware, could support uninteresting event rejection in low event rate physics experiments utilizing waveform samplers. Most of the machinery used by Napali is already present. Each device contains a waveform buffer, and is able to transmit some or all of it to the sink at the triggering systems discretion. If the single bit discriminator is instead replaced with a 3bit ADC, the Napali statistical triggering system can be used to filter unwanted events without compromising trigger efficiency. Furthermore, the 3 bit digitizer could be used for both high energy prompt and low energy delayed event without compromising detector dead time.

5.2 Future Work

There are still several unanswered question with regards to Napali benefits. From the claims described in Section 1.3, the power failure resiliency claim, and the privacy claim remain unproven. A strategy for evaluating these claims is described in the following sections.

5.2.1 Power failure resiliency

In order to evaluate Napali power failure resiliency, one of two methods must be employed. The most straightforward way is to add a battery backup the to OPQ Box and perform another evaluation deployment. The OPQ Box is already designed for battery backup, since the entire device (both the mains and the isolated side), can be powered from a common port. Furthermore, the expansion header on the OPQ Box is designed to accept as well as deliver power. A block diagram for the proposed battery backup subsystem is shown in Figure 5.1. A single cell 14430 LiFePo battery with a capacity of $\approx 1Wh$ could power the OPQ Box for 2 hours in case of a power outage. A charge controller would keep the battery cell charged during normal operation and the DC-DC boost converter would provide 5V to the OPQ circuitry in case of a mains failure. This

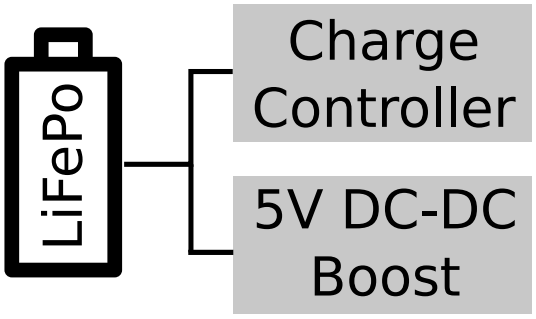


Figure 5.1: Proposed battery backup subsystem.

would allow the OPQ Box to record all of the data leading up to the power failure, and if the power outage is shorter than 2 hours capture the moment the power resumes. A PCB and the battery cell would fit into the existing OPQ Box enclosure with minimum modifications.

Another way to provide power failure resiliency is to add non-volatile memory to the sampler DSP. Every measurement taken by the DSP would be transferred to the non-volatile memory utilizing it as a circular buffer. This memory would serve as a black box recording the last measurements taken by the OPQ Box prior to the power failure. Flash is unsuitable for this due to a relatively small number of write cycles prior to failure, however FRAM devices would be perfect for this application. 2Mb FRAM devices are commonly available on the market, and would allow for an 11s recording window. The write endurance of 10^{14} IOPs results in the device remaining functional for 6000 years. Data retention of a common FRAM device is 5 years allowing for the data to remain retrievable through the worst of the power outages. The disadvantage of the non-volatile approach is the device would not be able to capture the moment the power delivery resumes.

5.2.2 Privacy Implications

In order to compare the privacy implications of power quality monitoring, and evaluate how Naive and Self-Triggered methods compare, a residential deployment of the OPQ devices must be carried out. Due to the time constraints such deployment was outside of the scope of this dissertation, however it would be straight forward to carry out. The main goal of this evaluation would be to empirically measure how much of the end user's activity can be ascertained from the local events which are recorded at their residence. As such the subjects of these evaluation would deploy the OPQ device at their place of residence and carefully record the timestamp of every electrical appliance they interact with. From laboratory tests, the OPQ Box was able to detect voltage drops associated with common electrical equipment as shown in Figure 1.2. Since Napali ignores these events, it is expected that it would fare better with respect to end user

privacy compared to the Naive and Self-Triggered methods. Napali does however transmit extracted metrics, however these are aggregated at 1s intervals. It's unclear how much privacy reduction the Napali methodology would be responsible for, or if the aggregation window length has an impact on privacy as a whole.

5.2.3 Grid Hierarchy Clustering on a Larger Scale

In Section 4.4.1, I described how the grid hierarchy can be extracted from subthreshold information in events captured by Napali. The advantages of event clustering are significant:

- Napali scalability can be linearized by concentrating subthreshold triggering to the cluster boundaries.
- Napali can localize the event origin in the grid hierarchy by examining affected clusters.

More research is required in order to better understand how to build the hierarchical map of the grid via subthreshold events in an organic automated way. This can potentially be accomplished by adding an additional layer above the Napali trigger in order to cluster and organize events from new devices added to the network. Every time a device is added to the network, Napali would start by acquiring its subthreshold data if requested regardless of which cluster the over-threshold event originated from. As enough data is acquired, the new device will be placed in the appropriate cluster in the triggering hierarchy, and restart normal operations.

5.2.4 Artificial Intelligence Integration into the Napali Trigger.

It is unclear if artificial intelligence approaches such as machine learning would fare better than the statistical trigger currently employed in Napali. While outside the scope of this dissertation, the OPQ deployment provided a large database of metrics and events for training and testing AI constructs. Furthermore, the Makai triggering service is flexible enough to accommodate any number of triggering algorithms alongside Napali for evaluation, characterization and comparison. The University of Hawaii power grid and the OPQ network remain a perfect test bed for evaluation of new and emerging power quality detection techniques through its flexible software architecture from hot-plugable metric extraction, to hot-plugable event detection and capture.

5.3 Summary of contributions

In this dissertation I showed that Napali provides a novel architecture that is both a feasible solution to the problem of distributed power quality monitoring and that provides significant benefits over the two standard alternative architectures (all computation/storage at nodes, all computation/storage at the sink). This was performed via demonstration of validity of the five claims

stated in Sections 1.6. The claims and contributions of this dissertation are summarized in the sections below.

5.3.1 Napali minimizes bandwidth usage

Section 4.3.1 shows the empirical comparison of Napali, Self-Triggered and Naive triggering system bandwidth consumption in the case of the OPQ deployment. Naturally, Napali and Self-Triggered methods outperformed the Naive method when it came to bandwidth consumption by a factor of 100x. By only selecting the anomalous temporal regions for readout, significant bandwidth reduction was observed. Furthermore, Napali outperformed the Self-Triggered method by further downselection of events to those that impact the power grid. While the Napali events were larger in size since they constituted raw waveform from multiple devices, they were far fewer in number. As such Napali was able to achieve a 4x bandwidth reduction over the Self-Triggered method, emerging as a clear winner when it comes to bandwidth consumption.

5.3.2 Napali minimizes sink processing requirement

Section 4.3.1 shows the analytical comparison of Napali, Self-Triggered and Naive triggering system bandwidth consumption in the case of the OPQ deployment. Naturally, the Self-Triggered method fared the worst in this evaluation, since every sample from every device needed to be processed on the sink. For Napali, overhead in the event detection is minuscule, since the metric comparison and filtering is linear with the number of devices, resulting in a modest resource consumption increase compared to the Self-Triggered method. The Self-Triggered method arguably requires no sink resources when it comes to event detection, however the higher levels of event evaluation are impacted by the additional data burden. For every false positive event that the Self-Triggered framework captures, an additional computational cost is incurred. While the characterization of this cost is beyond the scope of this work, it was shown that the incurred cost is significant when compared to Napali.

5.3.3 Napali mitigates device latency effects

Section 4.3.3 shows the analytical comparison of Napali, Self-Triggered and Naive triggering system device latency in the case of the OPQ deployment. Similar to the previous section, no direct comparison between Napali and Self-Triggered event detections methods could be made, since there is no latency impact on the Self-Triggered framework. The Naive triggering method, on the other hand, required an extremely large waveform buffer in order to accommodate latencies commonly seen by the OPQ Boxes during the University of Hawaii deployment. Napali was easily able to accommodate latencies of up to an hour by utilizing the waveform buffer in the device itself. This is significant, since power quality problems could easily result in parts of the network

infrastructure such as WIFI access points and routers becoming unavailable.

5.3.4 Gridwide monitoring via leaf nodes

Section 4.3.5 shows that the edge computing centered approach to power quality monitoring can result in high quality full and partial gridwide event detection via leaf node monitoring. The main evaluation of this claim comes from the ground truth measurements delivered via the Utility scale power monitors deployed across the University of Hawaii Campus. Napali was able to capture every event observed by the meters located higher on the power grid hierarchy apart from localized single phase faults. The main contribution of this claim is that the power grid can be monitored from regular household voltages without any contribution from the utility company. This opens a door to an oversight mechanism to monitor Utility due diligence when it comes to the power quality standard. Furthermore, OPQ was demonstrated to be a reliable system which can be used in conjunction with, but independently from, the Utility for power grid protection. With the increasing concern about Utility cyber security, an additional independent system is both desired and required.

5.3.5 Sub-threshold data acquisition is a viable event detection strategy

Section 4.3.6 describes the subthreshold detection capability of Napali in the OPQ network. The main claim of this section is that Napali is able to capture full and partial gridwide events via subthreshold triggering. Evaluation of this claim was accomplished via ground truth comparison. While the pool of the ground truth confirmed partial gridwide events which are useful for sub-threshold evaluation is small, Napali was able to both capture the over threshold and under threshold waveforms associated with each one. The main contribution of this claim is that the subthreshold event component is useful in event detection, localization and system scalability. The consequence of this is shown in Sections 4.4.1 and 4.4.1 where the University power grid was partitioned into the two substations using the subthreshold data, something that would be impossible to do via the Self-Triggered event detection method.

5.3.6 Open Power Quality System

Another contribution of this work is the Open Power Quality monitoring system. OPQ Box device has been fully characterized against synthetic data in the laboratory setting as shown in Section 4.1, and compares favorably to the commercial offerings. Furthermore the metric extraction algorithms utilized in the OPQ Box have been compared to the Utility meter counterparts in Section 4.3.5, to once again favorable agreement. The backend infrastructure had 97% availability during the deployment, brought down only by a power outage and system updates.

5.3.7 Napali in other domains

Applications of Napali in other domains are discussed in Section 5.1. This evaluation shows that Napali usability is not limited to the power quality domain. Instead Napali's edge-centric sub-threshold detection approach to event detection localization and capture is novel, useful, and can be applied to a variety of important domains.

BIBLIOGRAPHY

[1] Iot market predicted to double by 2021, reaching $520b. https://www.forbes.com/sites/louiscolumbus/2018/08/16/iot-market-predicted-to-double-by-2021-reaching-520b/#734d45911f94. Accessed: 2018-09-30.

[2] Pqube specifications. https://www.powerstandards.com/product/pqube-classic/specifications/. Accessed: 2018-09-30.

[3] IEEE recommended practice for monitoring electric power quality. *IEEE Std 1159-2009 (Revision of IEEE Std 1159-1995)*, pages c1–81, June 2009.

[4] S Abe, T Ebihara, S Enomoto, K Furuno, Y Gando, K Ichimura, H Ikeda, K Inoue, Y Kibe, Y Kishimoto, et al. Precision measurement of neutrino oscillation parameters with kamland. *Physical Review Letters*, 100(22):221803, 2008.

[5] Christy Anthony. *Laha: a Framework for Adaptive Optimization of Distributed Sensor Frameworks*. PhD thesis, University of Hawaii at Manoa, 2020.

[6] Ajay Kumar Bandi, Maher Rizkalla, and Paul Salama. A novel approach for the detection of gunshot events using sound source localization techniques. In *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 494–497. IEEE, 2012.

[7] Frede Blaabjerg, Remus Teodorescu, Marco Liserre, and Adrian V Timbus. Overview of control and grid synchronization for distributed power generation systems. *IEEE Transactions on industrial electronics*, 53(5):1398–1409, 2006.

[8] Erin R Burkett, Douglas D Given, and Lucile M Jones. Shakealertan earthquake early warning system for the united states west coast. Technical report, US Geological Survey, 2014.

[9] Kenneth L Cummins, Martin J Murphy, Edward A Bardo, William L Hiscox, Richard B Pyle, and Alburt E Pifer. A combined toa/mdf technology upgrade of the us national lightning detection network. *Journal of Geophysical Research: Atmospheres*, 103(D8):9035–9044, 1998.

[10] P Daponte, M Di Penta, and G Mercurio. Transientmeter: A distributed measurement system for power quality monitoring. *IEEE Transactions on Power Delivery*, 19(2):456–463, 2004.

[11] D De Yong, S Bhowmik, and Fernando Magnago. An effective power quality classifier using wavelet transform and support vector machines. *Expert Systems with Applications*, 42(15-16):6075–6081, 2015.

[12] Francesco Finazzi and Alessandro Fassò. A statistical approach to crowdsourced smartphone-based earthquake early warning systems. *Stochastic environmental research and risk assessment*, 31(7):1649–1658, 2017.

[13] Joshua D Gagliardi and Timothy S Munger. Content delivery network, June 14 2011. US Patent 7,962,580.

[14] Irene YH Gu, Nichlas Ernberg, Emmanouil Styvaktakis, and Math HJ Bollen. A statistical-based sequential method for fast online detection of fault-induced voltage dips. *IEEE Transactions on Power Delivery*, 19(2):497–504, 2004.

[15] Joel Höglund, Dejan Ilic, Stamatis Karnouskos, Robert Sauter, and Per Goncalves Da Silva. Using a 6lowpan smart meter mesh network for event-driven monitoring of power quality. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pages 448–453. IEEE, 2012.

[16] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, New York, NY, USA, 3rd edition, 2015.

[17] K Kahle. Power converters and power quality. *arXiv preprint arXiv:1607.01556*, 2016.

[18] Masoud Karimi, Hossein Mokhtari, and M Reza Iravani. Wavelet based on-line disturbance detection for power quality applications. *IEEE Transactions on Power Delivery*, 15(4):1212–1220, 2000.

[19] Muhammad Anas Khalid, Mohammad Inayatullah Khan Babar, Mohammad Haseeb Zafar, and Megat Farez Zuhairi. Gunshot detection and localization using sensor networks. In *2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, pages 1–6. IEEE, 2013.

[20] Martin Klapez, Carlo Augusto Grazia, Simone Zennaro, Matteo Cozzani, and Maurizio Casoni. First experiences with earthcloud, a low-cost, cloud-based iot seismic alert system. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 262–269. IEEE, 2018.

[21] Shang Li and Xiaodong Wang. Monitoring disturbances in smart grids using distributed sequential change detection. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 432–435. IEEE, 2013.

[22] Shang Li and Xiaodong Wang. Cooperative change detection for voltage quality monitoring in smart grids. *IEEE Trans. Information Forensics and Security*, 11(1):86–99, 2016.

[23] Viacheslav A Li, R Dorrill, MJ Duvall, J Koblanski, S Negrashov, M Sakai, SA Wipperfurth, K Engel, GR Jocher, JG Learned, et al. Invited article: minitimecube. *Review of Scientific Instruments*, 87(2):021301, 2016.

[24] Yong Liu, Shutang You, Wenxuan Yao, Yi Cui, Ling Wu, Dao Zhou, Jiecheng Zhao, Hesen Liu, and Yilu Liu. A distribution level wide area monitoring system for the electric power grid–fnet/grideye. *IEEE Access*, 5:2329–2338, 2017.

[25] Johan Morren, Sjoerd WH De Haan, Wil L Kling, and JA Ferreira. Wind turbines emulating inertia and supporting primary frequency control. *IEEE Transactions on power systems*, 21(1):433–434, 2006.

[26] Mohd Izhwan Muhamad, Norman Mariun, and Mohd Amran Mohd Radzi. The effects of power quality to the industries. In *Research and Development, 2007. SCOReD 2007. 5th Student Conference on*, pages 1–4. IEEE, 2007.

[27] A Nag, MJ Murphy, A Pifer, KL Cummins, and J Cramer. Upgrade of the us national lightning detection network in 2013. In *AGU Fall Meeting Abstracts*, 2013.

[28] Jessica Oueis, Emilio Calvanese Strinati, and Sergio Barbarossa. The fog balancing: Load distribution for small cell cloud computing. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–6. IEEE, 2015.

[29] Anthony C Parsons, W Mack Grady, Edward J Powers, and John C Soward. A direction finder for power quality disturbances based upon disturbance power and energy. In *Harmonics and Quality of Power Proceedings, 1998. Proceedings. 8th International Conference On*, volume 2, pages 693–699. IEEE, 1998.

[30] S Peisert, R Gentz, J Boverhof, C McParland, S Engle, A Elbashandy, and D Gunter. Lbnl open power data. 2017.

[31] Boštjan Polajžer, Gorazd Štumberger, and Sebastijan Seme. Evaluation of different methods for voltage sag source detection. *Energija*, 59(1-4):0–0, 2017.

[32] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[33] Y-J Shin, Edward J Powers, Mack Grady, and Ari Arapostathis. Power quality indices for transient disturbances. *IEEE Transactions on Power Delivery*, 21(1):253–261, 2006.

[34] Pedro LM Silva, José A Afonso, Vítor Duarte Fernandes Monteiro, JG Pinto, and João L Afonso. Development of a monitoring system for electrical energy consumption and power quality analysis. 2017.

[35] S. Sivanagaraju. *Electric Power Transmission and Distribution.* Dorling Kindersley, 2008.

[36] Xiang Sun and Nirwan Ansari. Edgeiot: Mobile edge computing for the internet of things. 54:22–29, 12 2016.

[37] RA Uhrhammer, M Hellweg, K Hutton, P Lombard, AW Walters, E Hauksson, and D Oppenheimer. California integrated seismic network (cisn) local magnitude determination in california and vicinity. *Bulletin of the Seismological Society of America*, 101(6):2685–2693, 2011.

[38] Alexandra Von Meier, David Culler, Alex McEachern, and Reza Arghandeh. Micro-synchrophasors for distribution systems. In *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, pages 1–5. IEEE, 2014.

[39] Kaiqiang Wang, Minwei Shen, Junguk Cho, Arijit Banerjee, Jacobus Van der Merwe, and Kirk Webb. Mobiscud: A fast moving personal cloud in the mobile network. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pages 19–24. ACM, 2015.

[40] Shiqiang Wang, Guan-Hua Tu, Raghu Ganti, Ting He, Kin Leung, Howard Tripp, Katy Warr, and Murtaza Zafer. Mobile micro-cloud: Application classification, mapping, and deployment. In *Proc. Annual Fall Meeting of ITA (AMITA)*, 2013.

[41] Dong-Jun Won, Il-Yop Chung, Joong-Moon Kim, Seung-Il Moon, Jang-Cheol Seo, and Jong-Woong Choe. A new algorithm to locate power-quality event source with improved realization of distributed monitoring scheme. *IEEE Transactions on Power Delivery*, 21(3):1641–1647, 2006.

[42] Anton Zaicenco and Iain Weir-Jones. Lessons learned from operating an on-site earthquake early warning system. In *PDF). Proceedings, World Conference on Earthquake Engineering. 15th*, volume 10, 2012.

[43] Dahai Zhang, Yanqiu Bi, and Jianguo Zhao. A new data compression algorithm for power quality online monitoring. In *Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on*, pages 1–4. IEEE, 2009.

[44] Yingchen Zhang, Penn Markham, Tao Xia, Lang Chen, Yanzhu Ye, Zhongyu Wu, Zhiyong Yuan, Lei Wang, Jason Bank, Jon Burgett, et al. Wide-area frequency monitoring network (fnet) architecture and applications. *IEEE Transactions on smart grid*, 1(2):159–167, 2010.