ABSTRACT

Edge Computing with an "Internet of Things" Based Sensor Array: An Innovative Approach to Near Real Time Seismic Exploration and Monitoring

Frank Sepulveda, Ph.D.

Mentor: Jay Pulliam, Ph.D.

We demonstrate the feasibility of leveraging an Internet of Things (IoT)-based sensor array to orchestrate edge-based (i.e., in a field setting) storage and computing resources capable of characterizing the subsurface, using ambient seismic noise, in near real-time. Our approach enables the continuous assessment of results and the identification of opportunities to modify the sensor array in more optimal configurations depending upon ambient noise field characteristics. Moreover, we can assess the need to leave the array in place for longer or shorter than originally planned with high levels of confidence our survey objectives have been met.

Over the course of our four deployments (i.e., Texas – May 2017, Nevada – June 2017, Texas – July 2018, and Nevada – May 2019) we developed an edge-based framework that utilized commercially available communication infrastructure, digitizers, embedded systems, and an established distributed database (i.e., DataStax Enterprise; DSE) to store and process sensor array data, in a field setting. This framework allowed us to overcome real-world performance limiters (e.g., bandwidth, power, etc.) commonly encountered while carrying out remote seismic exploration or monitoring. Moreover, it

provides an alternative solution to centralized (i.e., cloud-based) data storage and processing strategies that often have more demanding network capacity and reliability requirements.

The use of DSE (powered by Apache Cassandra), as our edge-based distributed database, is central to the scalability and reliability of our framework. To the best of our knowledge, no one has attempted to use DSE or Cassandra, on an embedded system, as a seismic sensor array's edge-based datastore. Our use of DSE is beneficial in the following three ways: 1) it supports the highly scalable write-heavy workloads common to sensor arrays, 2) it allows for the use of the same fault tolerant distributed database across a variety of commercially available hardware (e.g., embedded systems, servers, etc.), and 3) it seamlessly maintains and replicates data along a user defined continuum of locations (i.e., "edge to cloud"). We believe geoscientists can use our edge-based solution to improve existing and develop novel methods to characterize the subsurface.

Edge Computing with an “Internet of Things” Based Sensor Array: An Innovative Approach to Near Real Time Seismic Exploration and Monitoring

by

Frank Sepulveda, B.A., B.S., M.S.

A Dissertation

Approved by the Department of Geosciences

____

Steven G. Driese, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Doctor of Philosophy

Approved by the Dissertation Committee

____

Robert J. Pulliam, Ph.D., Chairperson

____

Vincent S. Cronin, Ph.D.

____

John A. Dunbar, Ph.D.

____

Steven I. Dworkin, Ph.D.

____

Michael W. Thompson, Ph.D.

Accepted by the Graduate School
December 2020

____

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

To my Ph.D. advisor, Dr. Jay Pulliam, I would like to express my sincerest appreciation for your encouragement and support over the years. Honestly, I did not always understand the “why”; however, once I learned to trust your judgement, the “what” I wanted to accomplish as a Ph.D. candidate elevated from a personal hobby to a grand endeavor. Most importantly, you helped smooth out some of the sharp edges of my personality that had held me back in my personal and professional life. You help me be better person and I am forever grateful.

To all those wonderful people in the Department of Geosciences, thank you for being so kind and supportive. Paulette Penney, you were never once too busy to sit and chat with me about family, school, or work. And I know it was not because you did not have plenty to do. Dr. John Dunbar, I truly enjoyed our conversations. You are a man with an unbelievable amount of knowledge; I am fortunate to have had an opportunity to learn from you. To my best friend Tim Meredith, the opportunity to go “…once more unto the breach…” with you made the past five or so years so much more enjoyable. You are my hero for many reasons, but mostly because you are such a good person.

I would like to thank some of the friends and mentors that helped me find my way to Baylor. Scott Clay, thank you for encouraging me to pursue a Ph.D. Lisa Bradshaw, you moved heaven and earth to find me a way to work and go to school, thank you. Last, Heather Kirschner and Dr. James Gridley, thank you for seeing me as more than a dusty field engineer and believing in me.

DEDICATION

For my Dad

CHAPTER ONE

Introduction

This dissertation represents the conclusion of six years of research comprising the design, development, and deployment of a system capable of acquiring, transmitting, and processing seismic data in a field setting (i.e., “the edge”). Exploiting 20 years of technology advancements, we integrated commercially available data acquisition systems, a distributed datastore, embedded systems, and telemetry into an edge storage and computing framework for seismic sensor networks. This framework provides geoscientists a solution to implement existing, or to develop novel, methods to characterize the subsurface in near real-time.

In 2016, Dr. Jay Pulliam and I integrated commercial off the shelf components, an embedded system and a data acquisition system used widely in geophysics, to create an Internet of Things (IoT)-based device that allowed for single-node processing of seismic data, in a field setting. This work, which resulted in an IoT-based sensor network node capable of performing single-station seismic processing, was published in Seismological Research Letters (Chapter Two).

Our proposal to develop a novel, near real time approach to geothermal seismic exploration and monitoring via ambient seismic noise interferometry was supported by funding from the U.S. Department of Energy (Energy Efficiency and Renewable Energy Agency, Geothermal Technologies Program, Award Number: DE-EE0007699). The principal investigator was Dr. Pulliam and project partners included Mr. Joe Iovenitti

(Consulting Geologist) and Drs. John and Marge Queen (Hi-Q Geophysical, Inc.). Joseph Soloman Thangraj (Baylor University Ph.D. Candidate) and Diego Quiros (Baylor University Postdoctoral Fellow) supported seismic analysis, Tim Meredith (Baylor University Geoscience Instrumentation Specialist) designed and implemented wireless telemetry, and Dr. John Dunbar performed deployment surveys. All collaborators participated in multiple field deployments.

Over the course of our four deployments (i.e., Texas – May 2017, Nevada – June 2017, Texas – July 2018, and Nevada – May 2019) the advantages of real-time processing were realized in a field setting. These advantages included: 1) the ability to continuously assess our results to determine if re-deployment in a more optimal configuration was necessary and 2) the early demobilization of the array if survey objectives have been achieved. The project period was between October 2016 and August 2019, and the final report was submitted December 2019. The final report will be adapted for future publication in Journal of Geophysics and Engineering or another comparable journal (Chapter Three).

Dr. Pulliam and I incorporated lessons learned over the course of our field deployments to formalize a general purpose edge storage and computing framework for IoT-based sensor networks. Geoscientist and others engaged in remote environmental monitoring could leverage this framework to support applications in which: 1) the dense acquisition of data is occurring, 2) low latency is required, or 3) network connectivity is either constrained or nonexistent. This work was submitted to Sensors for publication and is currently under review (Chapter Four).

Chapter Five describes new or prospective innovations that could be incorporated into an IoT-based sensor array, like the one described here, to extend geophysical exploration or monitoring functionality and performance. Lastly, “best practice” recommendations for successful development, testing, and data acquisition/processing are summarized and additional applications of the IoT-based sensor array are discussed.

CHAPTER TWO

The Internet of Geophysical Things: Raspberry Pi Enhanced REF TEK (RaPiER) System Integration and Evaluation

This chapter published as: Frank Sepulveda and Jay Pulliam (2016), The Internet of Geophysical Things: Raspberry Pi Enhanced REF TEK (RaPiER) System Integration and Evaluation, Seismological Research Letters 2016;87:345-57.

Abstract

The proliferation of commercial Internet of Things (IoT) devices is raising consumers' awareness of the benefits of enhancing everyday objects with the ability to communicate, sense, and process information. Commercial-off-the-shelf (COTS) versions of the embedded technologies responsible for the rise of the IoT are easy-to-use, inexpensive, and relatively powerful. IoT Makers, technological do-it-yourself enthusiasts, utilize these technologies and the IoT ecosystem to create IoT devices that range from casual hobbyist to entrepreneurial in nature.

In an effort to develop a "no-engineer-needed" Internet of Geophysical Things (IoGT) device, we integrated a COTS embedded computer with a geoscience-related COTS data acquisition system (DAS). Using skills common to geoscientists, we integrated the Raspberry Pi System-On-Chip with the REF TEK (a Trimble brand) 130-01 DAS. This Raspberry Pi enhanced REF TEK (RaPiER) provides IoGT Makers a platform for the development of geoscientific sensor node or network enhancements. IoGT Makers can use the RaPiER to tinker with IoT capabilities while simultaneously acquiring research quality data. Geoscientific (i.e., seismic) application areas that may

benefit from RaPiER, particularly RaPiER nodal processing, include earthquake and engineering seismology.

In this article, we discuss the emerging technologies that allow IoGT Makers to build their own IoGT device. We review our selection of the Raspberry Pi and REF TEK 130-01 for IoGT device integration. We provide a RaPiER system integration guide and REF TEK interface software compiled for the Raspberry Pi (Ⓔ available in the electronic supplement to this article). We present methods to expand RaPiER capabilities using the Raspberry Pi's ecosystem. Last, we discuss our evaluation of RaPiER's performance and its suitability for realistic field deployments.

Introduction

To tech-savvy consumers, the Internet of Things (IoT) represents a long-awaited technological utopia: a plethora of revolutionary IoT devices unified in the automatic and unobtrusive performance of day-to-day tasks intended to improve the quality of human life. Manufacturers selling these devices brand their wares with "IoT" as a buzz phrase intended to ascribe these devices with "smarts" or "intelligence." The use of IoT as a buzz phrase is similar to the use of the buzz phrases "cyber-this…" or "e-that…" that accompanied the rise of the Internet (Miller, 2015). And, as with the Internet, the expectation is that the IoT will revolutionize the world. However, the timeframe for its large-scale adoption and its socioeconomic impact remain uncertain (McEwen and Cassimally, 2014; Greengard, 2015; Miller, 2015).

The proliferation and connection of IoT devices has commenced (Miller, 2015); however, there remains a general lack of understanding regarding what the IoT actually is. Kevin Ashton, inventor of the phrase "Internet of Things," describes the IoT as computers capable of sensing and making sense of the world without the need for extraneous human interaction (Ashton, 2009; Gabbai, 2015). In practice, the IoT represents a set of everyday objects enhanced with devices that are capable of sensing, processing, and communicating with each other to achieve a common objective, on a scale that was previously unattainable (Whitmore et al., 2014).

In the past, the creation of "enchanted objects," a metaphor used by McEwen and Cassimally (2014) to describe IoT devices, was limited to those with seemingly arcane knowledge of computer science or engineering. With the rise of the Maker Movement (Voight, 2014), a cadre of technological do-it-yourselfers create their own IoT devices. Leveraging commercial-off-the-shelf (COTS) components, Makers participate in a rich IoT Maker ecosystem that allows all levels of users, from hobbyist to entrepreneur, to bootstrap projects that range from the useless to the essential (see Data and Resources).

In the spirit of the Maker Movement, we have "enchanted" a COTS data acquisition system (DAS) with an inexpensive and easy-to-use embedded computer to create our own Internet of Geophysical Things (IoGT) device. Our "no-engineer-needed" IoGT device integrates the Raspberry Pi with the REF TEK (a Trimble brand) 130-01. With 130-01 and REF TEK Interface (RTI) software capabilities intact, the Raspberry Pi Enhanced REF TEK (RaPiER) allows geoscientists to leverage the Raspberry Pi's "embarrassment of riches" (McEwen and Cassimally, 2014) and its Maker ecosystem to bootstrap myriad novel sensor node or network enhancements.

In this article, we discuss the emerging technologies that allow IoGT Makers, using skills common to geoscientists, to build their own IoGT device. We review the selection of the Raspberry Pi and REF TEK 130-01 for our IoGT device system integration. We provide a RaPiER system integration guide. We present methods to expand RaPiER capabilities using guides obtained from the Raspberry Pi's ecosystem. Last, we discuss RaPiER performance and its suitability for realistic field deployments. Refer to Table 2.1 for a complete list of acronyms that appear throughout this article.

Table 2.1 List of Acronyms

Acronym	Definition
bps	Bits per second
CF	Compact flash
COM	Computer-on-module
COTS	Commercial-off-the-shelf
CPU	Central processing unit
DAS	Data acquisition system
GPIO	General purpose input/output
GPU	Graphics processing unit
GSN	Global Seismic Network
IoGT	Internet of Geophysical Things
IoT	Internet of Things
IRIS	Incorporated Research Institution for Seismology
LAN	Local-area network
OEM	Original equipment manufacturer
PASSCAL	Portable Array Seismic Studies of the Continental Lithosphere
PPSD	Probabilistic power spectral density
RaPiER	Raspberry Pi Enhanced REF TEK
RPF	Raspberry Pi Foundation
RTI	REF TEK interface
SSH	Secure shell
SSMTP	Secure simple mail transfer protocol
SOC	System-on-chip
SWaP	Size, weight, and power
VSFTPD	Very secure FTP Daemon
WAN	Wide-area network
WSN	Wireless sensor network

Trends in Big Data and The Internet of Things

Big Data is an emerging technology defined by excesses in the “volume, variety, velocity, and value” of all kinds of data (Schutt and O’Neil, 2013). This includes conventional time series, geolocation, image, mobile, network, sensor, and social network data (Schutt and O’Neil, 2013). Although the IoT is not responsible for the glut of data currently besetting companies, the expectation is that, once the IoT “goes live,” its data will inundate and overwhelm those who try to use it (Baldwin, 2014).

The current state of Big Data and the IoT, as per the Gartner hype cycle for emerging technologies (LeHong and Fenn Gartner, 2014), places Big Data within the cycle’s “trough of disillusionment” and the IoT at the apogee of the cycle’s “peak of inflated expectations.” Big Data has been replaced at the top of the peak of inflated expectations in the previous year’s cycle by the IoT (Press, 2014). Considering that IT executives are almost evenly split between the IoT being worthy of its buzz or entirely overhyped (Bertolucci, 2014), it is important that we consider carefully the potential scientific value of Big Data resulting from IoT devices.

Cukier and Mayer-Schoenberger (2013) present three aspects to embrace in order to use Big Data: (1) collect and use a lot of data, (2) accept that the benefits of using a lot of data outweigh the drawbacks of including shoddy data, and (3) abandon the desire to understand the cause of a thing. Although most geoscientists would agree on the benefits of collecting and using a lot of data, few may be willing to accept shoddy data or accept understanding “what happened” in lieu of “how it happened.” Big Data, particularly data resulting from IoT devices, may not be of sufficient quality for serious scientific study (Schutt and O’Neil, 2013).

Wireless Sensor Network Legacy

Past deployments of wireless sensor networks (WSNs) for geoscientific purposes can be used to guide IoGT device system integration. Geoscience WSNs are typically designed, developed, and tested by computer scientists or engineers collaborating with geoscientists to accomplish a specific objective. The lessons learned over the course of five years of working with geoscientists to deploy various WSNs for volcanic monitoring (Challen and Welsh, 2010), were used to guide our IoGT device system integration.

Challen and Welsh (2010) recognize the benefits of providing geoscientists more data resulting from a large number of inexpensive and low size, weight, and power (SWaP) sensor nodes. However, a desire for more data must be weighed in the context of finite WSN resources (e.g., bandwidth, power, etc.) and the desire to provide geoscientists with the type of high-quality data to which they are accustomed (Challen and Welsh, 2010).

By building upon multiple successful deployments and redefining geoscientists' expectations, considerable advances have been made with WSNs; however, this has not been without trade-offs (e.g., data quality, deployment duration, etc.) (Challen and Welsh, 2010). These trade-offs may force geoscientists to accept that the benefits of more data outweigh the drawbacks of shoddy (i.e., lower-quality) data (Challen and Welsh, 2010). Geoscientists may be reluctant to accept these data, because they may be unable to use the data with their existing processing techniques.

Challen and Welsh (2010) repeatedly stress the importance of developing WSNs within the context of finite WSN resources and an overarching scientific objective. Ideally, this requires the identification of the minimum quality or quantity of data required to achieve a scientific objective (Challen and Welsh, 2010). Unfortunately, data of minimum quality or quantity may result in WSNs that are too application specific or data that can only determine that a phenomenon has occurred but not how it occurred. Understanding how a phenomenon occurs is typically a geoscientist's primary objective.

Interestingly enough, the Challen and Welsh (2010) lessons learned mirror Cukier and Mayer-Schoenberger (2013) three aspects to embrace in order to use Big Data. Using their efforts as a guide, we selected and integrated our IoGT device components to provide geoscientists with (1) high-quality data that allows for the detection and characterization of geosciences-related phenomena, (2) a non-application-specific solution, and (3) a versatile, yet technologically accessible, solution.

System Integration Background

The joining of components, near the end of their development cycle, into a system capable of increased functionality is known as system integration (Technopedia, n.d.). IoT device system integrators, typically engineers working as professional Makers, apply their broad range of skills to develop high-quality products intended for commercial distribution. IoGT Makers, perhaps lacking the breadth and depth of computer science and engineering skills of professional system integrators, must select their IoGT device components carefully. RaPiER uses non-application-specific, technologically accessible components. Our selection of components allows geoscientists familiar with basic electronics, Linux, networking, and scripting languages the ability to build their own RaPiER.

Embedded Systems

An internet search of the term "embedded system" yields a wide range of products with applications ranging from hobbies to industrial. Embedded systems are often classified, by name more than by function, into categories such as microcontroller, embedded computer, system-on-chip (SOC), computer-on-module (COM), or system-on-module (SOM). We use the term "embedded computer" as a generic term to describe an embedded system.

Embedded computers, at their most basic level, consist of a processor and memory located on a single chip. Embedded computers are typically low SWaP, intended for specific applications, and have modest capabilities when compared with desktop or laptop computers. There is considerable variability in embedded computer performance specifications and peripheral capabilities (e.g., Bluetooth, WiFi, etc.).

Embedded Computer Selection

Three embedded computers were considered for RaPiER system integration: (1) the Arduino microcontroller, (2) the Gumstix COM, and (3) the Raspberry Pi SOC (see Data and Resources). The Arduino, admittedly the most popular platform for IoT device development (McEwen and Cassimally, 2014), was not selected due to its use of a bootloader or real-time operating system. To bootstrap REF TEK's existing RTI software, RaPiER's embedded computer must run a Linux operating system.

Gumstix and Raspberry Pi both use suitable Linux operating systems and offer relatively similar performance specifications. However, a Gumstix option is notably more expensive than a Raspberry Pi option. The Gumstix Overo COM costs \$100–\$230 and requires an expansion board that can range from \$30 to \$150. In February 2015, the

Raspberry Pi Foundation (RPF) released Raspberry Pi 2 model B to replace the Raspberry Pi 1 model B+. The Raspberry Pi 2 model B offers significant performance improvements compared to its predecessor and retails for approximately $40.

The Raspberry Pi possesses a number of other features that make it an ideal embedded computer for IoGT device development. The RPF provides a Debian-based Linux operating system (i.e., Raspbian) that is optimized for the Raspberry Pi, officially supported, easy to install, and easily configured. Gumstix operating systems are primarily third party and, in our opinion, considerably more difficult to install and configure.

Third-party expansion boards, known as "hats" or "shields," are available for purchase and interface with the Raspberry Pi via its general purpose input/output (GPIO) pins. These easily accessible GPIO pins also allow Makers to interface custom electronic circuits with the Raspberry Pi. Many of the projects found within the Raspberry Pi's milieu utilize GPIO pins to connect sensors and actuators. First- and third-party expansion boards are available for the Gumstix; however, they are often more expensive and not as easy to use.

Ultimately, it was the Raspberry Pi's ease of use and robust ecosystem that made it the most compelling embedded computer option. The Raspberry Pi's ecosystem caters to a wide range of Maker skill levels; it was originally intended to help develop computer science skills in children (McEwen and Cassimally, 2014). Novice Makers unfamiliar with basic electronics, Linux, and programming can easily find Raspberry Pi projects to develop the skills required to move on to more advanced projects. Experienced Makers will find the Raspberry Pi's ecosystem to be an ever-expanding source of valuable resources that support all types of projects.

Data Acquisition System Selection

To develop an IoGT device that was truly "no engineer needed," we would need a DAS that required a minimum amount of troubleshooting. The original equipment manufacturer (OEM) would have to provide hardware and software support, product documentation, and be willing to support novel uses of their products. The DAS would be capable of acquiring data from a variety of sensors and have reasonable input impedance, input voltage range, and number of input channels. Last, using a common port and a well-established protocol, the DAS would be capable of transmitting data to a computer in near-real time. Ideally, the geoscience community would already be familiar with the DAS and have confidence in the quality of the data it produces.

Our choice of the REF TEK 130-01 as RaPiER's DAS was strongly influenced by its broad acceptance and wide use within the geoscience community. REF TEK (i.e., Refraction Technology Inc.) became part of Trimble Navigation Limited in October 2012. For a listing of geoscience-related REF TEK (a Trimble brand) instrument deployments (i.e., digital telemetry networks) refer to Trimble's website (see Data and Resources). Although the 130-01 has been replaced by the 130S-01, 130-01 product documentation, training, and troubleshooting support are available via REF TEK support.

The REF TEK 130-01 transmits data, in near-real time, to a computer running REF TEK's RTI software via a common port (i.e., Ethernet or serial) and a well-established protocol (i.e., User Datagram Protocol [UDP]). Last, the 130-01 has been used to collect data for a variety of scientific and engineering applications.

The Incorporated Research Institutions for Seismology Portable Array Seismic Studies of the Continental Lithosphere (PASSCAL) Instrument Center and EarthScope

USArray Array Operations Facility maintains over 850 REF TEK 130-01s in its inventory (see Data and Resources). The 130-01s are available to research and educational institutions as per PASSCAL and USArray governing policies. Given its availability, capability, flexibility, performance, quality, and widespread use, we feel confident in choosing the 130-01 as RaPiER's DAS.

REF TEK has supported our novel use of the REF TEK 130-01. Unfortunately, the market for REF TEK software products specifically developed for embedded computers is limited. REF TEK support has indicated that they do not intend to provide this software to other customers and cannot support these untested and unsupported REF TEK products. REF TEK support has granted us permission to share the software we were provided. REF TEK can provide customers a sales quote for the development costs associated with an embedded computer version of their RTI software upon request.

System Integration

Embedded computers, unlike desktop or laptop computers, typically do not use conventional 32-bit or 64-bit processors (e.g., AMD FX, Intel Core i7, etc.). The Raspberry Pi 2 model B uses an Advanced RISC Machine (ARM) 32-bit Cortex-A7 processor. ARM processor architecture is ideal for embedded computing applications that require high performance while maintaining low SWaP (see Data and Resources). Unfortunately, this means that software compiled for conventional processors will not run on the Raspberry Pi.

It was necessary to obtain a version of REF TEK's RTI software compiled specifically for the ARM architecture. We were provided an untested and unsupported ARM architecture version of some of the RTI software modules by REF TEK. This software did not include documentation or installation scripts. It was necessary to develop our own installation and configuration process for RaPiER system integration. This required that we draw heavily from the Raspberry Pi's ecosystem and REF TEK's product documentation. To best assist IoGT Makers in performing their own RaPiER system integration, we provide the ARM architecture RTI software (Ⓔ available in the electronic supplement), Raspberry Pi and RTI installation and configuration scripts (Ⓔ available in the electronic supplement), and a detailed RaPiER system integration guide.

Raspberry Pi

The Raspberry Pi Foundation (n.d.) provides official documentation that details initial setup, operating system installation, usage, and configuration of the Raspberry Pi via their website. IoGT Makers should utilize the Raspberry Pi's ecosystem to review the differences in Raspberry Pi models, identify what peripherals are required, perform initial setup, install an operating system, and utilize the configuration tool, command line, and desktop environment. We recommend IoGT Makers unfamiliar with Linux review the RPF's step-by-step instructions and fundamental Linux usage documentation before getting started with RaPiER system integration. The preparation of a Raspberry Pi for RaPiER system integration will require IoGT makers to (1) select appropriate peripherals, (2) install an operating system, (3) set configuration options, and (4) modify Linux settings.

IoGT Makers should consider the following when selecting Raspberry Pi peripherals. Use a micro-SD card with a minimum of 8 GB of storage. The Raspberry Pi should be powered using a 5 V USB power supply with a 2 A output. Makers may encounter reliability issues that are difficult to troubleshoot when using a power supply with less than 2 A output. We recommend Makers use an externally powered USB hub to connect mouse, keyboard, and WiFi adapter to avoid power-related reliability issues. The Raspberry Pi 1 model B+ and Raspberry Pi 2 model B+ have four USB 2.0 ports that can provide up to 1.2 A for USB devices; however, a 2 A power supply is required (Gibbs, 2015). Makers should reference the Raspberry Pi's ecosystem to verify the peripheral devices are compatible prior to purchase.

The RPF recommends beginners start with Noobs, an install manager that allows users to select from a variety of first and third-party operating systems (Raspberry Pi Foundation, n.d.). We recommend that IoGT Makers bypass Noobs and proceed to install Raspbian Wheezy (see Data and Resources) as their operating system. RaPiER system integration was performed using Raspbian Wheezy.

The initial boot of a Raspbian Wheezy install will automatically display the Raspberry Pi Software Configuration Tool. This tool is also referred to as raspi-config. IoGT Makers will use raspi-config to prepare the Raspberry Pi for RaPiER system integration. The RPF provides raspi-config usage documentation. Using raspi-config, Makers should Expand Filesystem, Change User Password, and set Enable Boot to Desktop/Scratch to boot to Desktop Log in…. Once raspi-config setup changes are finalized (i.e., Finish), Raspberry Pi will now automatically logon the pi user to the Raspbian Wheezy desktop environment. Makers can return to the Raspberry Pi Software Configuration Tool by running.

```
$ sudo raspi-config
```

To complete the preparation of the Raspberry Pi for RaPiER system integration, IoGT Makers will use the RaPiER_RaspberryPi.sh script (Ⓔ available in the electronic supplement). This script will do the following: (1) create a reftek user, (2) assign reftek superuser privilege, (3) change the default auto-logon user to reftek, (4) change the default computer keyboard layout from United Kingdom to United States, and (5) change default network interface settings. Copy all of the contents located within the RaPiER_Project directory (Ⓔ available in the electronic supplement) to the /home/pi directory. Next, run

```
$ cd /home/pi
```

to navigate to the pi user home directory. Run the command

```
$ sudo chmod +x RaPiER_RaspberryPi.sh
```

to make the script executable. Run

```
$ sudo ./RaPiER_RaspberryPi.sh
```

in the command line to execute the script. Upon completion of the script, the Raspberry Pi will automatically reboot and log the reftek user onto the Raspbian Wheezy desktop environment.

Raspberry Pi as RTPD Data Server

IoGT Makers should set a password for the reftek user before continuing with RaPiER system integration. Run

```
$ sudo passwd reftek
```

and follow ensuing prompts to set a new password.

The REF TEK Protocol Daemon (RTPD) RTI software module provides for the near-real-time communication of data between the 130-01 and an RTPD data server. RTI archive utilities are used to create, copy, rebuild, retrieve, review, and write RTPD archives. For an RTPD archive to acquire data, all of the required RTI files and folders must be in the correct location and have the correct owner and permission. Run

```
$ cd /home/reftek
```

to navigate to the reftek user home directory. Run

```
$ ./RaPiER_REFTEK.sh
```

to execute the RTI software install and configuration script. IoGT Makers can start, stop, and check the status of RTPD service by running

```
$ service rtp {start, stop, status}.
```

To automatically start the RTPD service after the Raspberry Pi boots, run

```
$ sudo crontab –e.
```

Scroll to the very bottom of the command line interface and add

```
@reboot sh /etc/init.d/rtp start.
```

Save and exit when done. Reboot the Raspberry Pi.

The Raspberry Pi is now configured to operate as an RTPD server. However, it is necessary to establish a network connection between the Raspberry Pi and the REF TEK 130-01 and to create an RTPD archive to finalize RaPiER system integration. This requires IoGT Makers to (1) establish a physical connection (i.e., Ethernet) between the Raspberry Pi and the 130-01, (2) configure the Raspberry Pi with a static IP address (i.e., Ethernet), (3) configure the RTPD rtpd.ini file, (4) configure the 130-01 acquisition (i.e., channel and stream) and network settings (i.e., IP address and RTP address), and (5) create an RTPD archive.

To make RaPiER system integration easier for IoGT Makers unfamiliar with REF TEK's products, the Raspberry Pi network interface (i.e., Ethernet), RTPD rtpd.ini file, and RTPD archive were automatically configured and created during execution of the RaPiER_RaspberryPi.sh and RaPiER_REFTEK.sh scripts. The Raspberry Pi was configured with a static IP address of 192.168.1.100/24. The rtpd.ini file was configured with a command client IP address and discovery IP address of 192.168.1.100. The rtpd.ini file was configured with an archive directory of /home/reftek/data/archive, and the RTPD archive was created.

IoGT Makers can finalize RaPiER system integration by establishing a physical connection between the Raspberry Pi and the REF TEK 130-01, via a network switch (Fig. 2.1), and configuring the 130-01's acquisition and network settings. Connecting the 130-01 to a network switch requires REF TEK's 130 to Ethernet RJ45 hub cable (part number 130-8019). IoGT Makers unfamiliar with 130-01 acquisition and network configuration should refer to REF TEK's product manuals for guidance. With 130-01 acquisition started and the RTPD service running, RaPiER's RTPD archive should start building. RaPiER integration is now complete.

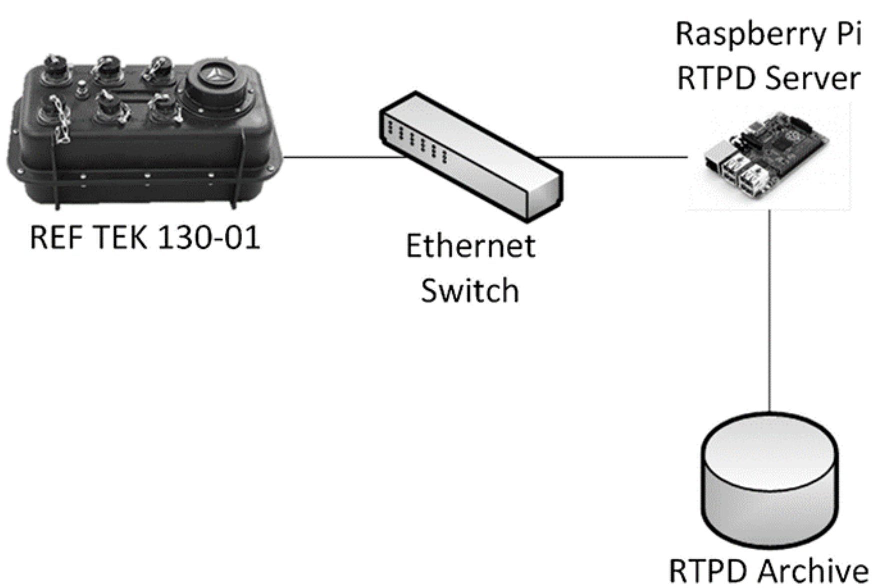


Figure 2.1. Raspberry Pi Enhanced REF TEK (RaPiER) concept.

There are several other Raspberry Pi configuration options IoGT Makers should consider. By default, the RaPiER system is configured to save RTPD archive data to the Raspberry Pi's micro-SD card (i.e., /home/reftek/data/archive). Once IoGT Makers are familiar with the RaPiER operation, we recommend they configure the Raspberry Pi to save RTPD archive data to a USB flash drive. Instructions are available, via the Raspberry Pi's ecosystem, for mounting a USB flash drive to a specific directory when the Raspberry Pi boots.

IoGT Makers intending to deploy their RaPiER to the field should use raspi-config to set Enable Boot to Desktop/Scratch to Console Text and set Advanced Options to enable secure shell (SSH). These modifications will reduce Raspberry Pi power consumption by allowing Makers to interface with RaPiER without the need for a desktop environment or peripherals. Refer to RPF documentation for raspi-config usage.

With RaPiER integration complete, we anticipate IoGT Makers will be eager to utilize the RaPiER to enhance their sensor nodes or networks; however, these efforts should not interfere with existing REF TEK 130-01 and RTI software capabilities. Typically, the 130-01 is configured to locally record data to a compact flash (CF) card, transmit data to a remote RTPD server, or both record locally and transmit data to a remote RTPD server. Collocation of a local RTPD server (i.e., Raspberry Pi) with a deployed 130-01 requires Makers configure the RaPiER to transmit data from the local RTPD server to the remote RTPD server. The collocation of a local RTPD server with a deployed 130-01 does not interfere with data recording to a CF card.

There are many methods to transmit data to multiple RTPD servers. The simplest method is for IoGT Makers to configure the REF TEK 130-01 to transmit data to two separate RTPD servers; however, this requires 130-01 firmware v.3.7.2 or later. An alternate method is the automatic synchronization of the local RTPD archive with a remote RTPD archive. Makers can accomplish this synchronization using RTPD "chaining" (Fig. 2.2). Refer to REF TEK's product manuals for details regarding 130-01 configuration and RTPD chaining.

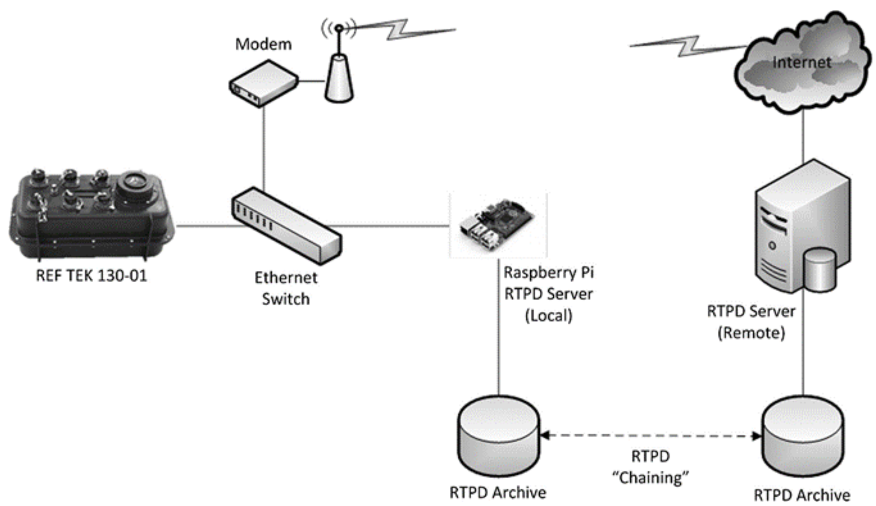


Figure 2.2. RaPiER, with local and remote RTPD servers shown.

Expanding RaPiER's Capabilities

To demonstrate the ease with which IoGT Makers can expand RaPiER capabilities, we present three guides, obtained from the Raspberry Pi's ecosystem, that allow Makers to configure the RaPiER as a data, e-mail, file transfer protocol (FTP), and webserver (Fig. 2.3). By expanding RaPiER capabilities, Makers can configure RaPiER to (1) analyze, manipulate, and store data in a client-server model, (2) send low-bandwidth notifications via e-mail, (3) allow remote access to the RTPD archive, and (4) monitor system performance.

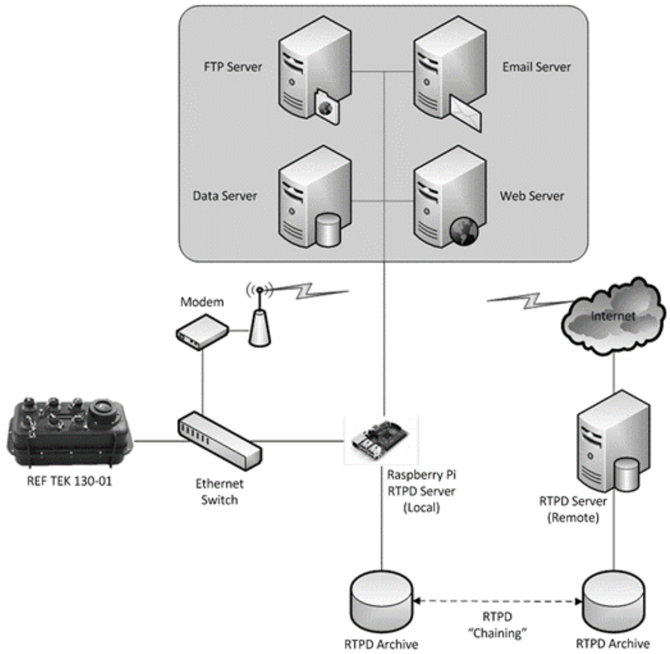


Figure 2.3. RaPiER with expanded capability shown.

With the RTPD archive as its database, the RaPiER can be configured as a data server. IoGT Makers (i.e., RaPiER owners) can configure remote access and control of RaPiER resources and allocate data-related services or tasks to the RaPiER (e.g., analysis, conversion, etc…). Client (i.e., RaPiER subscriber) requests would leverage the RaPiER's Raspberry Pi to perform useful data processing. There is no specific guide for RaPiER data server configuration; the use of RaPiER as a data server is a characteristic of the overarching RaPiER concept.

As an e-mail server, RaPiER can send its owner and subscribers notifications. These notifications could result from regularly scheduled activity or could be triggered in response to a change of state. Secure simple mail transfer protocol (SSMTP) (see Data and Resources) is a lightweight, low bandwidth, and reliable way for the RaPiER to send notifications via Email. IoGT Makers can easily incorporate SSMTP e-mail notifications into their shell or python scripts. Using SMTP Mail Setup (TNET, n.d.) as a guide, we set up our RaPiER as an e-mail server.

The REF TEK 130-01 can record data to a CF card locally, transmit data to a remote RTPD server, or both record locally and transmit data. Recording data locally on a 130-01's CF card is often considered a fail-safe in case network communication becomes unavailable. Unfortunately, 130-01 users must physically retrieve the CF card from the unit. The RaPiER's RTPD archive will continue to build even if remote network communication is lost. Once network communication is reestablished, the RaPiER's owner and/or subscribers can retrieve missing archive data via FTP.

Very secure FTP Daemon (VSFTPD; see Data and Resources) facilitates FTP server access and allows users to create custom user accounts with custom directories. Although not specifically intended for the Raspberry Pi, we found the Bourdeau (2012) VSFTPD setup guide to be a particularly good resource. Bourdeau's guide was used to configure our RaPiER as an FTP server.

IoGT Makers wishing to field a RaPiER will likely want to evaluate and monitor its system performance. Munin (see Data and Resources), a networked performance monitoring tool, allows the RaPiER's owner and subscribers to monitor Raspberry Pi performance data. Prior to the installation of Munin, it is necessary to configure the RaPiER as a webserver. "How to Host a Website with Raspberry Pi" (Orsini, 2014) was used as a guide to set up the Raspberry Pi as an Apache webserver (see Data and Resources). Munin was installed and configured using "Monitoring Raspberry Pi temperatures" (Langner, 2015) as a guide. Munin continuously monitors Raspberry Pi performance data (e.g., core temperature, central processing unit [CPU]/memory usage, etc.) and allows users to view or export data for analysis.

RaPiER Measures of Effectiveness

The REF TEK 130-01 has been in use since 2002 and has garnered a reputation as a reliable DAS. As a system intended for deployment to remote field stations with limited infrastructure and harsh environmental conditions, the 130-01 is lightweight and mechanically tough. REF TEK reports the 130-01's power consumption to be ∼1.4 W while recording three channels with Global Positioning System and communications active. Its operating temperature range is −20°C to +60°C.

In contrast, the Raspberry Pi was not developed with harsh environmental conditions and finite resources in mind. Nonetheless, as an embedded computer, the Raspberry Pi is thermally robust and has relatively low power requirements. According to the RPF, the operating temperature range of the Raspberry Pi 2 model B is determined by the lowest maximum and highest minimum operating temperature of its individual computer module components (i.e., −25°C to +80°C). The exact power consumption of Raspberry Pi is dependent upon an individual's use-case. In general, the Raspberry Pi 2 model B's power consumption can be estimated to be ∼1.2 W (for single-core, idling, with communications, and no peripherals; Eames, 2015).

To evaluate the effectiveness of RaPiER as a geoscience node, we evaluated its suitability and performance. System suitability is a measure of its ability to operate in its intended environment. System performance measures refer to quantifiable parameters associated with its ability to function as intended. Our evaluation of the RaPiER's effectiveness is not intended to be a comprehensive determination of its readiness for field deployment. Rather, IoGT Makers can use our preliminary evaluation to guide their own RaPiER integration and subsequent deployments.

Measures of Suitability

We fabricated two nodes for RaPiER effectiveness evaluation. The nodes' peripheral components (i.e., Ethernet switch and USB flash drive) were varied to determine if it would be necessary to use components with higher operating temperature ranges (Table 2.2). The two nodes were assembled similarly and placed within identical enclosures (Fig. 2.4).

Table 2.2 RaPiER Nodes 1 and 2 Components

	Raspberry Pi SOC*		Ethernet Switch		USB Flash	
RaPiER Node	Make and Model	Operating Temperature Range	Make and Model	Operating Temperature Range	Make and Model	Operating Temperature Range
1	Raspberry Pi 2†	−25°C to +80°C	Brainboxes SW-504	−40°C to +80°C	MIT TECH MLC 64 GB	0°C to +85°C
2	Raspberry Pi 2	−25°C to +80°C	Brainboxes SW-005	0°C to +70°C	PNY Attaché 128 GB	0°C to +60°C

*SOC, system on chip.
†Heat sinks on Raspberry Pi 2 CPU/GPU and Ethernet modules.

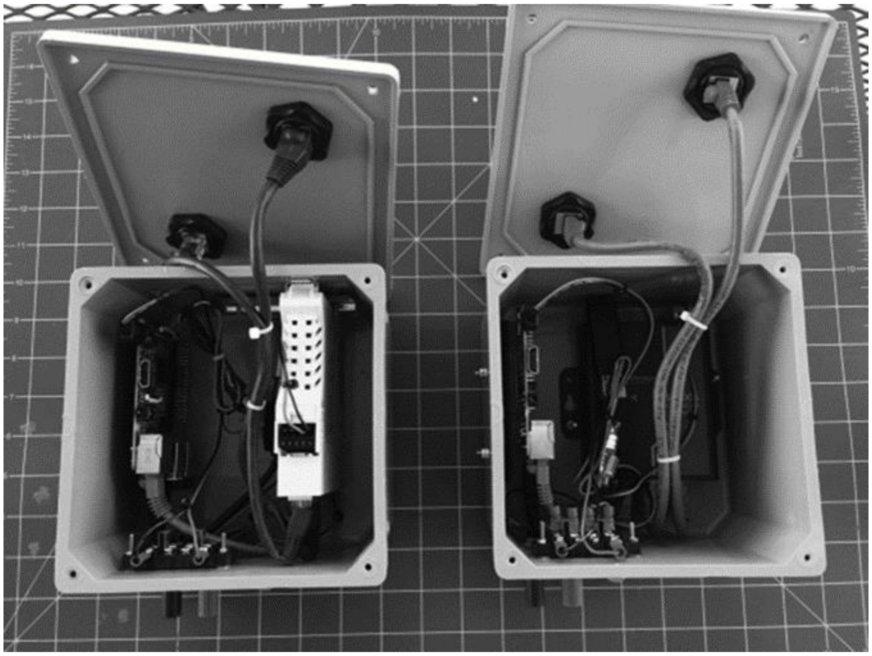


Figure 2.4. RaPiER node interior: (left) node 1 and (right) node 2.

The two RaPiER nodes were emplaced within an unventilated wooded shed for a seven-day suitability test (Fig. 2.5). The nodes were powered up and acquisition was started. Using Munin, we monitored the nodes' Raspberry Pi core temperatures and system uptime during our suitability test. Over the course of our suitability test, outdoor temperatures ranged from 22.0°C to 37.2°C. Raspberry Pi core temperatures ranged from 41.9°C to 63.2°C and 43.3°C to 64.3°C for RaPiER nodes 1 (Fig. 2.6) and 2, respectively. Typically, a Raspberry Pi's core temperature, within a climate-controlled room (i.e., 25°C), is approximately 40°C.

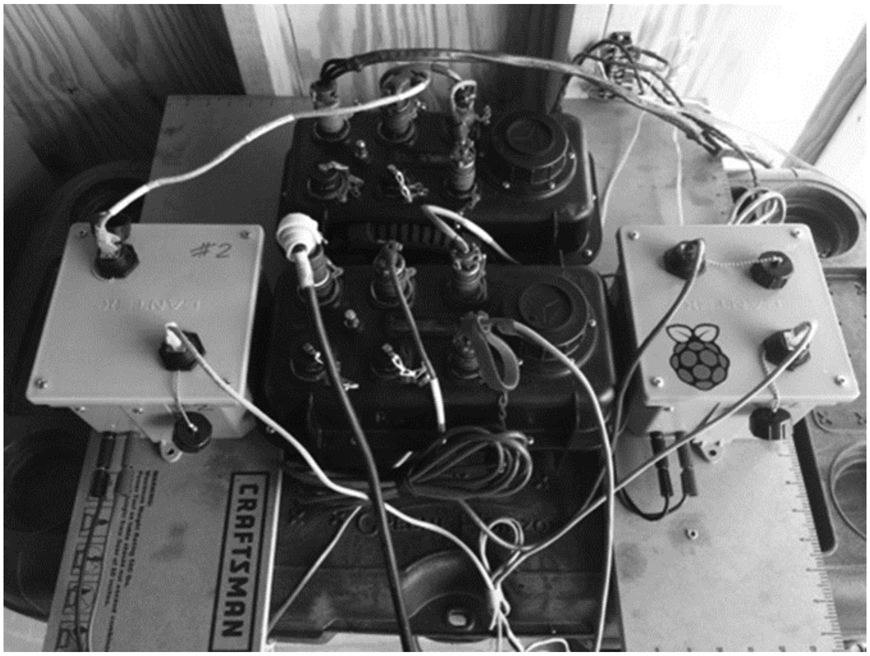


Figure 2.5. RaPiER nodes 1 and 2 (gray square enclosures) located within wooden shed for seven-day suitability test. REF TEK 130-01s (black rectangular enclosures) are located between the RaPiER nodes.

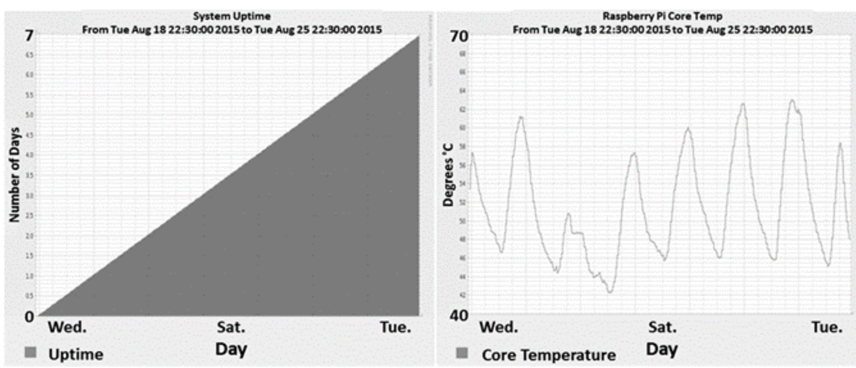


Figure 2.6. Seven-day RaPiER node 1 suitability test results. (left) System uptime and (right) Raspberry Pi core temperature plots were generated using a networked performance monitoring tool (i.e., Munin).

We observed no hardware-related failures during our seven-day suitability test. RaPiER node 1's Raspberry Pi core temperature was ~1.3°C lower than node 2's core temperature. The difference in node core temperatures is most likely due to the heat sinks placed on node 1's Raspberry Pi's CPU/graphics processing unit (GPU) and Ethernet modules.

We measured a RaPiER node's power consumption given our specific use-case. This included the DC-to-DC converter, Ethernet switch, Raspberry Pi 2 model B, REF TEK 130-01, and USB flash drive. Power consumption was also measured while the node was processing 130-01 data.

We estimate the RaPiER node's power consumption to be 2.36–3.10 W higher than the REF TEK 130-01's (Table 2.3). This 130%–170% increase in power consumption is appreciable; however, a single 20 W solar panel (at eight hours of sun, with a worst weather multiple of 0.65) could be used to offset the node's power consumption for our specific use-case. Nonetheless, we recommend IoGT Makers carefully examine their field station's power budget prior to node deployment.

Table 2.3 RaPiER Node Power Consumption Estimates

RaPiER Node	Power Consumption Estimates by Component (Watts)			Total Power Consumption (Watts)
	REF TEK 130-01*	Raspberry Pi SOC†	Ethernet Switch	
1	1.82	1.7	0.66	4.18
2	1.82	1.7	0.65	4.17
2 (processing data)	1.82	2.47‡	0.65	4.94

*Three channels, passive sensor, Global Positioning System (GPS) active, with communications.
†Single core system on chip (SOC), RTPD active, with communications.
‡Estimated power consumption, processing data.

To quantify a RaPiER node's impact on network resources, we collected 12 hours of network traffic statistics using Munin. These statistics included active local-area-network (LAN) traffic between the Raspberry Pi 2 model B and the REF TEK 130-01 and idle wide-area-network (WAN) traffic between the Raspberry Pi 2 model B and the Internet. Average LAN traffic was measured to be 2880 bits per second (bps) inbound and 179 bps outbound (Fig. 2.7). WAN traffic averaged ~13 bps inbound and outbound (Fig. 2.7).

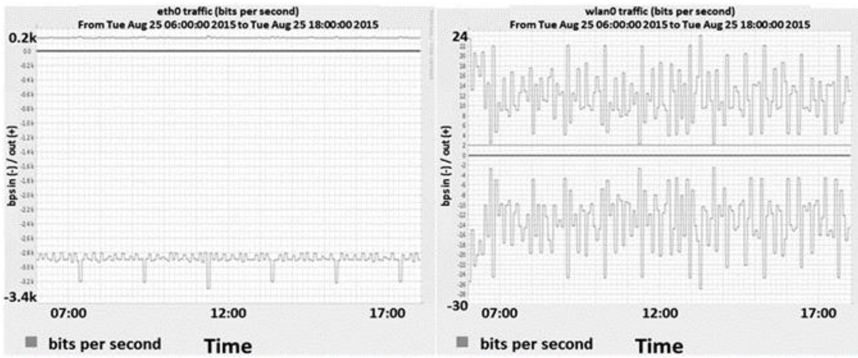


Figure 2.7. Twelve hour RaPiER node impact on network resources test results. (left) Local-area-network traffic and (right) wide-area-network traffic plots were generated using a networked performance monitoring tool (i.e., Munin; bps, bits per second).

Observed RaPiER node LAN traffic roughly correlates in size with 12 hours of RTPD archive data. Node idle WAN traffic represents a modest use of network resources. A year of idle WAN traffic would be ~100 MB. The RaPiER node does not appear to negatively impact LAN or WAN resources; however, IoGT Makers should be aware that interactions with the node (e.g., data retrieval, remote access, etc.) will utilize network resources.

Measures of Performance

REF TEK's RTI software is a well-established solution for the acquisition of near-real-time data from REF TEK 130-01 field stations to data centers. However, the ARM architecture version of the RTI software is untested. As such, we investigated the reliability of the ARM architecture RTPD module and the quality of archive data.

To evaluate the reliability of the ARM architecture RTPD module, we monitored the RaPiER nodes' RTPD archives and logs during our seven-day suitability test. We verified that the nodes' archives contained the expected number of data files and that these files were of the correct duration. We reviewed the nodes' daily RTPD logs for errors. We observed no RTPD module reliability issues during our suitability test.

We assessed RaPiER's RTPD archive data quality by comparing it with the REF TEK 130-01 CF card data. We selected 36 nonconsecutive two-hour files, collected over six days, from the RTPD archive and the CF card. We calculated the correlation coefficients for each channel of the 36 pairs of data files. The correlation coefficients indicated a perfect match between the RTPD archive and CF card data.

As a secondary measure of RTPD archive data quality, we ran a data-processing workflow (Fig. 2.8) on a RaPiER node for approximately eight days (i.e., nodal processing). During these eight days, the RaPiER node acquired data from a Nanometrics Trillium Compact (120s) broadband seismometer (Nanometrics, n.d.). The workflow automatically generated hourly waveform plots (Fig. 2.9). A second RaPiER data-processing workflow was run to generate a probabilistic power spectral density (PPSD) plot (Fig. 2.10) (McNamara, 2004). Waveform and PPSD plots were automatically uploaded to a cloud storage service for review. The ObsPy Python toolbox (Beyreuther et al., 2010; see also Data and Resources) was used to generate waveform and PPSD plots.

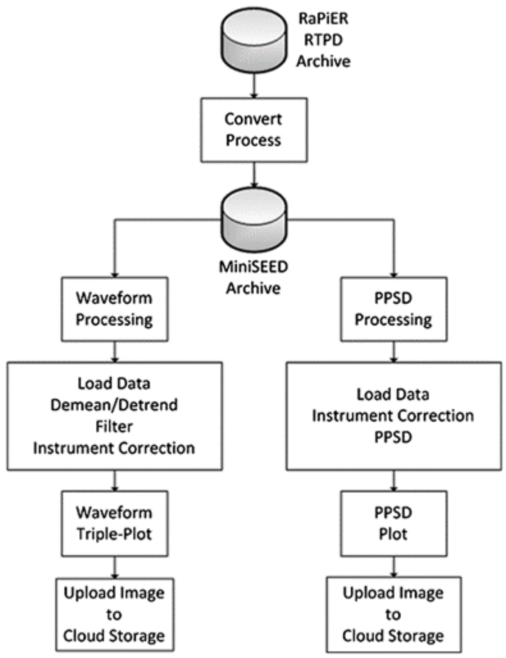


Figure 2.8. RaPiER nodal processing workflows for data-quality performance measure evaluation. Workflows automatically generate and upload time and frequency domain plots for qualitative review.

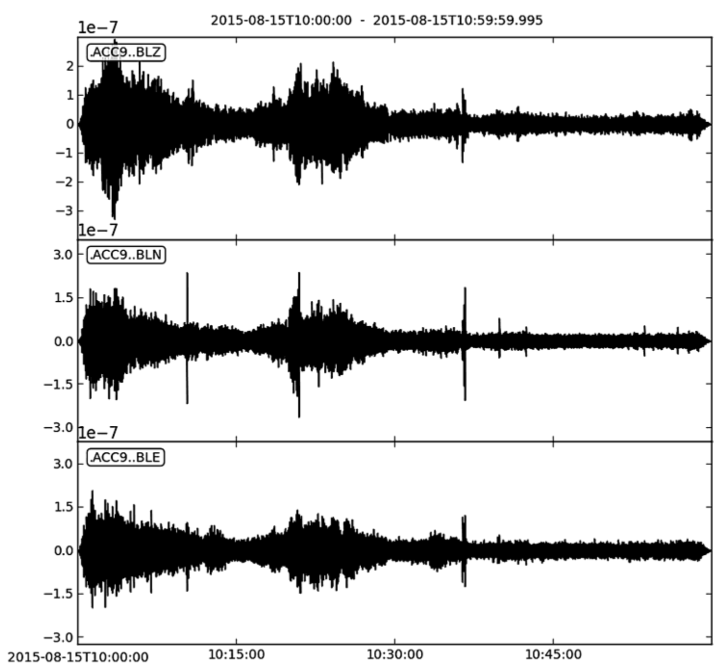


Figure 2.9. Sample of one-hour waveform triple plot (RaPiER nodal processing). Background broadband seismometer data are shown as a velocity (m/s) versus time (s) plot. The instrument response was removed, data were de-meaned and detrended, and a low-pass filter applied (10 Hz).

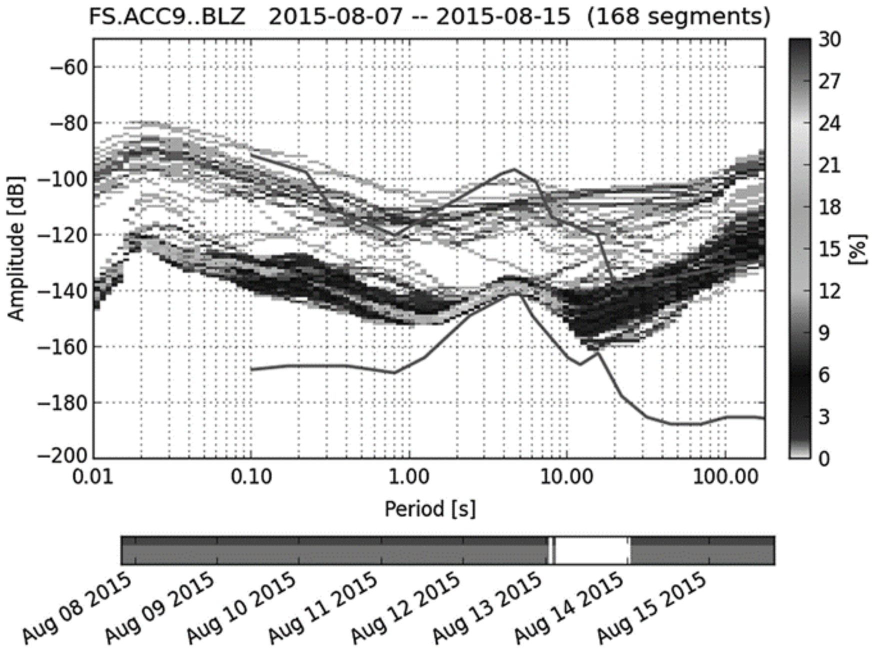


Figure 2.10. Sample of 168-segment (i.e., 168 hour) probabilistic power spectral density (PPSD) plot. The broadband seismometer PPSD plot shows amplitude (dB, noise) versus period (s).

Waveform plots were generated using a routine similar to the ObsPy Development Team (n.d.a) "obspy.signal – Signal Processing Routines for ObsPy." Waveform signal processing consisted of (1) loading a one-hour miniSEED data file, (2) removing the instrument response, (3) de-meaning and detrending the data, (4) applying a filter (i.e., low pass, 10 Hz, two corners, and zero phase), and (5) generating a triple plot (i.e., Z, N, and E) of velocity (in meters per second) versus time (in seconds).

The PPSD plots were generated using the ObsPy Tutorial "Visualizing Probabilistic Power Spectral Densities" (ObsPy Development Team, n.d.b) as a guide. This routine first establishes a path to the data (i.e., 168 one-hour miniSEED files). The routine then gets poles and zeros information from a dataless SEED file. Next, the routine initializes a new PPSD instance. Additional one-hour data files are then added to the PPSD estimate. All PPSD processes (e.g., de-meaning and detrending, instrument response, etc.) are done internally. When complete, this routine generates a graphical representation of the PPSD. This process was repeated for each channel (i.e., Z, N, and E). Refer to McNamara (2004) or the ObsPy Tutorial “Visualizing Probabilistic Power Spectral Densities” for additional information regarding internal PPSD processes or visualization.

Our experience with RaPiER nodal processing was resoundingly positive. Hourly waveform plots were typically processed and uploaded in less than two minutes and a 168-segment (i.e., 168 hour) PPSD plot was generated in approximately 50 min. We reviewed the 168 waveform plots and the 168-segment PPSD plots to identify anomalous RTPD archive data. No obviously atypical RTPD archive data were observed in the time or frequency domain.

Discussion and Conclusions

RaPiER’s relatively low power consumption, modest use of network resources, and apparent thermal robustness make it a viable option for remote field station deployments. Given our specific field station conditions, we determined that it was not necessary to use high-temperature node components (i.e., Ethernet switch and USB flash drive). However, we encourage IoGT Makers to consider their field station conditions and use-case prior to RaPiER system integration and node deployment. Although we did not fully test REF TEK’s ARM architecture RTI software archive utilities or the RTPD module’s functionality, our preliminary performance evaluation indicated that the software is reliable and produces quality archive data.

Given RaPiER’s ease of integration and effectiveness as a geoscience node, we consider RaPiER a practical addition to a geoscientist’s equipment inventory. Aside from

REF TEK products, our RaPiER node cost approximately $220. As an economically attractive and inherently educationally aligned solution, we anticipate RaPiER will be well received as a platform for geoscience graduate and undergraduate students to develop basic electronics, Linux, networking, and programming skills. As Hall and Bianco (2012) describe, students can develop the underrated skill of tinkering while simultaneously acquiring quality data for serious scientific study.

Geoscientists engaged in critical deployments may be reluctant to disturb their existing nodes to implement nonessential enhancements such as RaPiER. To alleviate possible reticence regarding RaPiER, geoscientists should consider that RaPiER does not interfere with the REF TEK 130-01's ability to record data to a CF card locally or to transmit data to a remote RTPD server. This is particularly true if the 130-01 is flashed with firmware 3.7.2 or later and is configured to transmit its data stream to a local RTPD server and remote RTPD server, independently.

The fact that RaPiER expands local data storage capabilities beyond the REF TEK 130-01's CF cards is another reason to consider its implementation. The RaPiER's owner can configure the node to store RTPD archive data to any type or size of storage device that is compatible with Raspberry Pi. By using RaPiER, geoscientists could increase their deployment durations and lower storage media costs. A 128 GB USB flash drive, micro-SD card, and CF card cost approximately $35, $100, and $180 respectively. The only CF cards for which REF TEK guarantees compatibility with the 130-01 are REF TEK CF cards. A REF TEK 4 GB CF card cost approximately $100.

RaPiER implementation provides automatic failover for 130-01s configured to continuously transmit data to a remote RTPD server; that is, should remote network communication go down, RaPiER's RTPD archive will continue to build regardless of remote network status. Once a connection with the network is reestablished, missing data can be retrieved using a variety of methods (e.g., FTP, SSH, etc.). Geoscientists are no longer required to retrieve 130-01 CF cards to reacquire missing data from remote RTPD archives.

Of all the reasons to consider RaPiER integration and implementation, perhaps the most compelling is nodal processing. After all, for RaPiER to be considered an IoT device it must be smart. Our foray into nodal processing was to meet our immediate need to efficiently assess RTPD archive data quality. Geoscientific (i.e., seismic) application areas that may benefit from nodal processing (e.g., automated, near-real time, etc.) include earthquake (e.g., aftershock studies) and engineering (e.g., site characterization) seismology.

Considering the Raspberry Pi's capabilities and the wealth of information available via its ecosystem, we only scratched the surface of RaPiER nodal processing. As geoscientists' bridge between familiar and unfamiliar technologies, RaPiER allows for the realization of Ashton's vision of IoT devices sensing and making sense of the world for themselves. However, in our case as geoscientists, our IoGT device senses and makes sense of the physical mechanisms of the world.

Data and Resources

The data acquired to evaluate the effectiveness of the RaPiER as a geoscience node (i.e., suitability and performance) were acquired as benchtop data and were not intended for subsequent seismic processing.

We recommend prospective IoT Makers review information available from http://www.adafruit.com (last accessed February 2016) or http://www.makezine.com (last accessed February 2016) for examples of IoT projects. For information specific to the Arduino microcontroller, Gumstix computer on module, or the Raspberry Pi system on chip, refer to http://www.arduino.cc (last accessed February 2016), https://www.gumstix.com (last accessed February 2016), and https://www.raspberrypi.org (last accessed February 2016), respectively. The Raspberry Pi Foundation (n.d.) provides official documentation that details initial setup, operating system installation (https://www.raspberrypi.org/downloads/raspbian/, last accessed February 2016), usage, and configuration of the Raspberry Pi.

For a listing of geoscience-related REF TEK instrument deployments, refer to the Trimble Navigation Limited website (http://www.trimble.com/Infrastructure/Digital-Telemetry-Networks.aspx, last accessed February 2016). The Incorporated Research Institutions for Seismology Portable Array Seismic Studies of the Continental Lithosphere Instrument Center and EarthScope USArray Array Operations Facility inventory is available from https://www.passcal.nmt.edu/content/general-information/equipment-inventory (last accessed February 2016).

Additional information regarding the ARM processors can be obtained from http://www.arm.com/products/processors (last accessed February 2016). Ⓔ The ARM architecture REF TEK Interface (RTI) software and Raspberry Pi and RTI installation and configuration scripts are available in the electronic supplement. Refer to https://packages.debian.org/stable/mail/ssmtp (last accessed February 2016) and https://security.appspot.com/vsftpd.html (last accessed February 2016) for additional information regarding SSMTP and VSFTPD. Additional information regarding Munin can be obtained from http://munin-monitoring.org (last accessed February 2016). The Apache Software Foundation HTTP Server Project (http://httpd.apache.org/, last accessed February 2016) provides Apache webserver set up information. Installation instructions for the ObsPy Python toolbox are available at http://github.com/obspy/obspy/wiki (last accessed February 2016).

Acknowledgments

We want to thank Tim Meredith (Baylor University) and Travis Rennemann (Vencore) for their support of the RaPiER project. We would also like to thank REF TEK support, specifically Ian Billings and Sergio Valenzuela. Last, we want to thank the Raspberry Pi Foundation and the Raspberry Pi’s ecosystem’s contributors.

References

Ashton, K. (2009). That “Internet of Things” thing, *RFID J.*, available from http://www.rfidjournal.com/articles/view?4986 (last accessed April 2015).

Baldwin, H. (2014). A match made somewhere: Big Data and the Internet of Things, *Forbes*, available from http://www.forbes.com/sites/howardbaldwin/2014/11/24/a-match-made-somewhere-big-data-and-the-internet-of-things/ (last accessed June 2015).

Bertolucci, J. (2014). Internet of things “overhyped,” say IT execs, *InformationWeek*, available from http://www.informationweek.com/big-data/big-data-analytics/internet-of-things-overhyped-say-it-execs/d/d-id/1317518 (last accessed July 2015).

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A Python Toolbox for Seismology, *Seismol. Res. Lett.* **81,** no. 3, 530–533, doi:10.1785/gssrl.81.3.530.

Bourdeau, J. (2012). Setup VSFTPD with custom multiple directories and (virtual) users accounts on Ubuntu (no database required), *Sigerr*., available from http://www.sigerr.org/linux/setup-vsftpd-custom-multiple-directories-users-accounts-ubuntu-step-by-step/ (last accessed July 2015).

Challen, G., and M. Welsh (2010). Volcano monitoring: Addressing data quality through iterative deployment, in *Wireless Sensor Networks*, E. Gaura (Editor), Springer, New York, New York, 71–113.

Cukier, K. N., and V. Mayer-Schoenberger (2013). The rise of big data, *Foreign Aff.*, available from https://www.foreignaffairs.com/articles/2013-04-03/rise-big-data (last accessed June 2015).

Eames, A. (2015). Raspberry Pi2: Power and performance measurement, *RasPi.TV*, available from http://raspi.tv/2015/raspberry-pi2-power-and-performance-measurement (last accessed August 2015).

Gabbai, A. (2015). Kevin Ashton describes, *Smithson. Mag.*, available from http://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/ (last accessed April 2015).

Gibbs, M. (2015). Ten reasons why the new Raspberry Pi 2 model B SBC is a killer product, *Networkworld*, available from http://www.networkworld.com/article/2887224/tech-primers/10-reasons-why-the-raspberry-pi-2-model-b-is-a-killer-product.html (last accessed July 2015).

Greengard, S. (2015). *The Internet of Things*, MIT Press Essential Knowledge Series, MIT Press, Cambridge, Massachusetts.

Hall, M., and E. Bianco (2012). *52 Tthings You Should Know about Geophysics*, Agile Libre, Mahone Bay, Nova Scotia.

Langner, C. (2015). Monitoring Raspberry Pi temperatures, *Raspberry Pi Geek*, no. 10, 16–19.

LeHong, H., and J. Fenn Gartner (2014). Digital business technologies dominate gartner 2014 emerging technologies hype cycle, *Forbes*, available from http://www.forbes.com/sites/gartnergroup/2014/09/17/digital-business-technologies-dominate-gartner-2014-emerging-technologies-hype-cycle/ (last accessed June 2015).

McEwen, A., and H. Cassimally (2014). *Designing the Internet of Things*, Wiley, Chichester, United Kingdom.

McNamara, D. E. (2004). Ambient noise levels in the continental United States, *Bull. Seismol. Soc. Am.* **94,** no. 4, 1517–1527, doi:10.1785/012003001.

Miller, M. (2015). *The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities are Changing the World*, QUE, Indianapolis, Indiana.

Nanometrics (n.d.). Trillium Compact Broadband Seismometer | Nanometrics, available from http://www.nanometrics.ca/seismology/products/trillium-compact (last accessed January 2016).

Obspy Development Team (n.d.a). Visualizing Probabilistic Power Spectral Densities—ObsPy Documentation (0.10.2), available from http://docs.obspy.org/tutorial/code_snippets/probabilistic_power_spectral_density.html (last accessed January 2016).

Obspy Development Team (n.d.b.). Obspy.signal-Signal Processing Routines for ObsPy—ObsPy Documentation (0.10.2), available from http://docs.obspy.org/packages/obspy.signal.html#module-obspy.signal (last accessed January 2016b).

Orsini, L. (2014). How to host a website with Raspberry Pi, *Readwrite*, available from http://readwrite.com/2014/06/27/raspberry-pi-web-server-website-hosting (last accessed July 2015).

Press, G. (2014). It's official: The internet of things takes over big data as the most hyped technology, *Forbes*, available from http://www.forbes.com/sites/gilpress/2014/08/18/its-official-the-internet-of-things-takes-over-big-data-as-the-most-hyped-technology/ (last accessed July 2015).

Raspberry Pi Foundation (n.d.). *Raspberry Pi NOOBS Setup, Get. Started NOOBS*, available from https://www.raspberrypi.org/help/noobs-setup/ (last accessed January 2016).

Schutt, R., and C. O'Neil (2013). *Doing Data Science*, First Ed., O'Reilly Media, Beijing, China.

Technopedia (n.d.). What is System Integration (SI)? - Definition from Techopedia, available from https://www.techopedia.com/definition/9614/system-integration-si (last accessed January 2016).

TNET (n.d.). TNET Raspberry Pi Site: SMTP Mail Setup, available from https://rpi.tnet.com/project/faqs/smtp (last accessed January 2016).

Voight, J. (2014). Which big brands are courting the maker movement, and why, *AdWeek*, available from http://www.adweek.com/news/advertising-branding/which-big-brands-are-courting-maker-movement-and-why-156315 (last accessed July 2015).

Whitmore, A., A. Agarwal, and L. D. Xu (2014). The internet of things: A survey of topics and trends, *Inf. Syst. Front.*, **17**, no. 2, 261–274, doi:10.1007/s10796-014-9489-2.

CHAPTER THREE

A Novel, Near Real Time Approach to Seismic Exploration and Monitoring via Ambient Seismic Noise Interferometry

Abstract

A new, cost effective and non-invasive exploration method using ambient seismic noise was developed and tested with promising results. Our general objectives were to build and test a new-generation seismic system capable of acquiring, transmitting, and processing seismic data in near real-time and to test the new technology in a geothermal field setting. The intent of the latter objective was to investigate opportunities for adapting survey acquisition parameters provided by near real-time data processing. Development proceeded in two phases: we first designed, built, and tested a 20-node array at the Soda Lake Geothermal Field, Nevada, then we scaled up to 144 nodes and greater aperture for a test at the San Emidio Geothermal Field, Nevada. We demonstrated the larger array was able to perform its data acquisition, transmission, and processing functions successfully; the advantages of real-time, “in the field” processing, were realized. These advantages include continuous assessment of results and opportunities to re-deploy stations in more optimal configurations depending upon characteristics of the ambient noise field and options to leave the array in place longer (or shorter) than originally planned with high levels of confidence survey objectives have been achieved, before the decision to demobilize the array is made.

Introduction

As of the early 2000s, engineering and scientific interest in wireless sensor networks (WSNs) was well established (Ilyas & Mahgoub, 2006). The availability of inexpensive and low size, weight, and power (SWaP) computers, sensors, and wireless communications allowed for the development of engineering and scientific related WSNs intended to: 1) acquire and process data, 2) detect and characterize activities or events of interest, and 3) communicate "sensed information" back to users (Ilyas & Mahgoub, 2006). Despite widespread interest, the development and adoption of WSNs remained technologically challenging due to the inherent complexities (e.g., bandwidth, power, etc.) of transforming sensor nodes into a resource-efficient network (Ilyas & Mahgoub, 2006).

Application areas for WSNs include, but are not limited to, civil engineering, defense, environmental monitoring, etc. (Ilyas & Mahgoub, 2006). However, due to the technical challenges of wireless sensor network (WSN) development, WSN developers often collaborate with domain experts (e.g., medical professionals, scientists, etc.). For instance, Challen and Welsh (2010) spent five years working with geoscientists to develop a variety of WSNs for volcanic monitoring. These volcanic monitoring WSNs provided geoscientists better data (i.e., quantity and spatial-distributed) by deploying many inexpensive and low SWaP nodes around an active volcano; however, the solution remained constrained by finite resources (e.g., bandwidth, power, etc.) (Challen & Welsh, 2010).

Challen and Welsh (2010) were forced to make tradeoffs between data quality, deployment duration, cost, etc. These tradeoffs may ultimately require that geoscientists accept that the benefits of more data offset the drawbacks of lower-quality data. To minimize tradeoffs, Challen and Welsh (2010) suggested the development of WSNs within the context of a specific scientific objective where the minimum quality or quantity of data required has been previously identified. Unfortunately, WSNs designed for specific scientific objectives may prove too application specific for use in other geoscientific endeavors without additional WSN developer support. Also, data of minimum quality or quantity may limit geoscientists' ability to understand the processes underlying an event of interest.

The utilization of WSNs to acquire, disseminate, and process geoscientific data in the field or other remote environments (sometimes referred to as "the edge") is challenging. Challenges are compounded as the number of WSN nodes increases or the nodes are distributed across a larger geographic area. Fortunately, within the past eight years there were major advances in the availability and capability of embedded systems (i.e., Raspberry Pi[1]) and distributed databases (i.e., Apache Cassandra[2]).

The proliferation of Internet of Things (IoT) devices has resulted in a need for edge computing: computing facilities that allow for the processing and analysis of data near the point of acquisition (Hamilton, 2018). Similarly, fog computing involves the processing and analysis of data across one or more nodes in a network (Hamilton, 2018). Confronted with identical real-world logistical constraints as geoscientific WSNs (e.g., bandwidth, outages, etc.), mainstream IoT-based sensor network edge computing is advancing rapidly in a direction that benefits geoscientific applications.

[1] https://www.raspberrypi.org/

[2] https://cassandra.apache.org/

In this paper, we document attempts to develop and implement a solution allowing for the automated acquisition, transmission, storage, and processing of seismic data at "the edge" and in near real-time. We developed and/or implemented solutions in the following areas: 1) embedded systems, 2) a distributed database, 3) solution architecture, and 4) telemetry infrastructure. We conducted four test and evaluation (T&E) events of varying durations: one each at Eastland Lakes, Texas in May 2017, Soda Lake Geothermal Field, Nevada in June 2017, Baylor Research and Innovation Collaborative (BRIC), Texas in July 2018, and San Emidio Geothermal Field, Nevada in May 2019. Because of the complexity of WSNs (Ilyas & Mahgoub, 2006) and the relative newness of edge computing (Hamilton, 2018), we leveraged experience gained during our four T&E events to develop, implement, and refine solutions that allowed us to increase the number of seismic stations while maintaining the ability to acquire, transmit, store, and process seismic data in the field and in near real-time. Additional information regarding embedded systems, distributed database, solution architecture, and telemetry infrastructure is provided below.

Embedded Systems

Embedded systems include a wide range of products and an even wider range of applications ranging from hobbyist to industrial. Embedded systems include microcontrollers, embedded computers, system-on-chip, computer-on-module, system-on-module, etc. Typically, embedded systems are low SWaP, intended for specific applications, and have modest capabilities when compared with desktop or laptop computers.

The Raspberry Pi is among the most popular processors to be incorporated into embedded systems. As of December 2018, the Raspberry Pi was the world's third best-selling general purpose computer (Heath, 2019). As a system originally intended to teach children computer science, the Raspberry Pi is inherently easy-to-use and inexpensive (Heath, 2019). However, having developed an immense community of users, beyond children, use of the Raspberry Pi for industrial and scientific endeavors are commonplace.

To provide geoscientists a "no engineer needed" solution for enhanced sensor node development and maintain the high-quality data geoscientists are accustomed to, Sepulveda and Pulliam (2016) integrated a Raspberry Pi 2 and the REF TEK 130-01 to develop the "Raspberry Pi Enhanced REF TEK" (RaPiER). Although the RaPiER represented a capable single-node solution for edge-based geoscientific applications (Sepulveda & Pulliam, 2016), our current efforts required a multi-node solution. Our desire to acquire, transmit, store, and process seismic data in a manner that minimizes software development (i.e., non-geoscientific) requires that we carefully select an embedded system that allows for "articulated" edge storage. Similar to IoT edge storage (Koegler, 2019), our solution should allow us the ability to combine data from multiple devices and perform seismic processing in a field setting.

Distributed Database

A distributed computing system, as defined by (Özsu & Valduriez, 2011), is a collection of autonomous computing devices, interconnected by a computer network, that allows for the cooperative completion of tasks. Although the complexities of the exact nature of "distribution" (e.g., data, processing logic, etc.) are beyond the scope of this discussion, the fundamental need to distribute processing flows from the need to improve computation speeds, the global nature of modern enterprises, and the need to accommodate large volumes of data (Özsu & Valduriez, 2011).

Within the context of an enterprise, the physical location of data is often an afterthought. With cloud, on-premises, and hybrid (i.e., both cloud and on-premises) database solutions available, the actual physical location of data can become unimportant if high-bandwidth, high-availability, and fault tolerant network connectivity is available. The intricacies of distributed processing are similarly obscured by the easy-to-use, enterprise-grade, database management system tools that business users have come to expect. Microsoft Access, MySQL, and Oracle are examples of popular relational database management systems (RDBMS) (Carpenter & Hewitt, 2016).

As data volumes increase, RDBMS administrators have two available scaling options: 1) the distribution of data across more machines (i.e., horizontal scaling) or 2) increasing the system performance of the existing machine (i.e., vertical scaling) (Carpenter & Hewitt, 2016). Vertical scaling, although simple to implement, may not be the most effective strategy. Horizontal scaling uses relatively inexpensive commodity hardware to distribute the database across multiple systems, thus reducing the overall workload of individual systems.

Unfortunately, a distributed RDBMS results in distributed transactions. This requires the implementation of a two-phase commit to prevent new transactions from executing until the prior transaction is complete and a commit response has been returned to the transaction manager (Carpenter & Hewitt, 2016). As the number of transactions (i.e., data velocity) and duration of transaction processing time (i.e., data volume) increase, the RDBMS will likely encounter performance issues resulting from the way RDBMSs inherently operate (Carpenter & Hewitt, 2016; Chen & Zhang, 2014).

Given our desire to ingest hundreds of millions of samples per day into a database running on a modest number of nodes, spread across thousands of meters, we required a distributed database that was well-suited to timeseries data, built for high availability, linear (i.e., horizontal) scalability, and decentralization. Circa 2016, we performed a literature review to identify RDBMS alternatives that were ideally suited for mid-to-large scale IoT-based sensor network applications (Abramova et al., 2014; Confais et al., 2016; Duarte & Bernardino, 2016; Le et al., 2014; van der Veen et al., 2012). Given our requirements, a “Not only SQL” (NoSQL) database, specifically Apache Cassandra, emerged as our database of choice.

Initially created by Facebook to solve their Inbox Search problem, Cassandra leveraged Amazon’s Dynamo and Google’s Bigtable to meet challenging write-heavy (i.e., billions per day), geographical distribution, reliability, and scalability requirements (Lakshman & Malik, 2010). Cassandra, accepted as an Apache Software Foundation top level project in February 2010, is an open source, distributed, decentralized, hybrid, operationally simple, elastically scalable, highly available, fault-tolerant, and tuneably consistent database (Carpenter & Hewitt, 2016; Ploetz et al., 2018). While Cassandra was an established NoSQL solution in 2016, to the best of our knowledge, no one had attempted to deploy Cassandra on an embedded system in support of a scientific endeavor. Although feasible, the extremely modest performance (in terms of CPU speed, RAM complement, etc.) of embedded systems and constraints of real-world ad hoc wireless networks would require a carefully considered approach to meet our goal. Our design evolved during the course of a three-year development program, particularly in response to lessons learned during four major T&E events. Important and instructive details of that evolution are documented in the following sections.

Solution Architecture

Eastland Lakes, Texas, May 2017 and Soda Lake Geothermal Field, Nevada, June 2017

In May 2017 and June 2017, we deployed 20 RaPiER nodes configured so that each Raspberry Pi 3 was responsible for: 1) the acquisition of data from one REF TEK 130-01, 2) conversion of said data from REF TEK native formatted (RTP) file and insertion into Cassandra, 3) running Cassandra as a Transactional Data Center (DC) participating in a two DC (i.e., three and 20 node) cluster (Fig. 3.1). The Transactional DC, composed of 20 RaPiER nodes, was responsible for the insertion of seismic data into the cluster. The Query DC was responsible for handling queries for subsequent seismic processing. Unfortunately, the modest resources (i.e., processor and memory) of the Raspberry Pi 3 resulted in frequent RaPiER downtime (e.g., hangs, reboots, etc.).

In order to improve reliability, we decided to offload the Raspberry Pi 3's Cassandra-related workload by replacing the Raspberry Pi 3 with an Asus Tinker Board[3]. The Tinker is like the Raspberry Pi 3; however, it has an additional gigabyte of RAM (i.e., two gigabytes). The additional gigabyte of RAM significantly improved Cassandra's performance and reliability.

---

[3] https://www.asus.com/us/Single-Board-Computer/Tinker-Board/

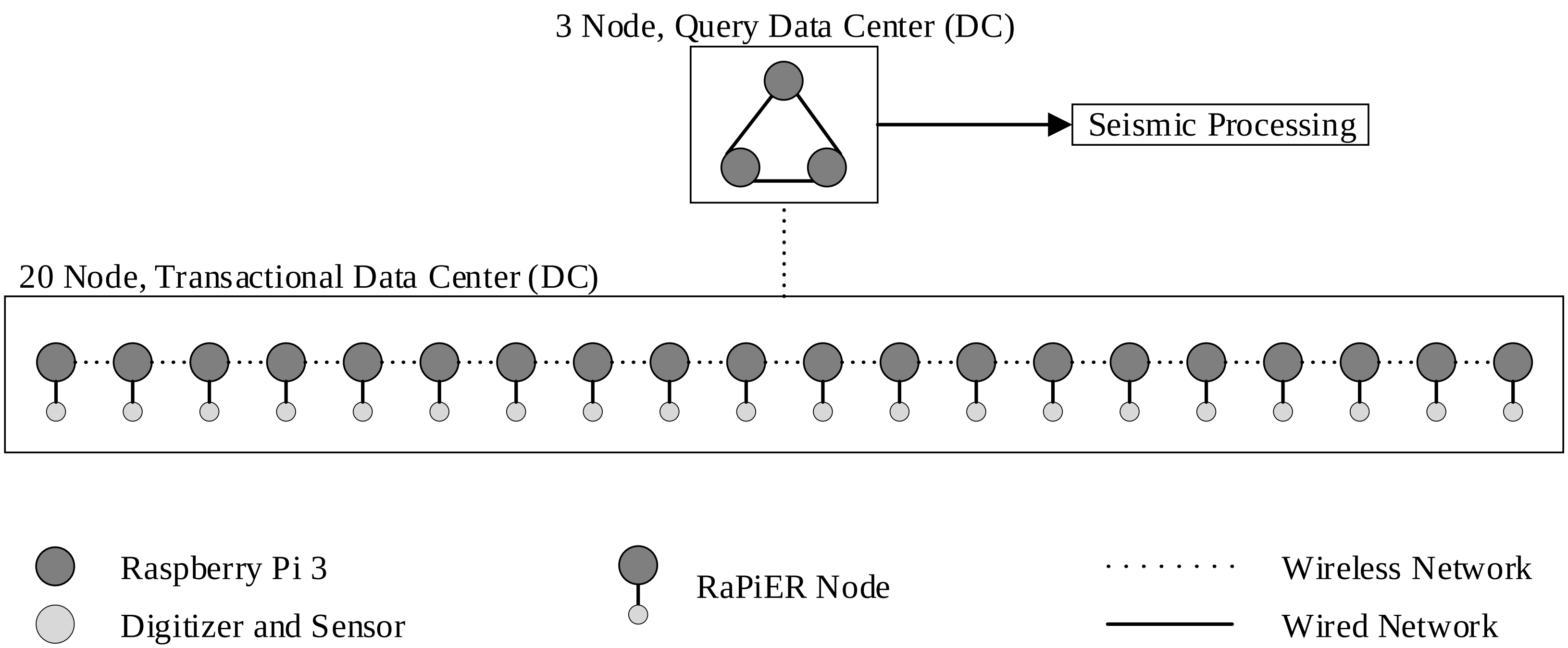


Figure 3.1. Eastland Lakes and Soda Lake T&E event architecture.

In addition to downtime related to the embedded system, we experienced downtime related to network compartmentalization. One section of the ad hoc wireless network we established to allow for data replication, gossip, etc., between node-to-node and DC-to-DC would disconnect from another section of the network. Eventually, nodes were deprecated due to unanswered topological gossip state updates. This resulted in a loss of Cassandra data insertion and replication.

Baylor Research and Innovation Collaborative, Texas, July 2018

Given the lightweight workload now required of the Raspberry Pi 3 (i.e., without Cassandra) and our desire to lower cost, we decided that it was feasible to increase the number of REF TEK 130-01s transmitting data to an individual Raspberry Pi 3 from one to three. In July 2018, we deployed a combination of Raspberry Pi 3s and Tinkers in a tiered configuration.

Our shift to a tiered configuration was motivated by a desire to: 1) transition to a hub-and-spoke topology (Fig. 3.2) that would allow us to implement a replication solution that tolerates inevitable network outages (i.e., real-world conditions), and 2) scale up the number of seismic stations to more than one hundred. A schematic of the configuration deployed at Baylor's BRIC complex is shown in Fig. 3.3. In order to fully realize the benefits of a tiered configuration, we transitioned from Cassandra (i.e., Community Edition) to DataStax Enterprise[4] (DSE). DSE is an enterprise-grade version of Cassandra that provides "commercial confidence" and extra capabilities such as automatic management services, advanced security, and advanced functionality.

[4] https://www.datastax.com/products/datastax-enterprise

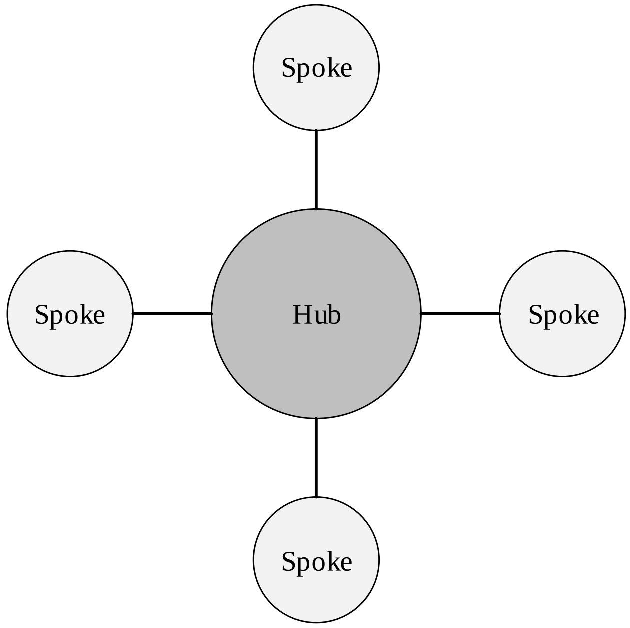


Figure 3.2. Example of hub-and-spoke topology.

Our primary interest in advanced functionality was DSE Advanced Replication[5], which supports the configurable replication of data from source to destination clusters in a manner that tolerates intermittent loss of connectivity. Using DSE Advanced Replication we transitioned from a single cluster with two DCs to a direct cluster-to-cluster configuration (i.e., Transactional Cluster and Query Cluster) that was inherently tolerant of network outages (i.e., backhaul telemetry).

During the BRIC T&E event, we observed throughput issues that required us to stagger the insertion of data in order to minimize potential data loss; however, this introduced a data replication lag (i.e., approximately 4 hours). Although the tiered configuration and DSE Advanced Replication improved performance and reliability, further improvements to telemetry infrastructure would be necessary and we also

[5] https://docs.datastax.com/en/dse/5.1/dse-dev/datastax_enterprise/advReplication/advRepTOC.html

recognized the need to scale down from four Raspberry Pi 3s per Tinker to three Raspberry Pi 3s per Tinker.

San Emidio Geothermal Field, Nevada, May 2019

The San Emidio T&E event would ideally consist of 144 seismic stations, 48 RaPiER nodes, and 32 Tinkers (i.e., Cassandra nodes). Given the scale of this T&E event, it was necessary to introduce new terminology. A group of nine REF TEK 130-01s, three Raspberry Pi 3s, and one Tinker was labeled a “flight.” Four flights became known as a “squadron.” Each squadron (i.e., Transactional Cluster) would replicate data to “headquarters” (i.e., Query Cluster) via DSE Advanced Replication. A schematic of the configuration deployed at San Emidio T&E event is shown in Fig. 3.4.

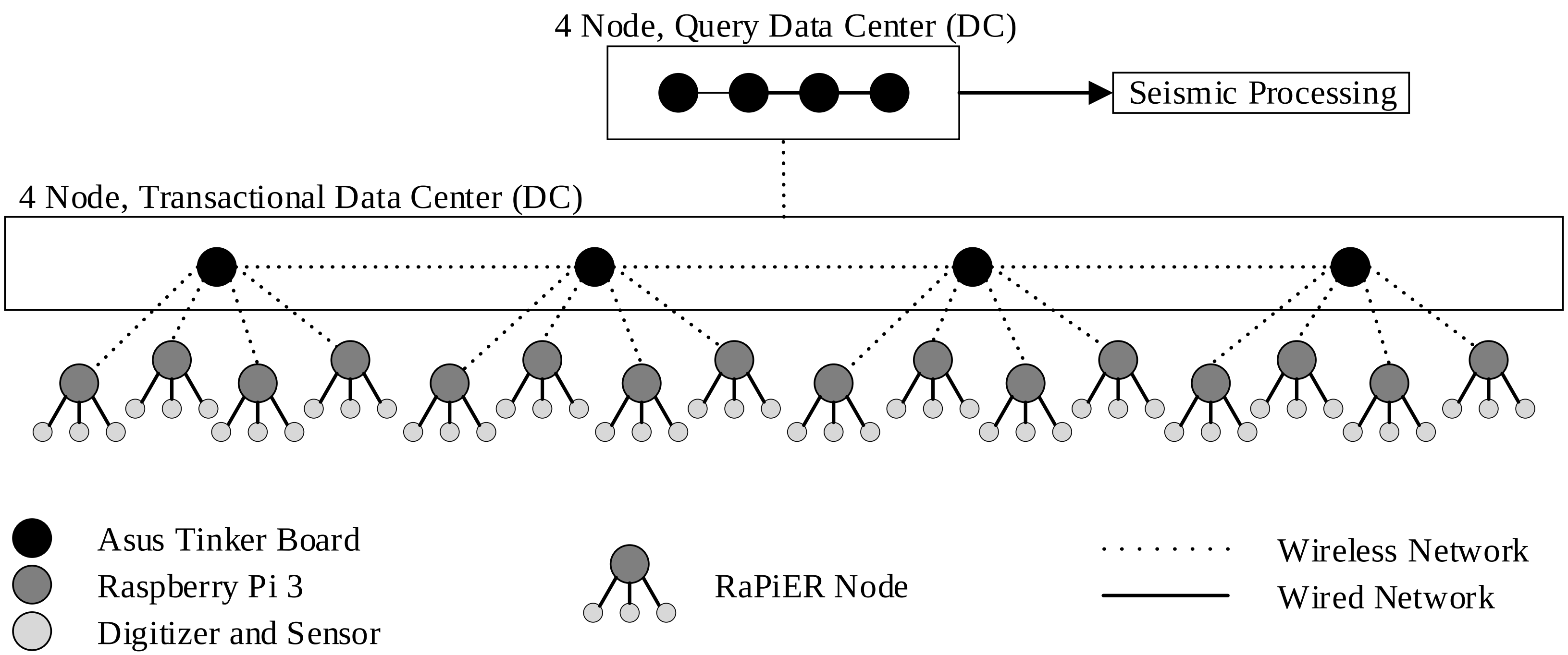


Figure 3.3. BRIC T&E event architecture.

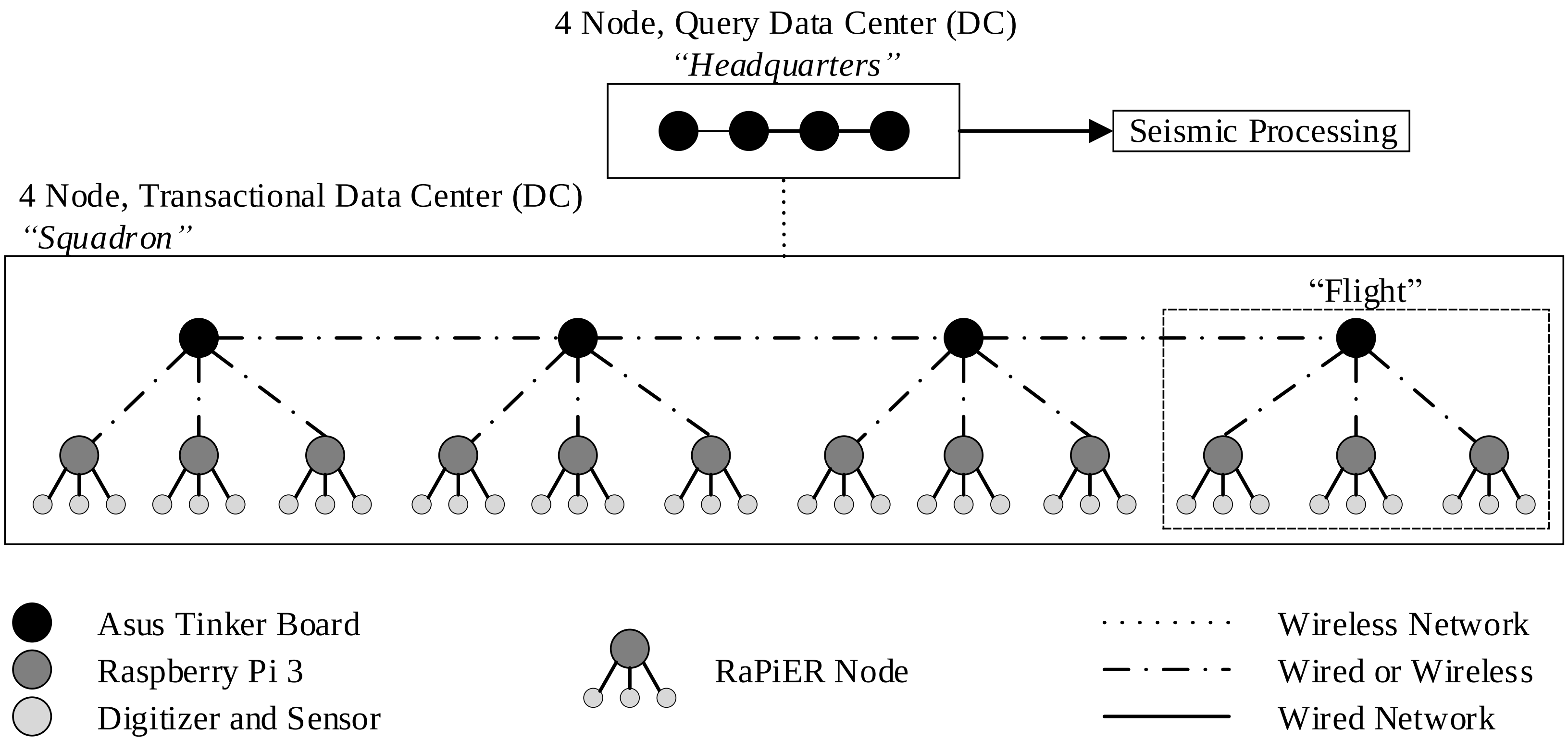


Figure 3.4. San Emidio T&E event architecture.

Telemetry Infrastructure

Over the course of four T&E events, our telemetry infrastructure evolved. Initially, we relied on a home-use wireless solution to transmit data from individual RaPiER nodes to a central location that was within a couple of hundred meters. In order to allow for an increased number of nodes spread over greater distances (i.e., thousands of meters), we transitioned from a home-use wireless solution to a more professional-grade wireless solution.

Eastland Lakes, Texas, May 2017 and Soda Lake Geothermal Field, Nevada, June 2017

The need to protect exposed Raspberry Pi 3 boards from the elements required that we place RaPiER nodes within enclosures located, on the ground, along our 2D line. Given the length of our 2D line and the location of enclosures on the ground, the Raspberry Pi 3's integrated onboard Wi-Fi was a suboptimal solution. As such, we leveraged a generic USB Wi-Fi dongle that allowed for the connection of an antenna external to the RaPiER's enclosure; thus, improving Wi-Fi signal quality. See Fig.3.5 for Eastland Lakes and Soda Lake T&E event telemetry infrastructure.

We relied on multiple EnGenius ENS202EXT (i.e., 802.11n) units operating as wireless access points (WAPs) and wireless repeaters to transmit data among the 20 nodes (i.e., Transactional DC) and transmit data to the three node Query DCs. Although effective, wireless repeaters represent a "first-generation" solution in which signal is acquired and rebroadcast on the same frequency; halving available bandwidth and increasing latency (Cossick, 2019).

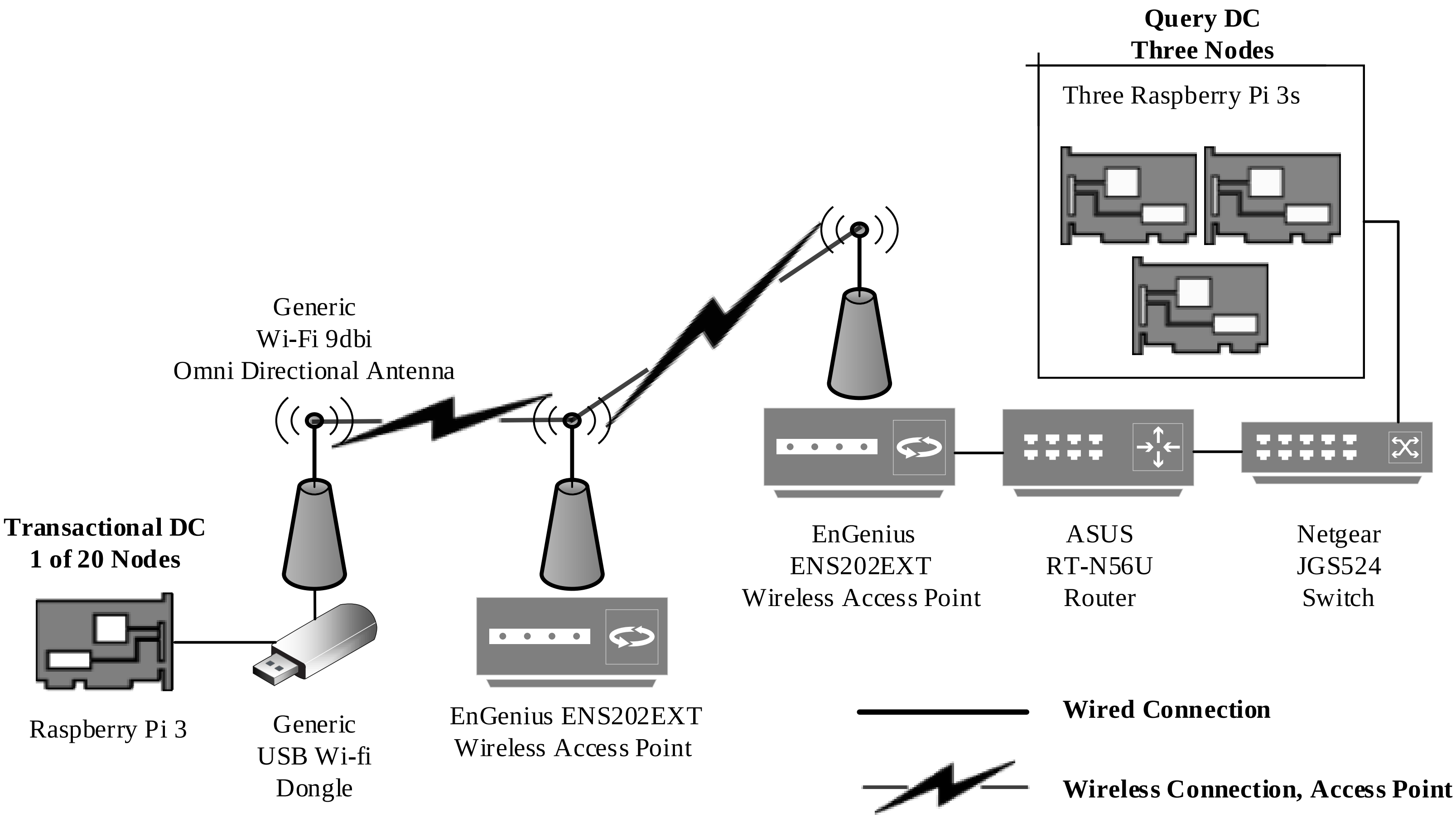


Figure 3.5. Eastland Lakes and Soda Lake T&E event telemetry infrastructure.

In addition to bandwidth limitations, we identified two other performance limiters that we believe were responsible for the network compartmentalization discussed previously. First, we relied on generic USB Wi-Fi dongles that proved to be unreliable due to driver related issues and/or insufficient power from our DC-to-DC converters. Second, we used a home-use router (i.e., ASUS RT-N56U) with limited configurability. We implemented the following upgrades to subsequent T&E events: 1) ASUS RT-N56U router to the Ubiquiti Edgerouter X router, 2) generic USB Wi-Fi dongles to TP-LINK Archer T2UH AC600 Wi-Fi dongles, and 3) 3 or 5-amp DC-to-DC converters to 10-amp DC-to-DC converters.

Baylor Research and Innovation Collaborative, Waco, Texas, July 2018

The need arose to protect Tinker boards from a storm, and we placed them within enclosures. Given the length of our 2D line and the location of enclosures on the ground, the Tinker's integrated onboard Wi-Fi required an antenna external to the Tinker's enclosure. To improve Wi-Fi signal quality, we initially planned to leverage a generic USB Wi-Fi dongle that allowed for the connection of an antenna external to the Tinker's enclosure. However, the integration of a Tinker and a Wi-Fi dongle introduced a complication.

USB Wi-Fi dongle driver support for the Tinker was limited. A Linux kernel mismatch prevented us from utilizing the Raspberry Pi 3's USB Wi-Fi dongle driver solution to solve the Tinker driver issues. Ultimately, we were unable to use the TP-LINK Archer T2UH AC600 Wi-Fi dongles with the Tinker. Fortunately, the Tinker allowed for the connection of an antenna directly to the Tinker board. Although an effective solution for the BRIC T&E event, we would need revisit Tinker specific Wi-Fi connectivity for a subsequent T&E event.

We continued to rely on multiple EnGenius ENS202EXT units operating as WAPs and wireless repeaters to transmit data amongst the four Transactional DC nodes and transmit data to the Query DC for the BRIC T&E event. We intentionally delayed upgrading (i.e., 802.11ac) our wireless solution (e.g., WAPs, repeaters, etc.) until we verified the suitability of DSE Advanced Replication for our specific purpose and benchmarked data insertion rates given a real-world wireless implementation. However, we did upgrade the router, USB Wi-Fi dongles, and DC-to-DC converters to enhance telemetry reliability. Fig. 3.6 illustrates BRIC T&E event telemetry infrastructure.

The implementation of our tiered configuration, in conjunction with the utilization of DSE Advanced Replication, improved overall performance and reliability. However, as previously mentioned, we were forced to stagger the insertion of data across the Transactional Cassandra cluster due to bandwidth limitations. Subsequent T&E events required that we upgrade our wireless solution, better balance our architecture (i.e., number of RaPiER per Tinker), and downsample seismic data.

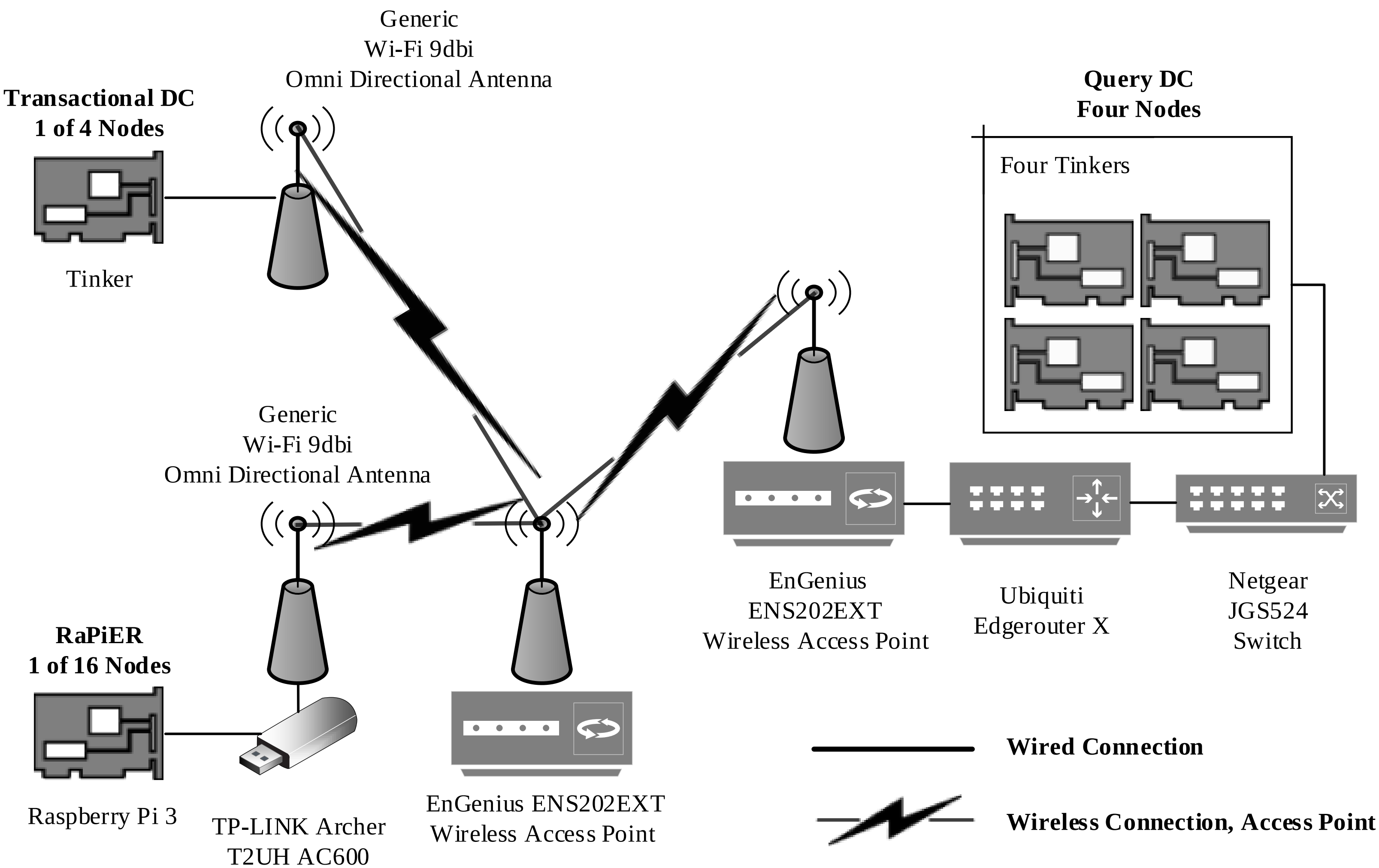


Figure 3.6. BRIC T&E event telemetry infrastructure.

San Emidio Geothermal Field, Nevada, May 2019

Ultimately, we decided not to use the TP-LINK Archer T2UH AC600 Wi-Fi dongles with the Tinkers. We connected individual Tinkers to a radio via the Tinker's Ethernet port and a switch. These radios facilitated squadron-to-headquarters, squadron-level Tinker to Tinker, and squadron-level Tinker to Raspberry Pi 3 communication; use of Wi-Fi varied depending on our specific telemetry configuration (Fig. 3.7-3.9). There were three upgrades to our San Emidio T&E event wireless solution: 1) transition from 802.11n to 802.11ac, 2) transition to a wireless distribution system (WDS) access point, and 3) implementation of backhaul telemetry. Details regarding these three wireless solution upgrades are provided below.

The 802.11ac standard was established by IEEE in 2013 and is fully backward-compatible with previous Wi-Fi standards (e.g., 802.11a, 802.11b, etc.). At 1300 megabits per second (Mbps), 802.11ac is theoretically 3x faster than 802.11n (i.e., 450 Mbps). However, real-world 802.11ac and 802.11n speeds are closer to 720 Mbps and 240 Mbps, respectively (Kelly, 2014). More importantly for our purposes, beamforming is built into the 802.11ac standard which makes 802.11ac speeds inherently better than 802.11n at greater range (Kelly, 2014). In order to take advantage of 802.11ac performance enhancements, we transitioned from EnGenius ENS202EXT to EnGenius ENS500EXT-AC WAPs for the San Emidio T&E event.

A WDS access point allows users to interconnect multiple WAPs into a single network without the need for wired connections (Kalinich, 2010). Although WDS access point theoretically remains subject to the "halving available bandwidth" limitation (Kalinich, 2010), some vendors (such as EnGenius) implement proprietary WDS access point enhancements (e.g., EnJet™, Wave 2, etc.) to maximize throughput and increase reliability. Our upgrade to a WDS access point from “first-generation” wireless repeater technology improved the overall quality of our San Emidio T&E event wireless solution.

Prior to the San Emidio T&E event, we configured our wireless solutions as a single network to which all nodes connected wirelessly in a similar manner. However, by increasing the number of seismic stations (i.e., 20/48 to 144) and increasing the overall distance of our 2D line (i.e., approximately 500 to approximately 2000 meters), we needed to implement independent backhaul telemetry between each squadron and headquarters. We chose the EnStation5-AC wireless bridge as our backhaul telemetry solution going forward.

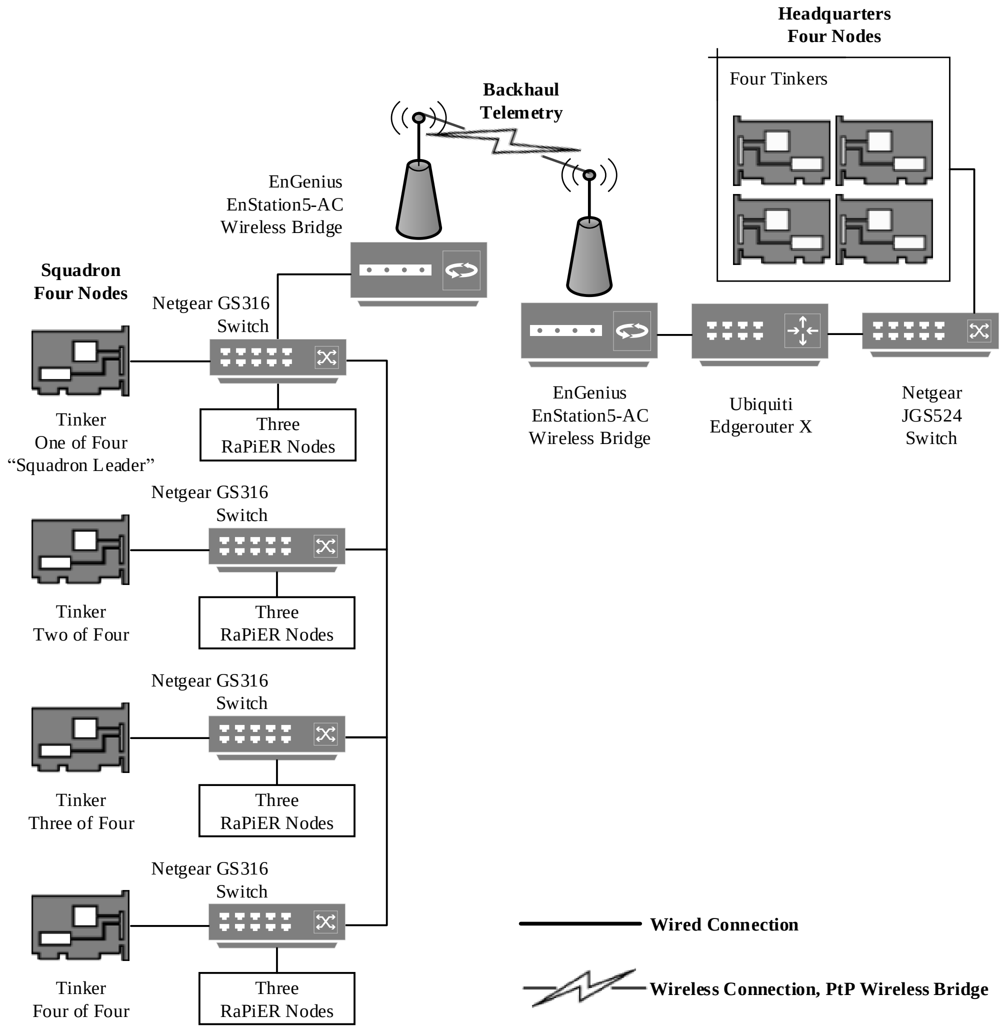


Figure 3.7. San Emidio T&E event “wired” telemetry configuration.

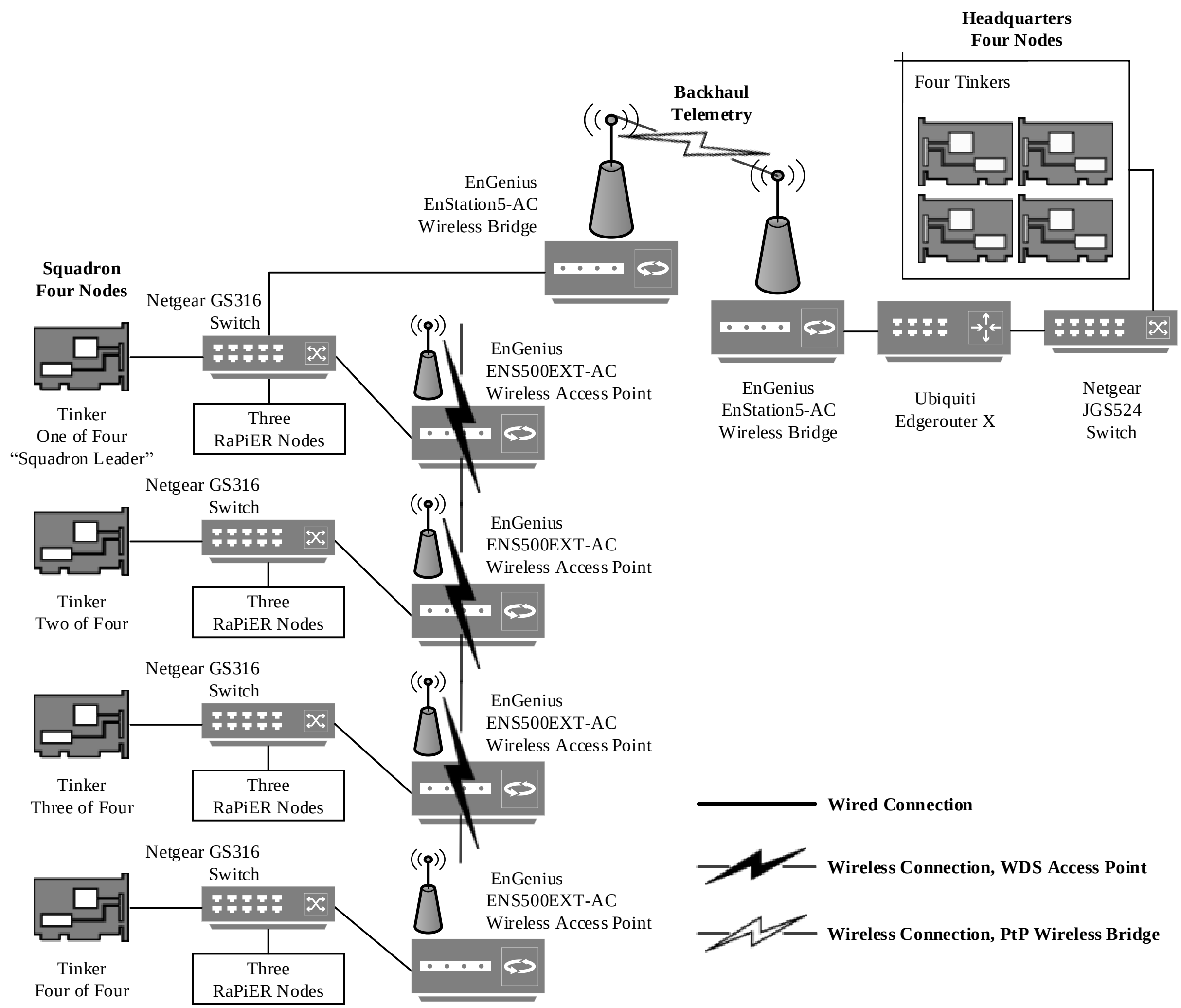


Figure 3.8. San Emidio T&E event “hybrid” telemetry configuration.

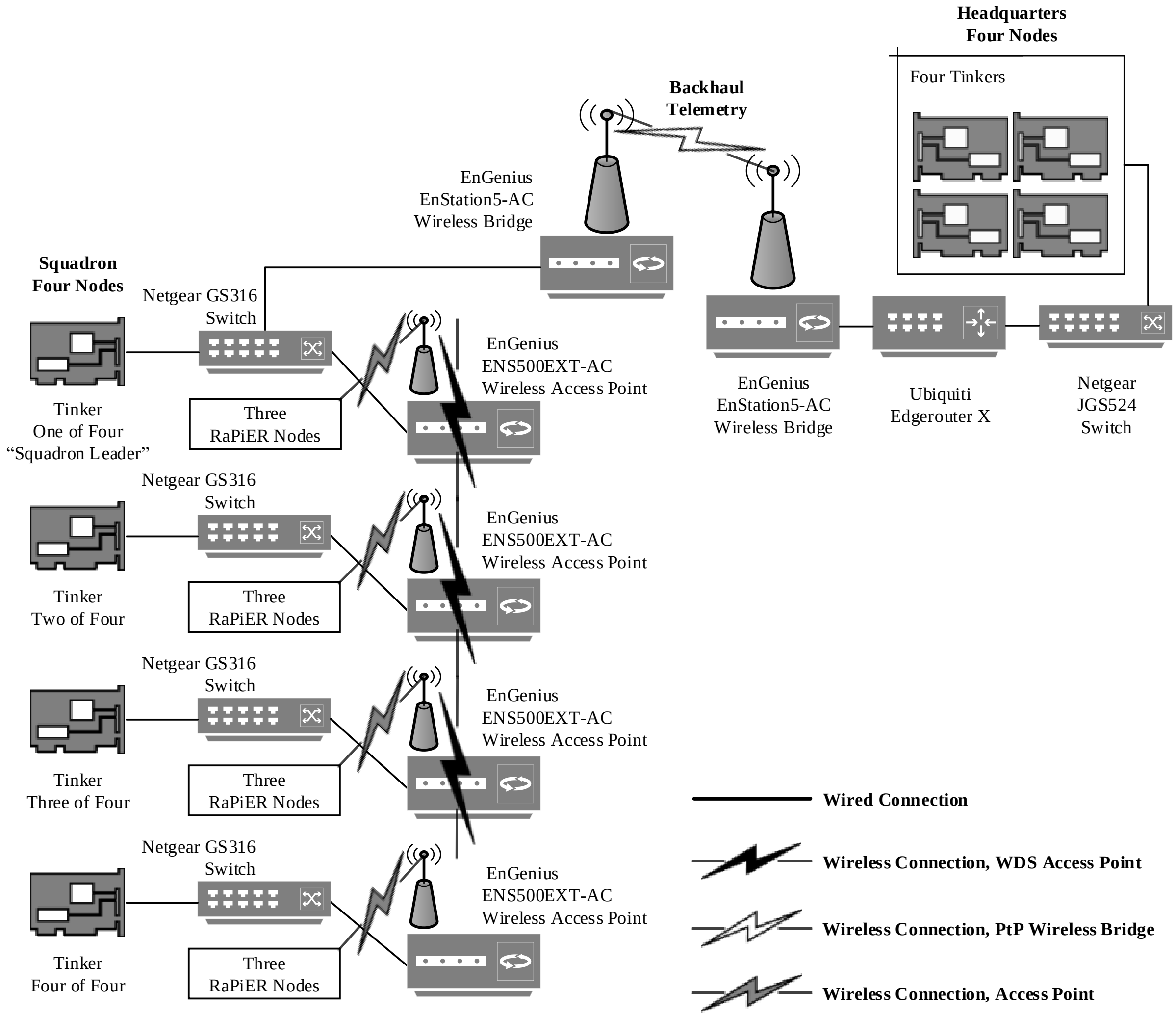


Figure 3.9. San Emidio T&E event “wireless” telemetry configuration.

To evaluate the feasibility of various telemetry options, we implemented three telemetry configurations during the San Emidio T&E event: 1) “wired,” 2) “hybrid,” and 3) “wireless.” Table 3.1 summarizes San Emidio T&E event telemetry configurations. In the “wired” configuration (Fig. 3.7), all squadron-level communication (i.e., REF TEK 130-01 to Raspberry Pi 3, Raspberry Pi 3 to Tinker, and Tinker to Tinker) was via Ethernet cables (i.e., Cat6). Squadron-to-headquarters (i.e., backhaul) communication was via a point-to-point (PtP) wireless bridge. The “hybrid” configuration (Fig. 3.8) utilized a WDS access point for squadron-level Tinker-to-Tinker communication. Ethernet cables were used for Raspberry Pi 3 to Tinker and REF TEK 130-01 to Raspberry Pi 3 communication. Squadron-to-headquarters communication was via a PtP wireless bridge. Lastly, the “wireless” configuration (Fig. 3.9) utilized a WDS access point for squadron-level Tinker-to-Tinker communication and a wireless access point for Raspberry Pi 3 to Tinker communication. Ethernet cables were used for REF TEK 130-01 to Raspberry Pi 3 communication. Squadron-to-headquarters communications was via a PtP wireless bridge.

Table 3.1. Summary of San Emidio T&E Event Telemetry Configurations

Telemetry Configuration	Squadron-Level: REFTEK 130-01 to Raspberry Pi 3	Squadron-Level: Raspberry Pi 3 to Tinker	Squadron-Level: Tinker to Tinker	Backhaul Telemetry: Squadron to Headquarters
Wired	Ethernet Cable	Ethernet Cable	Ethernet Cable	Point-to-Point Wireless Bridge
Hybrid	Ethernet Cable	Ethernet Cable	Wireless Distribution System Access Point	Point-to-Point Wireless Bridge
Wireless	Ethernet Cable	Wireless Access Point	Wireless Distribution System Access Point	Point-to-Point Wireless Bridge

Results and Discussion: T&E Event at Eastland Lakes, Texas May 2017

Fig. 3.10 shows the location of Baylor's Eastland Lakes facility and the T&E event's deployment geometry. Station locations, along with serial numbers of major components are tabulated in the data report archived at the IRIS DMC[6].

[6] https://ds.iris.edu/ds/nodes/dmc/

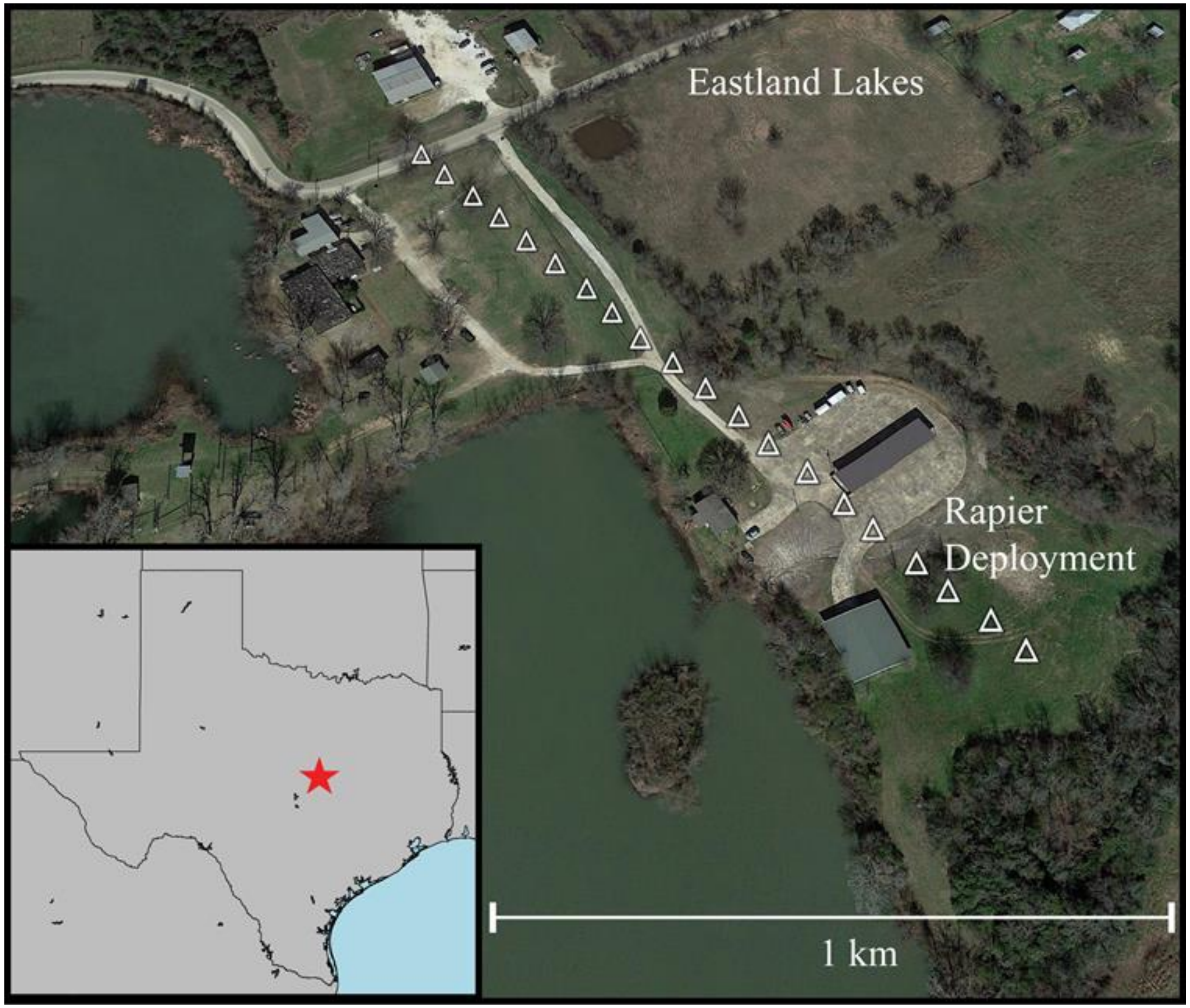


Figure 3.10. Location of the Eastland Lakes test site (Waco, Texas).

Data Products

RTP files are closed and transferred to the Raspberry Pi 3's solid state drive every five minutes at time intervals that are staggered to balance network traffic. There they are inserted into Cassandra. Raw data recorded at Eastland Lakes are shown in Fig. 3.11; station spacing is 15 m.

Fig. 3.12 shows a source gather for hammer blows generated at station 13. The straight lines indicate direct arrivals of a compressional wave with velocity 520 m/s at left and a shear wave with velocity 320 m/s at right. Arrivals that marked a compressional wave with a velocity of 520 m/s are indicated. These arrivals suggest Vp/Vs equal to 1.62. Hammer blows were performed to test and calibrate all the instruments and to help identify waves that are identified in virtual source gathers that are produced with ambient noise. Times of all hammer blows are tabulated in the data report archived at the IRIS DMC.

Figures 3.13 and A.1 (i.e., supplemental material) show virtual source gathers (VSGs) generated at Eastland Lakes after 25 hours of data acquisition. VSGs are produced automatically after each hour of data acquisition by the RaPiER array. However, each hour's data is added to all data recorded previously and the complete dataset is processed after each hour. Over time, the VSGs are expected to converge to optimum Green's functions as waves arriving from off-axis cancel each other and new, on-axis sources contribute arrivals that stack constructively with previous arrivals.

In Fig. 3.13, each virtual source location is indicated at the top of each figure and the distance of each station from the virtual source location is indicated as "offset." Fig. 3.13 y-axis refers to time lag. Virtual source gathers are sometimes folded about the zero time lag but we prefer the unfolded presentation because it shows the waves' directions of approach. Waves that arrive from an off-axis directions are expected to cancel but waves that traverse the array along its axis in opposite directions will stack to opposite sides of the zero time lag (i.e., a wave traveling from, for example, left to right will have negative time lags for the virtual source at right while a wave traveling from right to left will have positive time lags for the virtual source at the right). For example, Fig. 3.13a shows the VSG for station 19. The geophone at station 20 appears to have been poorly coupled to the ground compared to the majority of the other stations' geophones. Nevertheless, arriving waves are still in evidence and a rough moveout (change in time of arrival with distance) can begin to be discerned. Station 20 (Fig. 3.13b) shows moveout more clearly.

Note that trace 5 appears to be unusually noisy in Fig. 3.13a,b. This is partly because the plots are trace-normalized and the arrival at zero lag is small. However, it is also likely that the geophone at station 5 suffered from poor coupling (i.e., placed at the gravel edge of a paved road). We attempted to troubleshoot the station, its cables, and geophone but none of our efforts produced a significant improvement. While station 5 shows clear arrivals in some time intervals, the data used to produce these VSGs represents the total amount of data acquired during the test, so improvements that resulted from troubleshooting are included with the problematic data. As a result of our experience with the Eastland Lakes T&E event we developed a set of real-time assessment tools, including data metrics, which we employed subsequently. These allow us to identify issues and respond more quickly.

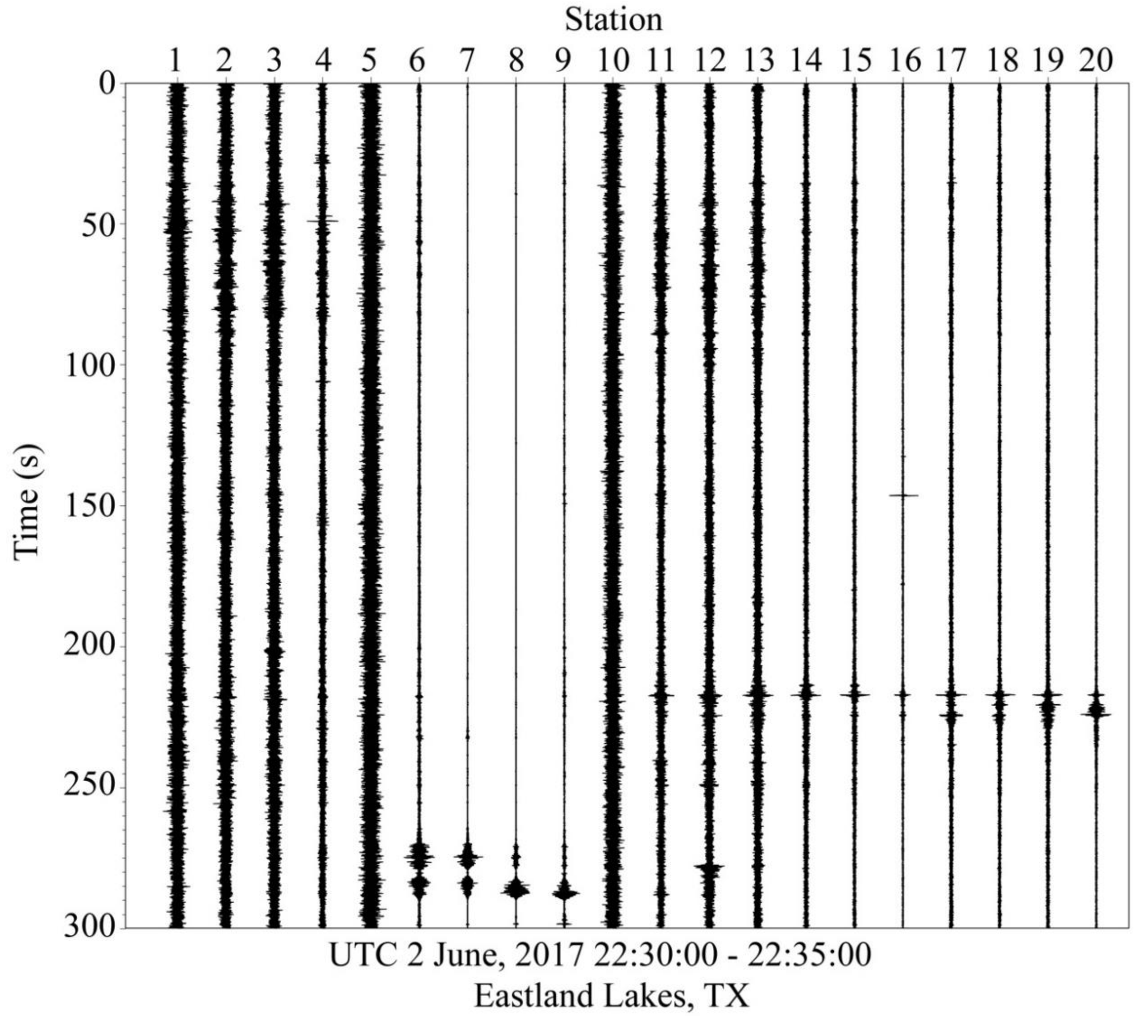


Figure 3.11. Sample of raw RTP files recorded during the Eastland Lakes T&E event.

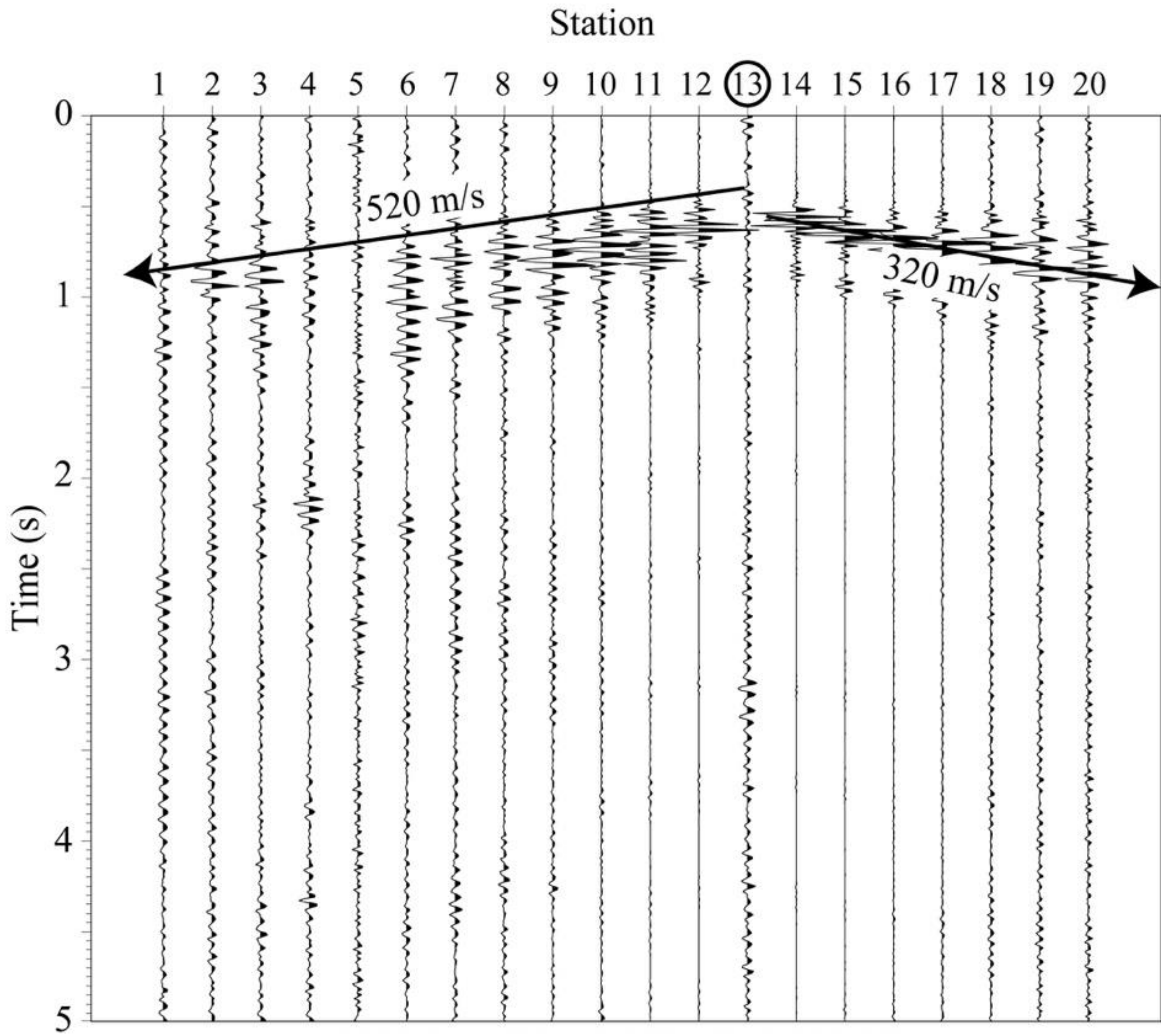


Figure 3.12. Shot gather for hammer blows at station 13 during the Eastland Lakes T&E event.

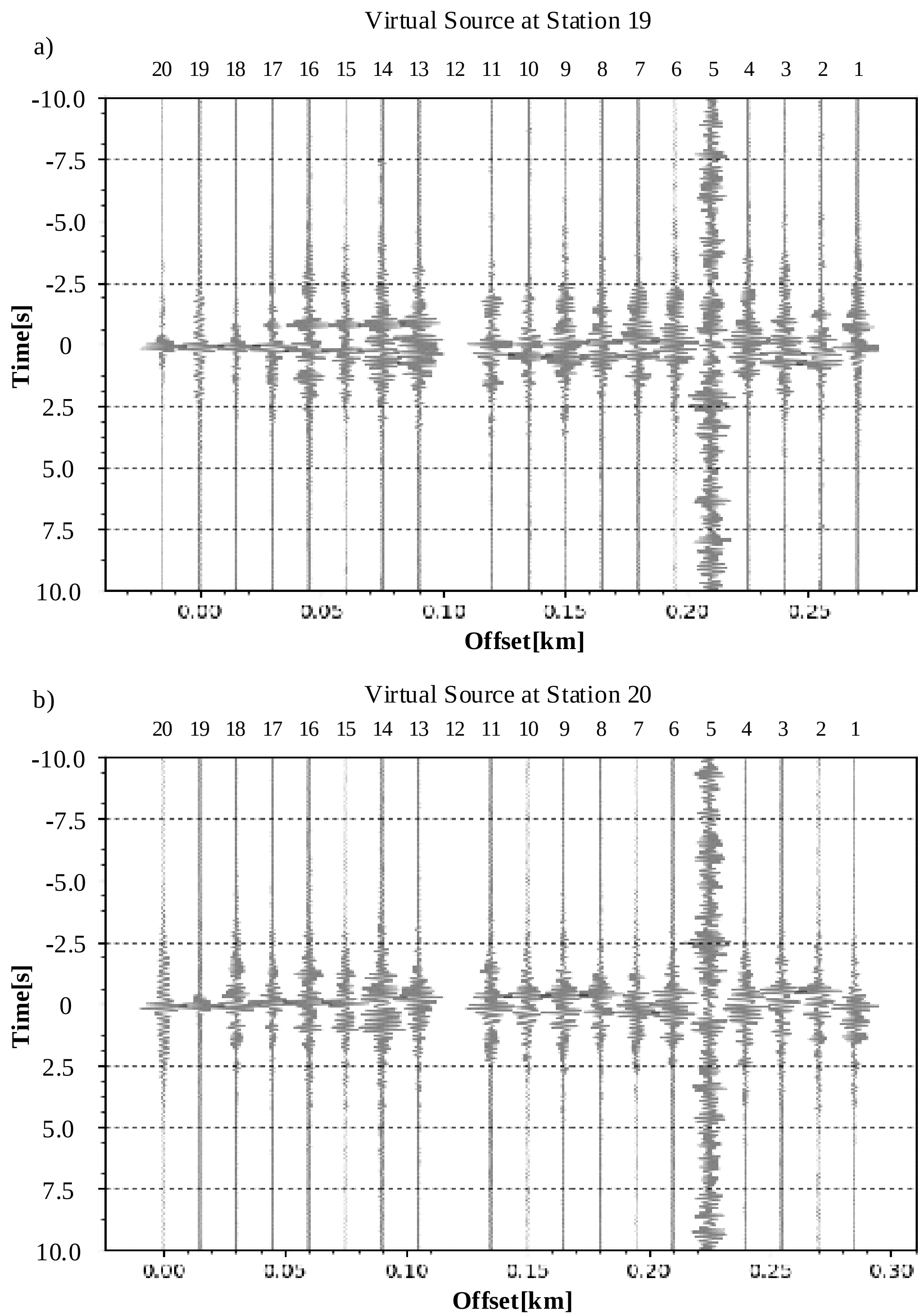


Figure 3.13. VSGs at a) station 19 and b) station 20 after 25 hours of data acquisition and processing of Eastland Lakes T&E event data.

Among the supplemental figures, VSGs become richer and more complex from Station 11 (Fig. A.1k) to Station 1 (Fig. A.1s). Multiple arrivals appear prominently in, for example, Stations 16 (Fig. A.1o) and 18 (Fig. A.1q), among others. These occur during a section of the line that is clearly a low-lying, now silted, settling pond. It rained heavily overnight during our survey and this part of the line filled with water, so some of our data were acquired before the rain and roughly ten hours of data were acquired during and after the rain.

The Eastland Lakes T&E event was intended to test the RaPiER array's functionality, including data acquisition, telemetry, and processing and an assessment of the data products. While the goal of deployment was to produce VSGs, the clarity, impulsiveness, and number of arrivals identified in those VSGs was not paramount. The VSGs in Figures 3.13 and A.1 show asymmetric arrivals; the asymmetry is caused by waves arriving from directions other than along the axis of the array and not being canceled by arrivals from their opposite direction (i.e., 180° azimuth). Such asymmetry is sometimes unavoidable when working with ambient noise, since we cannot control the locations of sources, but if the array is left in place longer these wayward arrivals are more likely to be canceled. The levels of noise on the VSGs will likewise decrease with greater deployment duration, further, the peaks of arrivals will sharpen, and the number of "events" (arriving waves) will increase. These desirable effects are demonstrated in the VSGs from the Soda Lake T&E event presented in the following section.

Discussion

Overall, the results of the Eastland Lakes test were successful in that they produced clear arrivals with consistent moveout across the array, allowed us to determine a reasonable Vp/Vs ratio to begin 2D modeling, and identified several issues that needed to be addressed to improve the RaPiER array's performance. Most importantly, however, it led to the conclusion that the strategy we proposed and design we created for automated seismic interferometry is workable, useful, and inexpensive. Further, computations were performed more quickly than anticipated, producing results within 15 minutes after each hour's data acquisition was completed.

Issues that needed to be addressed included a) improving the robustness of connections and, ideally, components such as the Wi-Fi dongles, b) leaving the array in place for a longer duration (to allow off-axis waves to cancel), and c) developing real-time quality control metrics and figures that would allow us to identify shortcomings easily. Complete failures are easy to spot but "less-than-ideal" data quality is less obvious. However, with the real-time assessments we developed after Eastland Lakes we can mobilize troubleshooting efforts more quickly and focus on the offending components or issues more acutely.

Power and telemetry. Power systems performed well, with battery voltages dropping modestly (i.e., approximately 0.4 volt) over 24 hours of continuous operation (without solar panels recharging batteries). Telemetry bandwidth appeared more than adequate for data transfer and command and control (i.e., one channel sampled at 100 sample/s). We experienced issues connecting to nodes via ssh from the central hub; however, it is likely these issues were related to network compartmentalization rather than network drops. Wi-Fi dongles performed reasonably well overall but were affected by the relatively high ambient temperatures (i.e., approximately 37 C) and occasionally the Raspberry Pi 3 would fail to initialize the Wi-Fi dongle after a reboot. A subsequent reboot was required to properly initialize the Wi-Fi dongle. It is possible that this is a low-power issue related to the 3-amp limit of our DC-to-DC converters.

Data. RaPiER single node data acquisition performed well. However, the RaPiER node RTPD archive status files became corrupted several times, likely due to multiple RaPiER reboots, and had to be rebuilt for REF TEK 130-01 data acquisition to continue. Of the 20 nodes, only two nodes experienced REF TEK 130-01 data loss over the course of approximately 18 hours of continuous acquisition.

Results and Discussion: T&E Event at Soda Lake Geothermal Field, Nevada June 2017

Fig. 3.14 shows the location of the T&E event at Soda Lake Geothermal Field near Fallon, Nevada and the event's the deployment geometry. Station locations, along with serial numbers of major components are tabulated in the data report archived at the IRIS DMC.

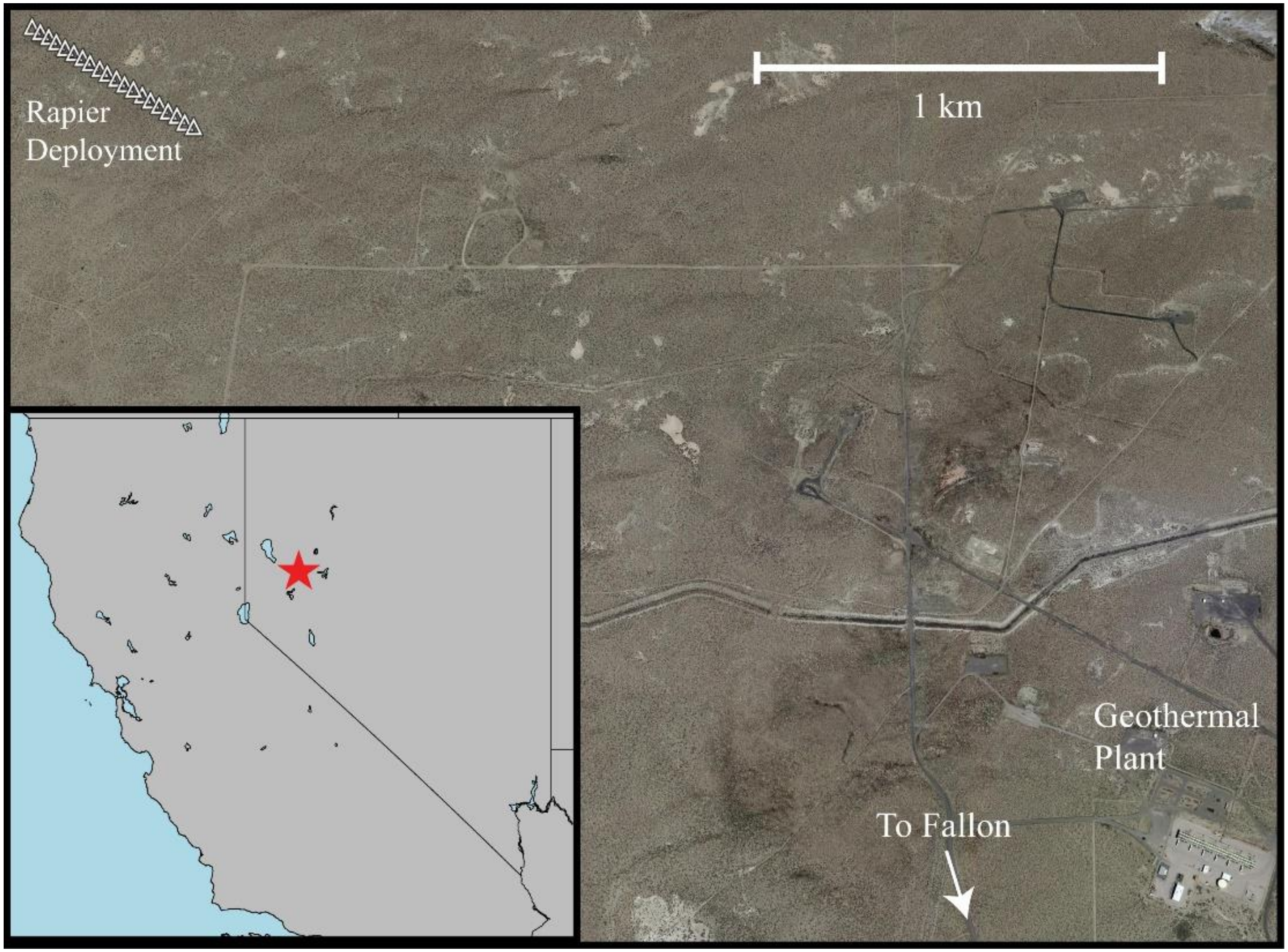


Figure 3.14. Location of Soda Lake test site (Fallon, Nevada).

Data Products

Note that the geothermal plant operated by Cyrq Energy is roughly 2.5 km from the end of the deployment and is nearly aligned with the axis of the array (Fig. 3.14). The plant is likely to be a source of energy recorded by the array, which may or may not be beneficial. Its seismic energy is more likely to be partitioned into surface waves than body waves and is therefore less useful for reflection imaging.

Each station's REF TEK 130-01 digitizer was programmed to produce data files of 5 minutes duration. A continually running process on the RaPiER ('cron job', in Linux language) wrote each file to the RaPiER's Micro SD card. An example of a five-minute data file written by a REF TEK 130-01 in its RTP format is shown in Fig. 3.15.

The data were then inserted into Cassandra and another cron job extracted data from Cassandra in 1-hour intervals and converted the data to miniseed format. Miniseed is a binary format that is widely used in seismology and is the format required by MSNoise[7], the program we use to compute and stack cross-correlations. Miniseed files in our implementation are "cumulative," in the sense that the file produced at each hour for each station contains the total amount of data recorded by that station since the processing "block" was started. For example, the RaPiER array computes the first set of Green's functions two hours after recording has begun and then computes a new set every hour after that. At hour 2, two hours of data are processed. At hour 3, three hours of data are processed: the same two hours that were processed previously plus additional data that were recorded in the intervening hour. At hour 4, four hours of data are processed,

[7] http://www.msnoise.org/

and so on. While this is computationally inefficient, the time required to perform all the computations is negligible, even on the Raspberry Pi 3.

Fig. 3.16 shows a source gather for a hammer blow at station 11 at Soda Lake Geothermal Field. The station spacing is 30 m. Station 6 (not shown) malfunctioned due to extremely high temperatures. Straight lines indicate the direct arrivals of a compressional wave with a velocity of 270 m/s and a shear wave with a velocity of 167 m/s, resulting in a Vp/Vs of 1.61. We performed a number of hammer blows to check array performance and to identify target arrivals with their wavespeeds.

Fig. 3.17 shows VSGs in which each station serves as a virtual source. Figures 3.17 and A.2a-g show a strong arrival at positive time lags that traverses the entire array. Another strong arrival appears at negative time lags from Stations 12-20 but that one folds into the positive-lag arrival and therefore represents the same “event” (wave arrival). There are additional hints of both earlier (faster) arrivals and later (slower) arrivals at positive time lags. Fig. 3.17b, the VSG for Station 19, shows the same strong arrival as in the VSG for Station 20 but the faster arrival is much clearer, as well. Later events are less clear than at Station 20.

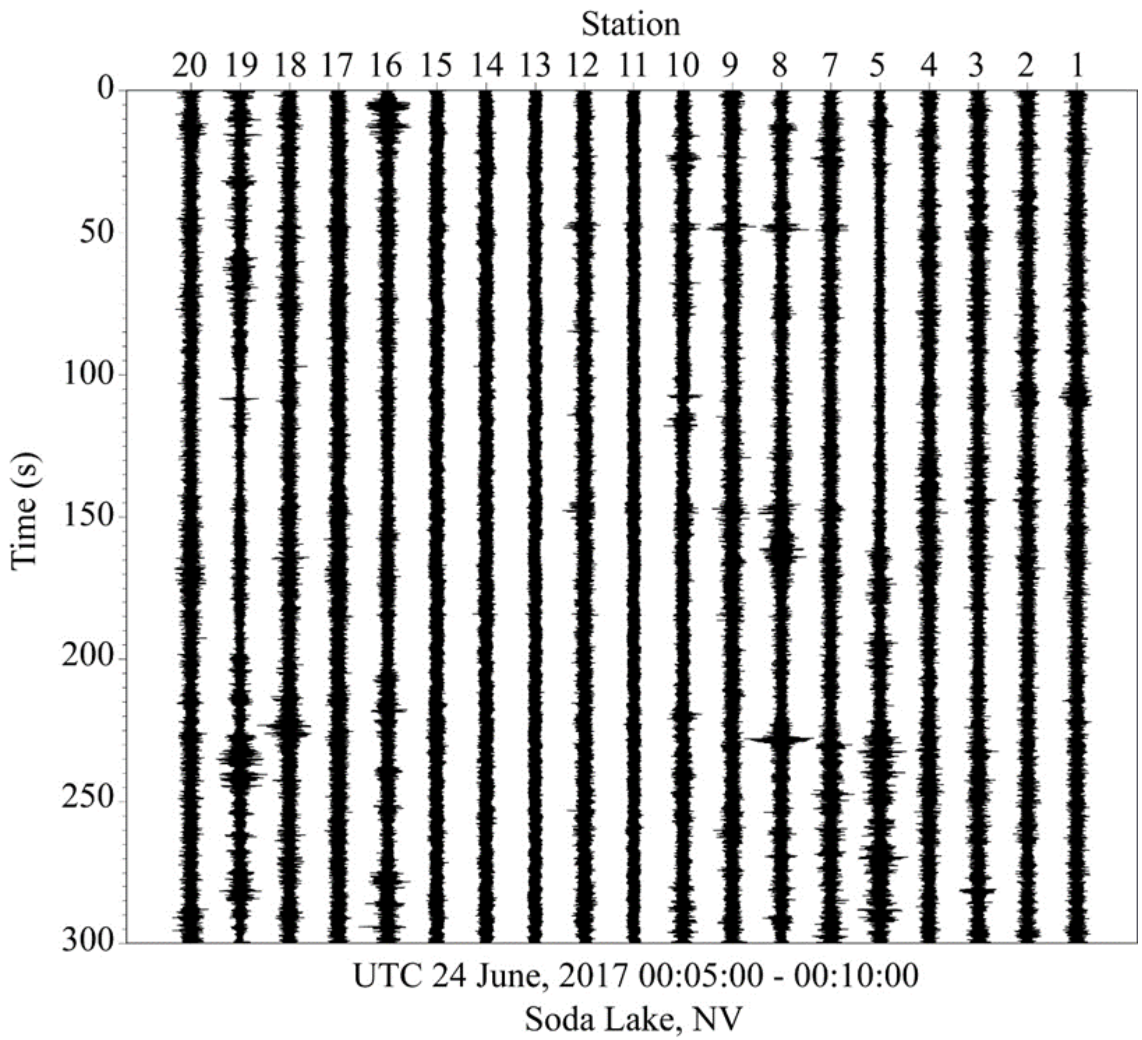


Figure 3.15. Sample of raw RTP files recorded during the Soda Lake T&E event.

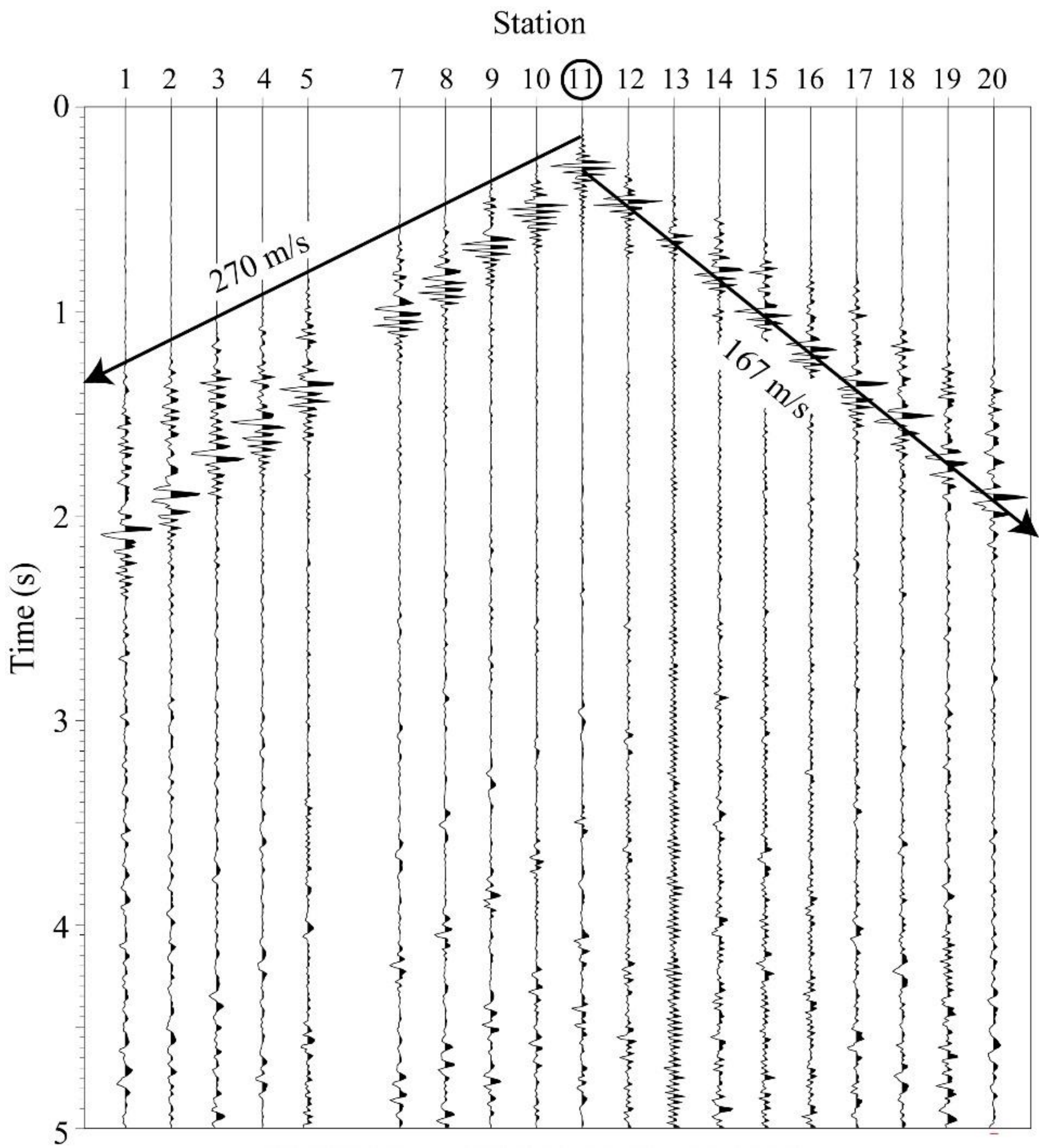


Figure 3.16. Shot gather for hammer blows at station 11 during the Soda Lake T&E event.

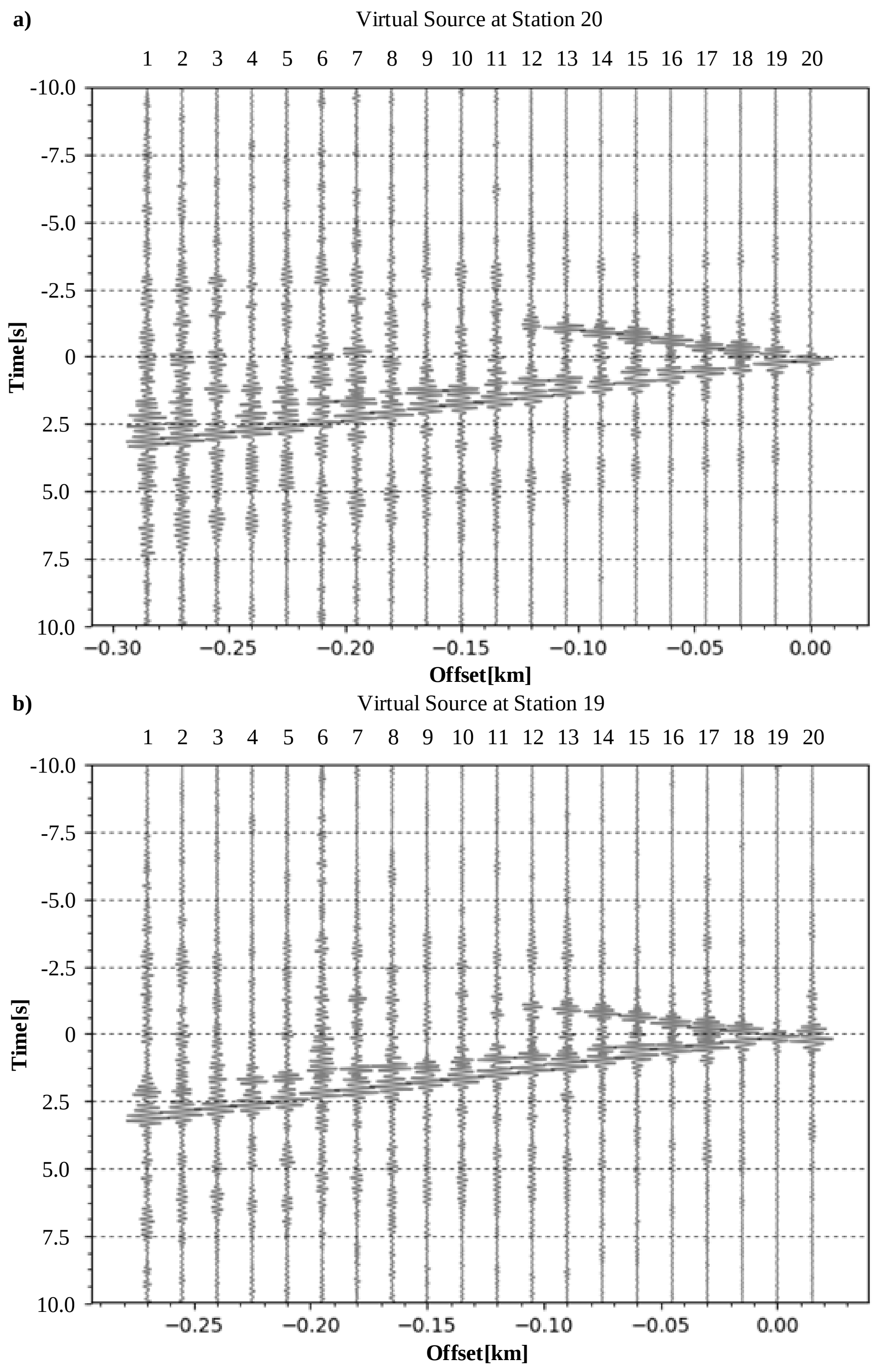


Figure 3.17. VSGs at a) station 20 and b) station 19 produced, after 45 hours of data acquisition and processing of Soda Lake T&E event data.

As the virtual source moves from right to left in Figures A.2a-g, the arrival from the opposite end of the line (the wave arriving at negative time lags) remains clear. However, the arrival is much less clear in Figures A.2h-r. There are other differences as well. Figures A.2a-g clearly show later arrivals on the left side of the line (negative offsets) whereas later arrivals on Figures A.2h-t are much less clear. Given that the events appear on other VSGs, they must also exist on Figures A.2h-r and their amplitudes could probably be boosted with additional processing. Note the tendency of the traces at greatest offsets from the virtual source in each gather to have the greatest levels of noise. This just reflects the drop-off in amplitude as a function of distance. Each Green's function (trace) on each gather is normalized to its greatest amplitude, so the appearance of more noise on distant traces indicates the smaller difference between noise and wave amplitudes at that distance. The absolute amplitudes of noise are likely to be the same on each trace for a given VSG.

Note the improved appearance overall of the VSGs produced at Soda Lake (Figures 3.17 and A.2) compared to those produced at Eastland Lakes (Figures 3.13 and A.1). The improvement is due to the longer deployment duration (45 hours vs. 25 hours) but also is due to the frequency content and directionality of the ambient noise at each site. The sandy soil in which we installed geophones should have produced less strong coupling than at Eastland Lakes (where soils had a much greater clay content) but that is not evident in the VSGs. It's possible the longer deployment duration (i.e., greater "fold" of the stacks) overwhelmed the lower amplitudes produced by poor coupling. Or perhaps the coupling was comparable for the vertical components at each location and the difference will be confined to the horizontal components (which we have not worked with yet).

Regardless, the difference between the two sets of VSGs highlights the need for an automated, near real-time acquisition and processing system. Ideally, one would not want to end the Eastland Lakes deployment after 15 hours and the Soda Lake deployment may have gone on longer than was necessary. We will discuss Green's function convergence in more detail below. In particular, we will show that 20 hours might well have been adequate for the deployment at Soda Lake even though 25 hours was (most likely) not adequate to characterize the subsurface at Eastland Lakes.

Discussion

Power. The Micro-USB B Connector to DC power, via DC-to-DC (i.e., 12 volt to 5 volt) converter that we fabricated caused power issues. These issues caused intermittent drops in data acquisition and network connectivity. The relatively low-quality connectors (i.e., Micro-USB B) were often damaged during connection or disconnection to the Raspberry Pi 3 and small movements of the damaged connector subsequently caused unplanned power cycling. A sporadic loss of data resulted from the Raspberry Pi 3 being powered down or the RTP files being corrupted during power cycling.

We also observed, while troubleshooting, that the USB Wi-Fi dongle occasionally did not power up correctly during the initial boot of the Raspberry Pi 3. This issue occurred randomly, approximately 10-15% of the time, while using the power cables we fabricated. It is possible that the 3-amp limit of the DC-to-DC converter is insufficient.

We recommend using a 5-amp or 10-amp DC-to-DC converter and splicing the converter to a factory terminated Micro-USB B connector.

Telemetry. Although telemetry bandwidth was more than sufficient for our needs (i.e., one channel at 100 samples/s), we experienced multiple instances of "compartmentalization" in the Wi-Fi network. Compartmentalization prevented stations, connected to different repeaters, from communicating as expected. We spent a considerable amount of time troubleshooting the access point, repeaters, and the router; however, we were unable to diagnose and reconfigure the network to eliminate compartmentalization reliably. We recommend that subsequent efforts incorporate a professional-grade, as opposed to home-use, router. Professional-grade routers allow a network to be configured explicitly, with more advanced and flexible features, so that we can specify which stations connect to which access point or repeater.

Data. The last 2.5 seconds of every five-minute RTP file were not properly parsed and inserted into Cassandra. This likely occurred due to a bug in the Python parsing script. The total percentage of RTP data lost due to this bug is approximately 0.83%, so it did not significantly impact the results (and all the RTP data were preserved, as well, so no data were lost). The parsing script was subsequently reworked to handle all data in every RTP file properly.

Green's function convergence. Fig. 3.18 shows the average L2 norm computed for each VSG vs. hour (numbered from the beginning of data acquisition). We compute this value by finding the L2 norm (square root of the sum of the squares) of amplitudes for each trace, find the sum over all traces, then divide by the number of traces represented on the VSG. Failures, including telemetry, REF TEK 130-01, cable connectivity, or processing errors, can cause a station to fail to appear on any given VSG. We normalize by the number of traces that appear on a VSG.

The average L2 norm (i.e., norm) value can be expected to decrease as the results converge to correct Green's functions because most of the excursions on each trace are caused by random noise. The smallest norm for a trace would be one in which a wave arrival appears as a perfect delta function and all other amplitudes are zero. This will never be the case, of course, but we can approach this ideal value by stacking cross-correlation functions computing for successive time intervals. Random noise will stack destructively, decreasing in amplitude, while coherent wave arrivals will stack constructively, increasing their amplitudes.

The average norm on a VSG should, therefore, decrease over time but the nature of that decrease is critical. One can expect the norm to decrease rapidly during the first few hours as traces that consist primarily of random noise are replaced by traces that contain coherent waves. Improvements will continue but eventually the improvements in successive time intervals will no longer be significant. The threshold at which improvements are deemed to be "insignificant" is subjective, of course, but it is important to have quantitative tools such as these norms on which to base discussions, further study, and, ultimately, clear decisions.

For example, Fig. 3.18a shows a dramatic improvement through the 20^{th} hour for the VSG at Station 20; the improvement is even more dramatic in Fig. 3.18b (Station 19). After that, the average norm climbs a bit before ultimately returning to its previous value. This degradation is likely due to our use of a gas-powered generator that was needed to power notebook computers at our central staging facility. Recall that all data recorded up to a particular time are used to compute the VSGs for that time point, so “bad” data that are added during a particular interval will have an effect for a time but will ultimately be overwhelmed by additional “good” data. This appears to be what happened. We used the generator during the last half of the second day of recording when our notebook computers needed to be recharged, but then we left the site in the evening and did not start the generator the next day, leading to quieter conditions in the vicinity of the array.

Most, but not all, of the norm vs. time curves for other stations are similar to the curves shown in Fig. 3.18. Some stations (e.g., station 7) did not have data prior to hour 20 due to failures of Raspberry Pi 3 processors, which we replaced. The telemetry at Station 6 (Fig. 3.19) failed for a few hours but the processing continued as though new data were arriving, so the “changes” during those hours were exactly zero, which resulted in the same norm being computed. This is due to a small bug that we later fixed in our codes, in the latter case.

Fig. 3.20 shows examples of VSGs computed after 20 hours of data acquisition, for comparison to VSGs computed after 45 hours shown in Fig. 3.17. Fig. 3.20 caption summarizes some differences between the two sets of VSGs. The final decision that VSGs are “good enough” and that data acquisition should be ended is subjective and will differ from application to application and from group to group. Aside from its technical issues, greater convergence (“improvement”) can be achieved through longer deployments and more assiduous attention to troubleshooting, both of which involve greater costs.

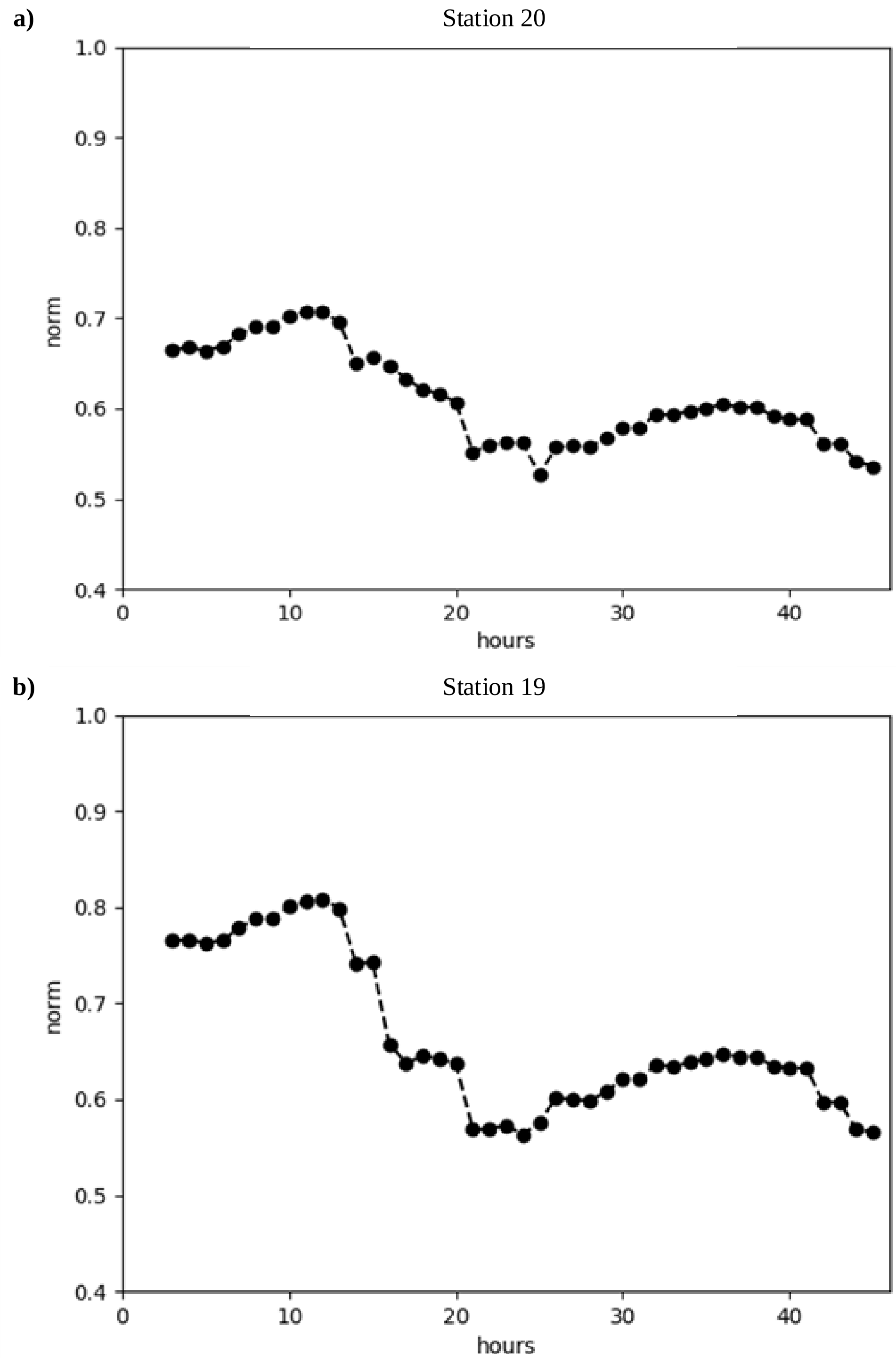
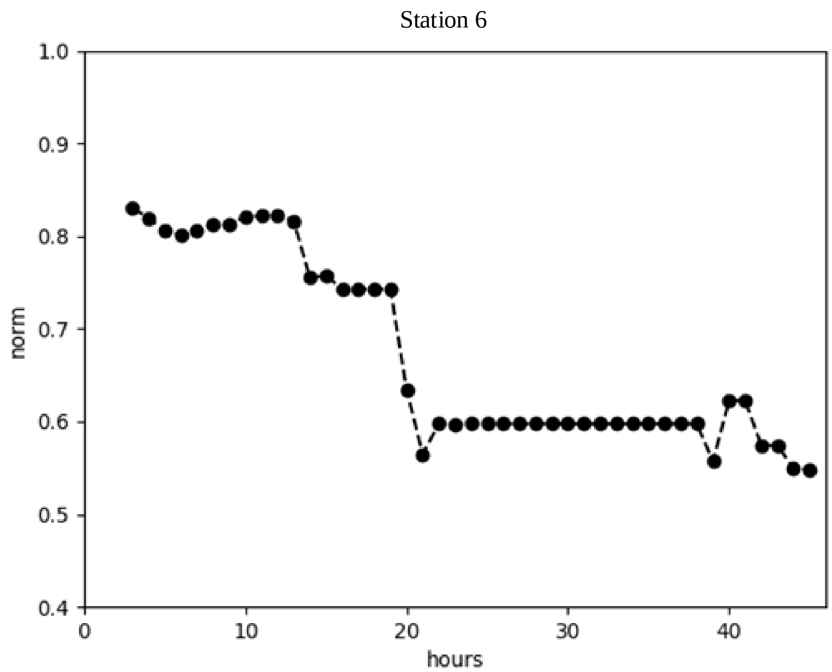


Figure 3.18. Average VSG L2 norm – a rough measure of Green's function convergence – as a function of total hours of data acquired and processed, for a) station 20 and b) station 19, during the Soda Lake T&E event.



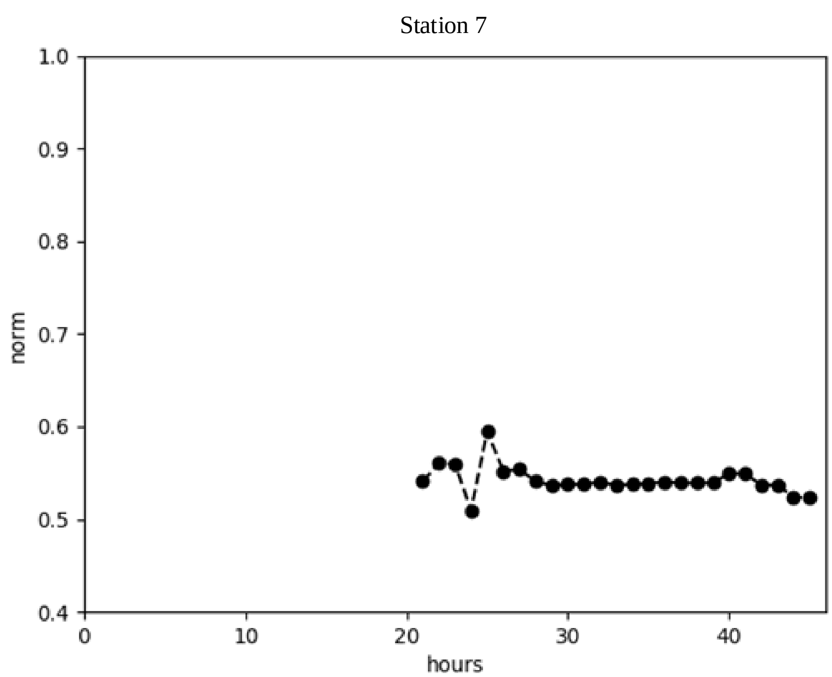
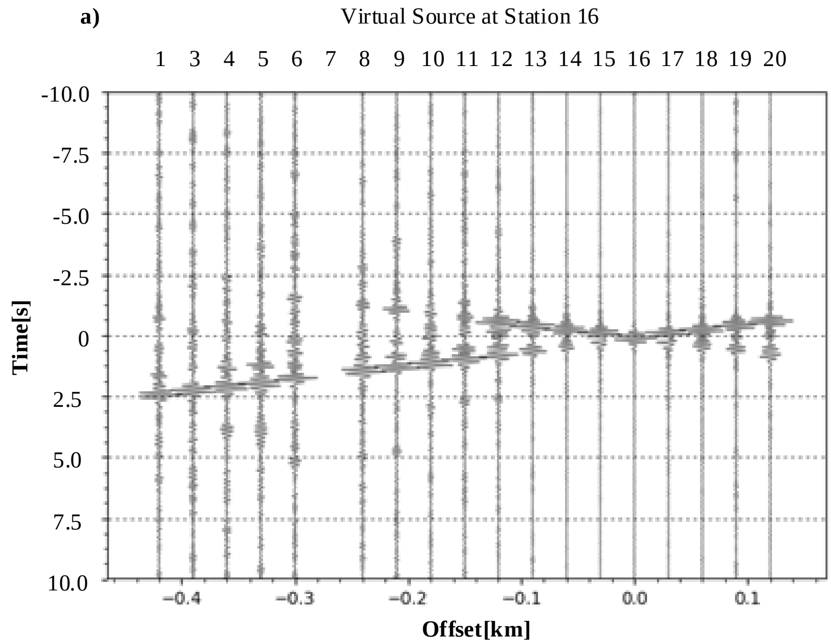


Figure 3.19. Average VSG L2 norm – a rough measure of Green's function convergence – as a function of total hours of data acquired and processed, for a) station 6 and b) station 7, during the Soda Lake T&E event.



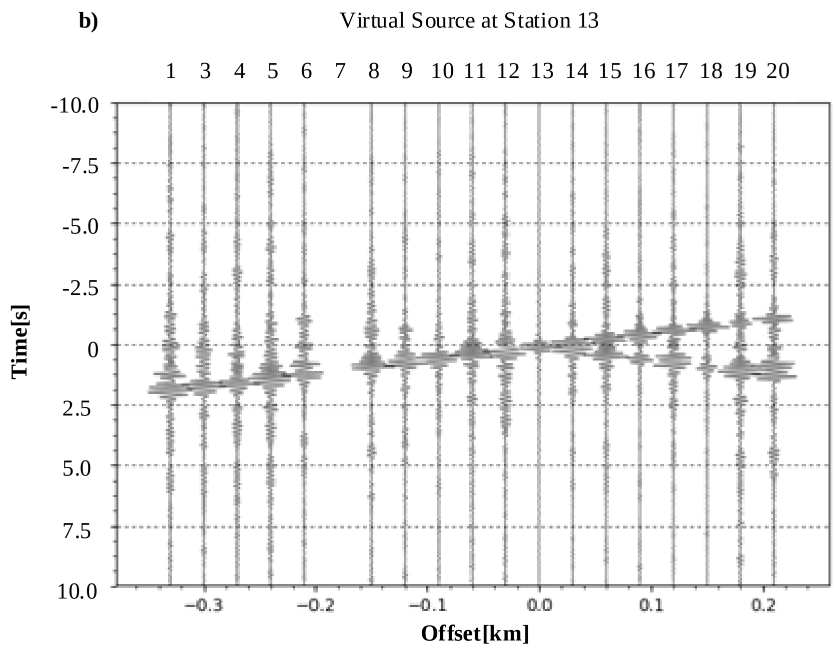


Figure 3.20. a) VSG for Station 16 after 20 hours of data acquisition and processing and b) VSG for station 13 after 20 hours of data acquisition and processing during the Soda Lake T&E event. The faster arrival at negative offsets and positive time lags is clearer and has greater time separation from the slower arrival after 45 hours than after 20 hours. Arrivals at positive offsets and positive time lags are clearer and extend over a narrower time interval after 45 hours than after 20 hours. Also, the two VSGs' noise levels at distant offsets are greater after 20 hours than after 45 hours.

We examined but rejected several other strategies for determining Green's function convergence, including the computation of "residual traces," in which we subtract each hour's VSG trace for a given station from the previous hour's trace and plot the result. While this is useful for identifying arrival times of certain waves, the most important result is cumulative (the sum of all the waves that traversed the array during our deployment) because seismic interferometry depends on waves that arrive from a particular off-axis direction being canceled by waves arriving from the opposite direction. A second approach involves computing and plotting the average norms of those "VSG residual" plots but that approach suffers from the same shortcoming as the residual plots themselves. Assessing minor changes in wave arrivals in small time increments is not a reliable indicator of convergence to the Green's function between two stations.

Results and Discussion: T&E Event at San Emidio Geothermal Field, Nevada, May 2019

Fig. 3.21 shows the San Emidio Geothermal Field (near Gerlach, Nevada) and the T&E event's deployment geometry. Station locations, along with serial numbers of major components are tabulated in the data report archived at the IRIS DMC.

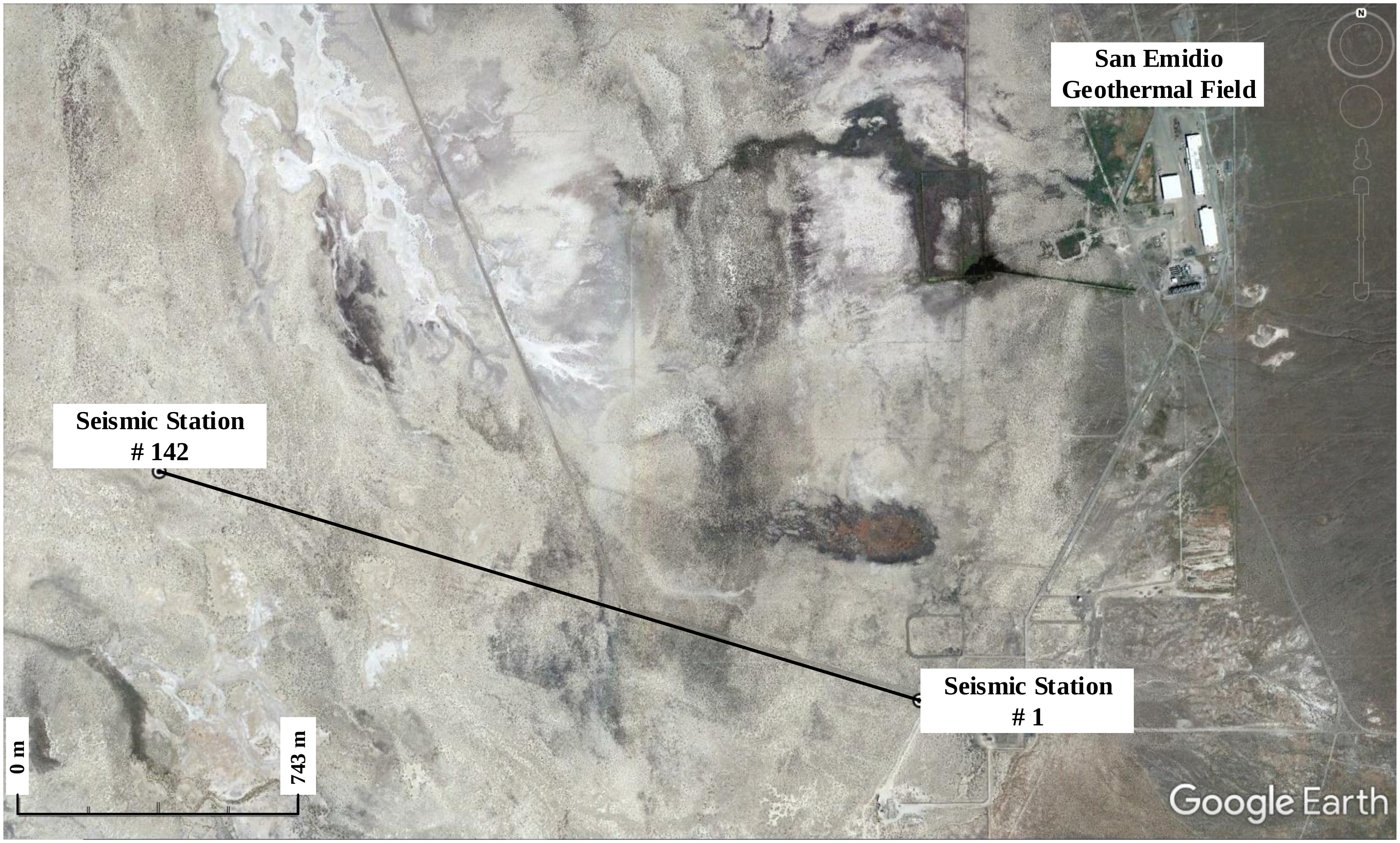


Figure 3.21. Location of the San Emidio Geothermal Field (Near Gerlach, NV). Station 1 and 142 are shown along an approximately 2120 m seismic line. The seismic line consists of 142 seismic stations (only end stations shown) spaced approximately 15 m apart.

We planned to deploy four squadrons and four headquarters totaling 144 seismic stations, 48 Raspberry Pi 3 (RaPiER nodes with three seismic stations each), 16 Tinkers (squadron-level), and 16 Tinkers (headquarters-level). A squadron would acquire approximately 778 million samples per day, insert approximately 156 million samples per day (downsampled) into a Transactional Cluster, and subsequently replicated to the Query cluster (i.e., headquarters).

Due to broken or missing REF TEK 130-01 components, unplanned troubleshooting, and weather-related delays, we only deployed 142 seismic stations, 8 Tinkers (squadron #1 and squadron #2), and 8 Tinker (headquarters #1 and headquarters #2). Regardless of the total number of squadrons deployed, the organization of squadrons and headquarters into independent groups allowed us to assess our data acquisition, transmission, storage, and processing.

Data Products

Fig. 3.22 shows the VSG for station 60 using all vertical-component data. The VSG's main features include a first arrival with velocity approximately 333 m/s, visible to the end of the array, and slower arrivals, visible to approximately 300 m offsets, which appear to be dispersive. The latter arrivals are almost certainly surface waves; the higher-amplitude first arrival's velocity is typical of air waves, although in the context of passive recording of ambient noise a significant air wave is not expected. Regardless, no reflected arrivals are obvious in the raw stacks. This is not surprising, given the array deployment over deep sediments on the valley floor.

The same arrivals appear prominently on the VSG for station 66 (Fig. 3.23a). Median filtering over seven traces decreases the coherence and amplitude of the first arrival (v=333 m/s) (Fig. 3.23b). Windowing a shorter time period (3 s) (Fig. 3.24) shows the effectiveness of median filtering in revealing the surface waves but no additional arrivals are apparent. However, applying a similar process to station 99 VSG (Fig. 3.25), where surface waves are less coherent reveals possible reflection events (highlighted). Regardless, the main goal of this deployment was to test and evaluate the RaPiER array's performance in terms of data acquisition and processing. We would expect to produce more revealing VSGs with higher gain sensors and/or deployment in areas without such deep, highly attenuating sediments.

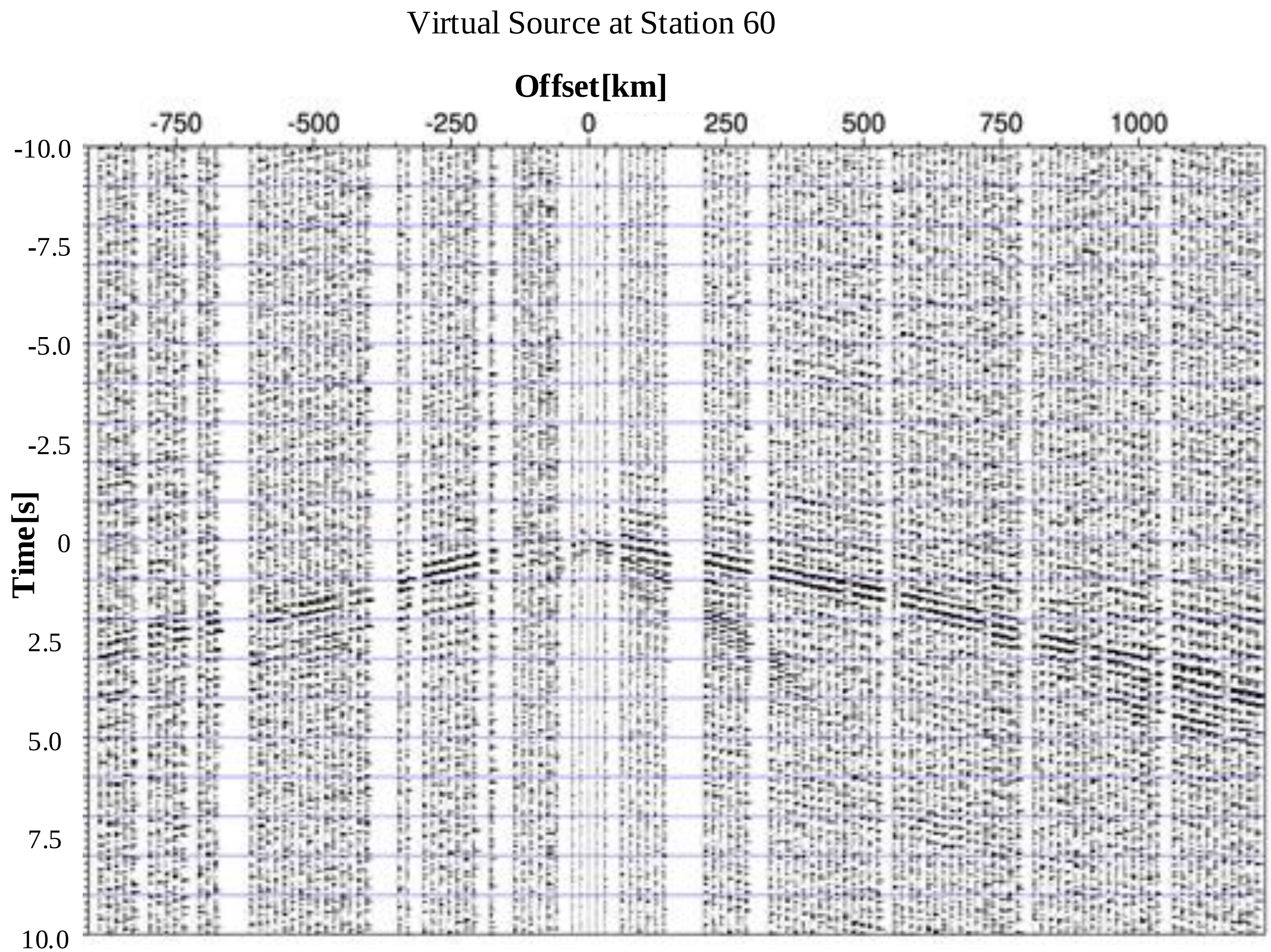


Figure 3.22. VSG for station 60 (San Emidio T&E event) shows a first arrival with velocity of ~333 m/s, which is similar to a typical velocity of an air wave, and dispersive surface waves, the slower, more steeply dipping, arrivals visible from 0-300 m offset.

a)

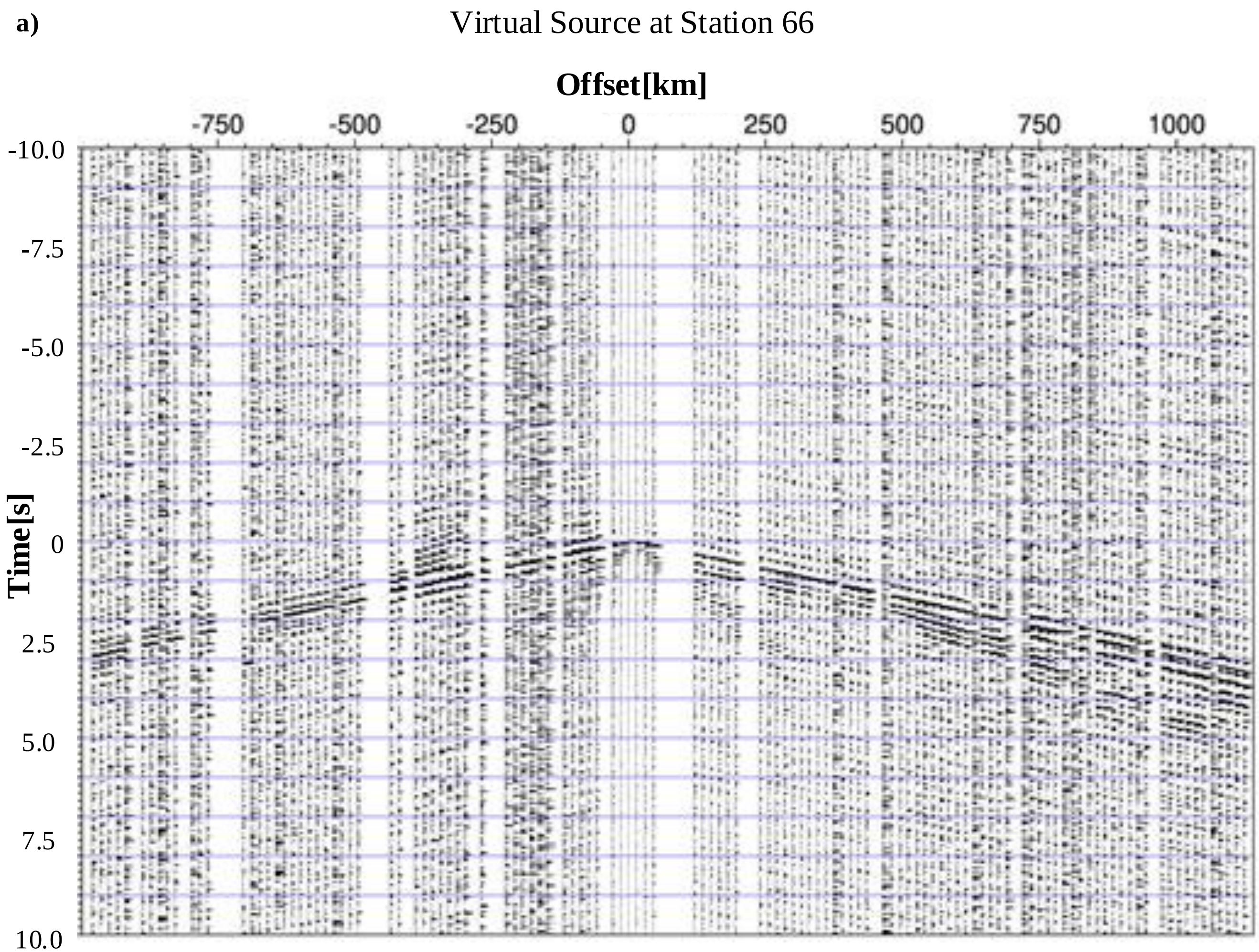


Figure 3.23. a) VSG for station 66 (San Emidio T&E event) shows a first arrival with velocity of ~333 m/s, which is similar to a typical velocity of an air wave.

b)

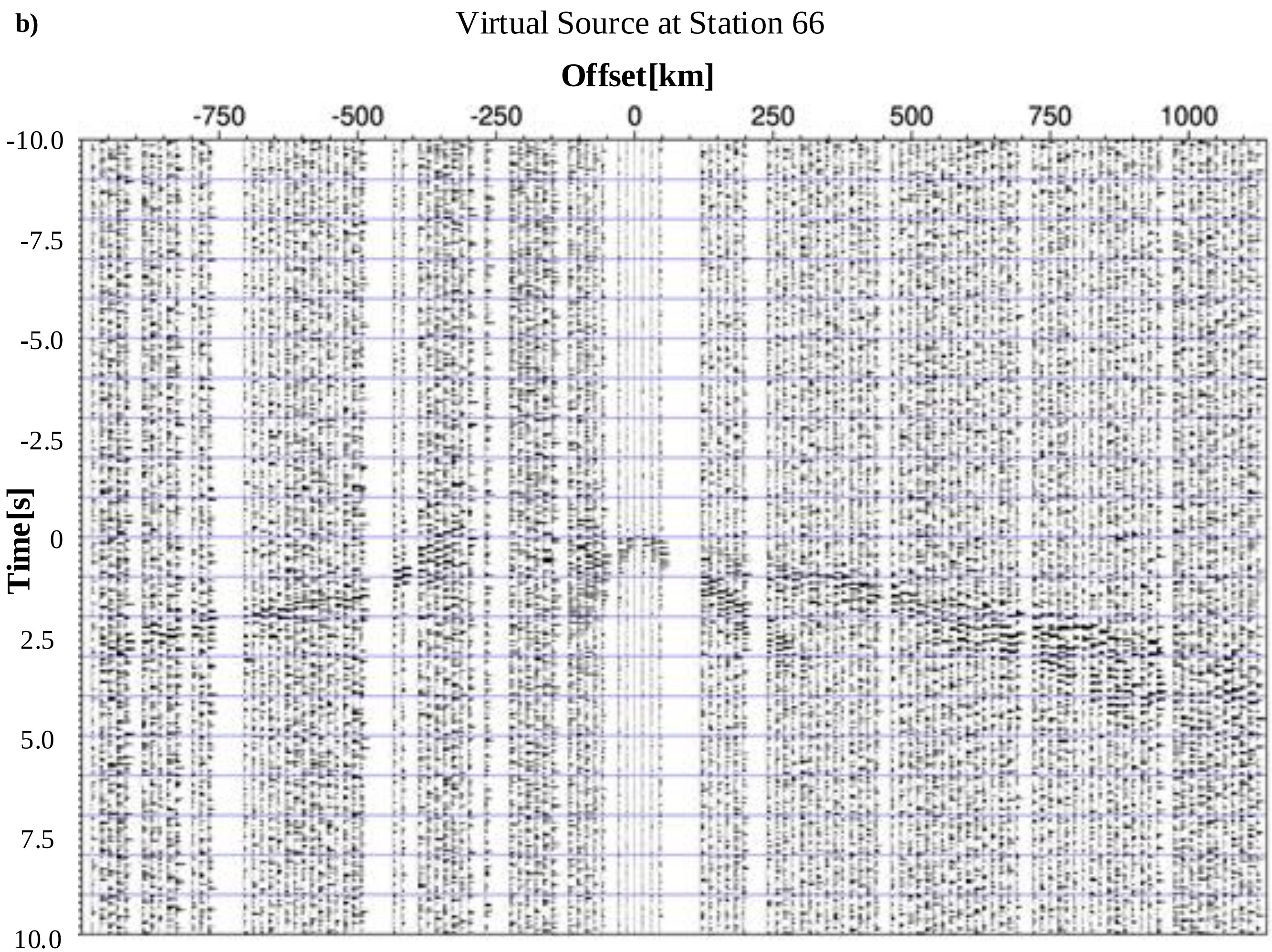


Figure 3.23. b) VSG for station 66 (San Emidio T&E event) after applying a seven-trace median filter to minimize the first arrival with velocity 333 m/s.

Virtual Source at Station 66

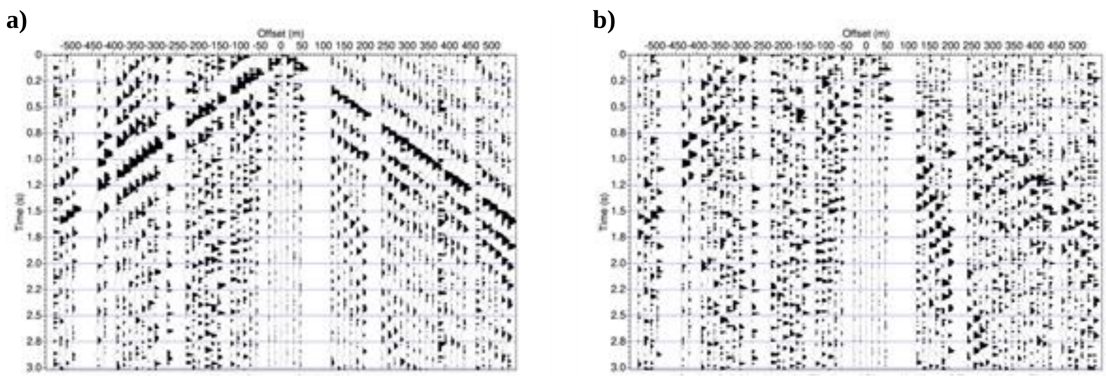


Figure 3.24. VSG for station 66 (San Emidio T&E event): windowed comparison of a) raw VGS for station 66 (left) and b) the median-filtered version (right). Later arrivals, i.e., reflections, are not apparent in b).

Virtual Source at Station 99

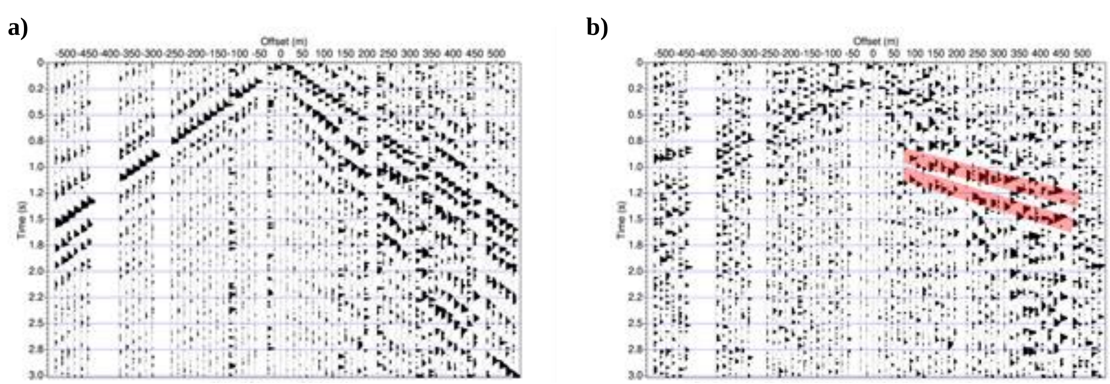


Figure 3.25. VSG for station 99 (San Emidio T&E event:) Windowed comparison of a) raw VSG for station 99 (left) and b) median-filtered version (right). The two transparent lines highlight possible reflections in b).

Discussion

Data acquisition issue. In order to reduce data lag during the San Emidio T&E event, we decided to insert the vertical component of seismic data, at a decimated sample rate of 50 samples/s, into Cassandra. Vertical component data with a maximum frequency of approximately 20-25 Hz was what we needed for near real-time processing of seismic data at “the edge” so a Nyquist frequency of 25 Hz was acceptable. We set the gain setting of the REF TEK 130-01 stream parameters to unity for the San Emidio T&E event. Subsequent, post-event analysis revealed that we should have set the REF TEK 130-01 gain setting to 32x. Our use of geophones, in a relatively low-noise environment, in conjunction with a gain setting of 1x resulted in generally low amplitude signals that complicated seismic processing and reduced sensitivity.

Emplacement issue. We transported the San Emidio T&E event equipment from Texas to Nevada in two 20’ trucks. To conserve space, we stacked empty bins and utilized unassembled node bins (e.g., RaPiER, Tinker, etc.) to store equipment. Although this approach allowed us to minimize the volume of our transportation footprint, it required that we assemble nodes onsite. The assembly of the nodes proved to be labor-intensive and time-consuming; as was the assembly of backhaul telemetry (e.g., antenna masts, enclosures, etc.). Inclement weather further complicated node assembly and reduced our ability to emplace San Emidio T&E event equipment in a timely manner.

Equipment issue. Each seismic station was supposed to include a quick deployment box. Unfortunately, some quick deployment boxes contained broken or were missing components. We were forced to fixed broken components “on the fly” and cannibalize seismic stations to emplace a total of 142 seismic stations.

REF TEK 130-01 GPS issue. On April 6, 2019 many Garmin GPS devices were susceptible to a GPS Week Number Rollover issue; “– a sort of mini Y2K Bug…” that could corrupt GPS devices’ location and time data (Vincent, 2019). Older REF TEK 130-01 GPS receivers (i.e., “humpback”) models had been identified by IRIS PASSCAL as vulnerable to the rollover issue prior to our San Emidio T&E event; however, older version of the “puck” REF TEK 130-01 GPS receivers had not. We were provided new versions of the “puck” REF TEK 130-01 GPS receiver by PASSCAL without out an explanation for the need to replace identical looking older “puck” versions. As such, our initial “wired” test block resulted in datetime corrupted REF TEK 130-01 data. See Fig. A.3 for details.

REF TEK 130-01 backup battery issue. The REF TEK 130-01 uses a backup battery to maintain acquisition parameters when the power is disconnected from the REF TEK 130-01. All REF TEK 130-01 acquisition parameters were set prior to the San Emidio T&E event. Six REF TEK 130-01s had bad backup batteries. Once emplaced, acquisition parameters (i.e., channel and stream) were verified on all REF TEK 130-01s. However, even though the REF TEK 130-01s with bad backup batteries showed properly set acquisition parameters via REF TEK field setup controller (FSC) application, these REF TEK 130-01s created many short-duration RTP files instead of single files of five-minute duration. For REF TEK 130-01s with bad backup batteries it was necessary to delete and reenter acquisition parameters via FSC before they functioned correctly.

Although a relatively easy fix, this issue was difficult to diagnose and required that we repeatedly visit malfunctioning REF TEK 130-01s along the seismic line.

"Wireless" configuration issue. Despite our best efforts to identify a solution that allowed for wireless communication between the Raspberry Pi 3 and Tinker during our San Emidio T&E event's "wireless" configuration, we encountered unexpected problems. Specifically, we were unable to connect the Raspberry Pi 3 via the TP-Link Archer T2UH AC600 Wi-Fi dongle to the Tinker via the EnGenius ENS500EXT-AC. The issue was likely caused by a frequency band mismatch between the TP-Link Wi-Fi dongle and the 3rd party Wi-Fi antenna we chose. We had previously (BRIC T&E event) used the TP-Link Wi-Fi dongle with the 3rd party Wi-Fi antenna (i.e., 802.11n/b/g) but not with the EnGenius ENS500EXT-AC (i.e., 802.11ac).

We attempted to utilize an EnGenius ENS500EXT-AC to enable wireless connectivity for approximately half of squadron #2's Raspberry Pi 3s. However, after approximately six hours into our "wireless" configuration test we began to experience power-related (i.e., low voltage) failures on multiple squadron #2 Cassandra nodes. The power-related failures were due to multiple days of cloud cover negatively impacting the recharging of Cassandra node batteries.

Conclusions and Recommendations

With respect to earlier editions, the T&E event at San Emidio in May 2019 represented a major refinement of solutions (e.g., embedded systems, a distributed database, etc.) and a challenging logistical effort to prepare, mobilize, deploy, and maintain seismic stations, RaPiER nodes, Tinker nodes, and the requisite power and telemetry infrastructure. The emplacement issues, equipment issues, and unplanned troubleshooting, documented above, and weather delays significantly impacted our ability to deploy our solution as planned. Ideally, future efforts should include a simplified assembly and deployment procedure that does not require an in-depth knowledge of RaPiER and Tinker nodes. Also, processes should be incorporated (e.g., checklists, test shots, etc.) that better verify initial status and that gauge subsequent solution performance and the quality of acquired data. Last, additional time to perform adjustment and evaluate results should be budgeted.

Embedded Systems Performance

Over the course of four T&E events, the Raspberry Pi 3 (i.e., RaPiER node's embedded system) performed reliably. During our most recent T&E event (i.e., San Emidio) there were no instances where we failed to capture data due to a Raspberry Pi 3 related failure. Although we increased RaPiER workload and operated the system in environmentally challenging conditions (e.g., temperature, rain, etc.) the Raspberry Pi 3 continued to provide an easy-to-use, inexpensive, and reliable solution to maintain the RTP archive at "the edge."

Our use of the Tinker as our Cassandra node's embedded system presented challenges. The Tinker's user community is not as robust as the Raspberry Pi 3's. Simple problems, such as our USB Wi-Fi dongle issue, were excessively challenging. Although the extra gigabyte of RAM the Tinker provided was necessary for Cassandra to perform reliably, we spent a disproportionate amount of time troubleshooting basic Tinker functionality.

Ideally, a Raspberry Pi 3 with greater than one gigabyte RAM and improved USB/Ethernet throughput would replace the Tinker. Unfortunately, we did not have access to an "enhanced" Raspberry Pi 3 during our four T&E events. However, in June 2019 the Raspberry Pi Foundation announced the release of the Raspberry 4 (Upton, 2019). The Raspberry Pi 4 currently comes in three available RAM configurations (i.e., 2, 4, and 8 GB), offers USB 3.0 support, and Gigabit Ethernet connectivity (Upton, 2020). In the future, we recommend using the Raspberry Pi 4 as the Cassandra node's embedded system instead of the Tinker.

Cassandra (i.e., DSE) and DSE Advanced Replication Performance

Our implementation of Cassandra (i.e., DSE) at "the edge" was relatively straightforward. It was necessary to tune settings given our decision to run DSE on hardware well below DataStax's minimum recommend resources; however, we did not encounter compatibility issues (e.g., processor architecture, Linux, etc.). Our use of DSE Advanced Replication was equally straightforward. We successfully utilized DSE Advanced Replication to implement a hub-and-spoke topology that "hardened" our solution against inevitable network outages.

Our use of DSE and DSE Advanced Replication allowed us to maintain a distributed database at "the edge" that ingested approximately 156 million samples per day (i.e., one squadron) with a mean ingest rate of approximately 1700 rows per second (i.e., "hybrid" configuration). 1700 rows per second was more than double our target minimum ingest rate of 750 rows per seconds. Our ability to query and extract one hour's worth of squadron data (i.e., 36 seismic stations) in approximately ten minutes (i.e., approximately 11,000 samples retrieved per second) allowed for more than enough time to perform our seismic processing hourly.

“Wired configuration”. Approximately 99.7% and 98.9% of the data acquired (i.e., REF TEK 130-01) was ingested into squadron-level Transactional Clusters, squadron #1 and #2 respectively. 100% of the data ingested into the Transactional Clusters (i.e., squadron #1 and #2) was replicated to their respective Query Clusters.

“Hybrid configuration”. Approximately 99.2% and 99.9% of the data acquired (i.e., REF TEK 130-01) was ingested into squadron-level Transactional Clusters, squadron #1 and #2 respectively. Approximately 99.9% and 100% of the data ingested into the Transactional Clusters (i.e., squadron #1 and #2) was replicated to their respective Query Clusters.

“Wireless configuration”. Approximately 85.0% of the data acquired (i.e., REF TEK 130-01) was ingested into squadron-level Transactional Cluster (i.e., squadron #2). However, 99.5% of the data acquired (i.e., REF TEK 130-01) was ingested into headquarters-level Query Cluster (i.e., squadron #2). We observed an anomaly where more data resided within a Query Cluster than a Transactional Cluster. This anomaly resulted in an 8.8% difference between the amount of data residing within squadron #2’s Transactional Cluster and Query Cluster. Although the cause of the anomaly remains unknown, we interpret this as a Cassandra “failure” most likely due to squadron-level Wi-Fi issues and a DSE Advanced Replication “success” via backhaul Wi-Fi telemetry.

Solution Architecture

Our tiered configuration allowed us to successfully create an "articulated" edge storage solution; "articulated" in that our solution consisted of more than two sections (i.e., clusters) connected by a "flexible joint" (i.e., solution architecture and telemetry). Learning lessons over the course of four T&E events, we balanced resources and requirements to deliver an ostensibly endless elastic solution that affords geoscientists the ability to combine data from multiple devices and perform seismic processing at "the edge." Our tiered configuration, in conjunction with Cassandra (i.e., DSE) and DSE Advanced Replication, allows for the implementation of edge, fog, and/or cloud solutions without the need for non-geoscientific software development.

Telemetry Performance

Although we experienced technical issues with our San Emidio T&E event's "wireless" configuration, we believe that use of commercial off the shelf Wi-Fi represents a viable solution for bandwidth intensive geoscience applications. The squadron-level and backhaul Wi-Fi telemetry used during the San Emidio T&E event represented an "out-of-the-box" solution (i.e., IP-based) that integrated seamlessly with embedded systems and Cassandra. Nonetheless, professional-grade wireless solutions do require a level of expertise that goes beyond typical home-use wireless expertise.

Suggestions for Future Work

The REF TEK 130-01 is readily available and in wide use by the geoscience community; however, its end-of-life status motivates us to consider other data acquisition systems. A cableless seismic acquisition system such as the iSeis Sigma may represent a more cost-effective and rugged alternative to the REF TEK 130-01. With iSeis' support, the recompiling Sigma acquisition software for the Raspberry Pi (i.e., ARM architecture) is a relatively straightforward process.

The release of the Raspberry Pi 4 represents a tremendous opportunity. In addition to simplifying the development process, the 4 or 8 GB Raspberry Pi may provide an opportunity to leverage distributed computing at "the edge." DSE advanced functionality includes Apache Spark[8]. Spark is an open source framework that allows for distributed in memory processing across a cluster of machines that includes machine learning, stream processing, and graph analytics libraries (Ryza et al., 2015). The Tinker did not provide enough RAM support both Cassandra and Spark; however, the 4 or 8 GB version of the Raspberry Pi may.

With respect to processing: naively stacking all available data is unlikely to produce convergence to empirical Green's functions as quickly as selectively stacking the "highest quality" VSGs for subsets of the time periods computed. We are actively pursuing strategies for identifying the "panels" of VSGs that carry the most information about later arrivals, ideally reflections. Features that can be identified and extracted in the field could potentially speed up convergence significantly and help deliver the full potential of ambient noise processing at "the edge."

Acknowledgments

The development and testing described here was supported by funding from the U.S. Department of Energy (Energy Efficiency and Renewable Energy Agency,

[8] https://spark.apache.org/

Geothermal Technologies Program, Award Number: DE-EE0007699). Instrumentation and data archiving were provided by the Incorporated Research Institutions for Seismology through their PASSCAL and Data Management Centers, respectively. Both centers provided technical support.

All data and associated metadata from our T&E events at Eastland Lakes, Soda Lake Geothermal Field, the Baylor Research and Industrial Collaborative, and San Emidio Geothermal Field have been archived at the Data Management Center (DMC) operated by the Incorporated Research Institutions for Seismology (IRIS). The open-source algorithms and codes used to derive processed and interpreted seismic results, including detailed descriptions of data processing methodology, are available with MSNoise distribution: www.msnoise.org.

Disclaimer

References

Abramova, V., Bernardino, J., & Furtado, P. (2014). Which NoSQL Database? A Performance Overview, *1*(2), 8.

Carpenter, J., & Hewitt, E. (2016). *Cassandra: the definitive guide* (Second edition). Sebastopol, CA: O’Reilly Media, Inc.

Challen, G., & Welsh, M. (2010). Volcano Monitoring: Addressing Data Quality Through Iterative Deployment. In E. Gaura (Ed.), *Wireless Sensor Networks* (pp. 71–113). New York: Springer.

Chen, C. L. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*(Supplement C), 314–347. https://doi.org/10.1016/j.ins.2014.01.015

Confais, B., Lebre, A., & Parrein, B. (2016). Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 294–301). https://doi.org/10.1109/CloudCom.2016.0055

Cossick, S. (2019, July 16). Here’s what you need to know about a Wi-Fi extender. Retrieved July 31, 2019, from https://www.allconnect.com/blog/3-reasons-house-needs-wi-fi-extender/

Duarte, A., & Bernardino, J. (2016). Cassandra for Internet of Things: An Experimental Evaluation. In *IoTBD*. https://doi.org/10.5220/0005846400490056

Hamilton, E. (2018, December 31). What is Edge Computing: The Network Edge Explained. Retrieved July 16, 2019, from https://www.cloudwards.net/what-is-edge-computing/

Heath, N. (2019, December). How the Raspberry Pi was created: A visual history of the $35 board. Retrieved August 6, 2019, from https://www.techrepublic.com/pictures/how-the-raspberry-pi-was-created-a-visual-history-of-the-35-board/

Ilyas, M., & Mahgoub, I. (Eds.). (2006). *Sensor network protocols*. Boca Raton: CRC/Talor & Francis.

Kalinich, S. (2010, January 4). Wireless Distribution System - Is Wireless ready to replace the Wired network? Retrieved August 4, 2019, from https://www.tweaktown.com/articles/3085/wireless_distribution_system_is_wireless_ready_to_replace_the_wired_network/index2.html

Kelly, G. (2014, December). 802.11ac vs 802.11n WiFi: What's The Difference? Retrieved July 31, 2019, from https://www.forbes.com/sites/gordonkelly/2014/12/30/802-11ac-vs-802-11n-wifi-whats-the-difference/

Koegler, S. (2019, February 7). 3 IoT Edge Storage Considerations [Text]. Retrieved August 7, 2019, from https://www.iotplaybook.com/article/3-iot-edge-storage-considerations-0

Lakshman, A., & Malik, P. (2010). Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, *44*(2), 35. https://doi.org/10.1145/1773912.1773922

Le, T. D., Kim, S. H., Nguyen, M. H., Kim, D., Shin, S. Y., Lee, K. E., & Righi, R. da R. (2014). EPC information services with No-SQL datastore for the Internet of Things. In *2014 IEEE International Conference on RFID (IEEE RFID)* (pp. 47–54). https://doi.org/10.1109/RFID.2014.6810711

Özsu, M. T., & Valduriez, P. (2011). *Principles of distributed database systems* (3rd ed). New York: Springer Science+Business Media.

Ploetz, A., Neeraj, N., & Malepati, T. (2018). *Mastering Apache Cassandra 3.x: an expert guide to improving database scalability and availability without compromising performance*. Retrieved from http://proquest.safaribooksonline.com/9781789131499

Ryza, S., Laserson, U., Owen, S., & Wills, J. (2015). *Advanced analytics with Spark* (First edition). Beijing ; Sebastopol, CA: O'Reilly.

Sepulveda, F., & Pulliam, J. (2016). The Internet of Geophysical Things: Raspberry Pi Enhanced REF TEK (RaPiER) System Integration and Evaluation. *Seismological Research Letters*, *87*(2A), 345–357. https://doi.org/10.1785/0220150234

Upton, E. (2020, May 28). 8GB Raspberry Pi 4 on sale now at $75. Retrieved August 26, 2020, from https://www.raspberrypi.org/blog/8gb-raspberry-pi-4-on-sale-now-at-75/

———. (2019, June). Raspberry Pi 4 on sale now from $35 - Raspberry Pi. Retrieved August 26, 2019, from https://www.raspberrypi.org/blog/raspberry-pi-4-on-sale-now-from-35/

van der Veen, J. S., van der Waaij, B., & Meijer, R. J. (2012). Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 431–438). Honolulu, HI, USA: IEEE. https://doi.org/10.1109/CLOUD.2012.18

Vincent, J. (2019, March 8). Older GPS devices are facing their own mini Y2K bug next month. Retrieved February 2, 2020, from https://www.theverge.com/2019/3/8/18255847/gps-week-rollover-issue-2019-garmin-tomtom-devices-affected

CHAPTER FOUR

The Edge of Exploration: An Edge Storage and Computing Framework for Internet of Things Based Sensor Networks

This chapter submitted as: Frank Sepulveda and Jay Pulliam (in review), The Edge of Exploration: An Edge Storage and Computing Framework for Internet of Things Based Sensor Networks, Sensors.

Abstract

In the past 20 years, technological advances have reduced the complexity and cost of developing sensor networks for remote environmental monitoring. However, the challenges of acquiring, transmitting, storing, and processing remote environmental data remain significant. The transmission of large volumes of sensor data to a centralized location (i.e., the cloud) burdens network resources, introduces latency and jitter, and can ultimately impact user experience. Edge computing has emerged as a paradigm in which substantial storage and computing resources are located at the "edge" of the network. In this paper, we present an edge storage and computing framework leveraging commercially available components organized in a tiered architecture and arranged in a hub-and-spoke topology. The framework includes a popular distributed database to support the acquisition, transmission, storage, and processing of Internet of Things based sensor network data in a field setting. We present details regarding the architecture, distributed database, embedded systems, and topology used to implement an edge-based solution. Lastly, a real-world case study (i.e., seismic) is presented that leverages the edge storage and computing framework to acquire, transmit, store, and process millions of samples of data per hour.

Introduction

The availability of inexpensive low power microcontrollers, sensors, and transceivers in the late 1990s resulted in a flurry of Wireless Sensor Network (WSN) activity in the early 2000s (Corke et al., 2010). By the mid-2000s, WSN experts (i.e., computer and software engineers) were collaborating with geoscientists to deploy WSNs for environmental monitoring (Martinez et al., 2006; Talzi et al., 2007; Werner-Allen et al., 2005). The increased spatial and temporal measurements provided by WSNs were beneficial; however, the design, development, and deployment of WSNs continued to rely heavily on WSN experts (Challen & Welsh, 2010). Commercially available WSN components required considerable modification before they could be deployed in real-world environments (Talzi et al., 2007) and they often experienced reliability problems requiring multiple iterations of design and development before a reliable solution could be delivered (Challen & Welsh, 2010). WSN development was further complicated by interoperability problems resulting from the wide variety of available proprietary and nonproprietary solutions (e.g., hardware, protocols, etc.) (Mainetti et al., 2011).

WSNs and the Internet of Things (IoT) both originated in the late 1990s. The IoT represents a convergence of technologies allowing things (e.g., devices, objects, etc.) to communicate via the Internet (Gaber et al., 2019). Initially, interest in WSNs outpaced the IoT. Google Trend[1] data indicated year-to-year (i.e., 2004 to 2012) interest in WSNs varied between 1.3 to 4.7 times greater than the IoT. In 2013, interest in the IoT overtook WSNs and has increased year-to-year (i.e., 2013 to 2019) monotonically from 1.4 to 14.7 times greater than WSNs. Greengard (2015) credits the 2007 release of the iPhone and

[1] https://trends.google.com/

2010 release of the iPad with increased interest in the IoT. However, we believe the availability of inexpensive and easy-to-use embedded systems with Internet connectivity (e.g., Arduino[2], Gumstix[3], etc.) in the late 2000s contributed to increased IoT development activity.

Internet Protocol (IP) is the principal network layer protocol of the Internet that provides communication among disparate networks (Neves & Rodrigues, 2010). IP was not initially considered suitable for WSNs given the limited computational resources and constrained power budget typical of WSN nodes (Vasseur & Dunkels, 2010). Nonetheless, by 2010, researchers had demonstrated IP-based WSN applications were feasible (Neves & Rodrigues, 2010) and had cataloged numerous examples of IP-based WSNs within industry and the scientific community (Vasseur & Dunkels, 2010). IP-based sensor networks (i.e., wired and wireless) aligned closely with the IoT; thus, resulting in IoT-based sensor networks that reduced overall complexity, promoted interoperability, and increased scalability (Alcaraz et al., 2010; Lazarescu, 2013; Zorzi et al., 2010).

Currently, WSNs are contributing greatly to the IoT by transforming agriculture, healthcare, and industrial automation (Jan et al., 2019). WSNs are considered a basic component of the IoT and the primary means of communication between machines and the future Internet (Chung & Kim, 2016). The continued integration of WSNs and the IoT is expected to result in a significant increase in the number of sensors connected to the Internet (Kocakulak & Butun, 2017). 20 to 30 billion IoT devices are expected to be

[2] https://www.arduino.cc/

[3] https://www.gumstix.com/

connected to the Internet by 2020 (Nordrum, 2016) and countless numbers of sensors will be connected to those devices. The scale and complexity of IoT data, specifically sensor network data, will be unprecedented.

Given the modest resources of IoT devices, IoT data is typically offloaded to the cloud for storage and subsequent processing (Premsankar et al., 2018). Cloud computing (i.e., the cloud) consists of centralized applications offered as services via the Internet and the resources in the cloud provider's data center providing those services (Armbrust et al., 2010). The cloud, with its virtually limitless resources, supports the management of IoT devices as well as the applications and services exploiting IoT data (Botta et al., 2016). However, cloud computing may not be the ideal solution for IoT applications where edge devices (e.g., IoT, mobile, etc.) are major producers data (Garcia Lopez et al., 2015; Shi et al., 2016; Varghese et al., 2016). The transmission of large volumes of edge device data to the cloud burdens network resources, introduces latency and jitter, and ultimately impacts user experience (Satyanarayanan, 2017). Moreover, excessive backhaul network traffic to the cloud negatively impacts the performance and survivability of edge devices by increasing power consumption, introducing a single-point-of-failure, and wasting edge device computing resources (Garcia Lopez et al., 2015; Satyanarayanan, 2017; Shi et al., 2016; Varghese et al., 2016).

Edge computing represents an emerging paradigm where substantial storage and computing resources are placed at the edge of the network (Satyanarayanan, 2017; Yousefpour et al., 2019). The "edge" is the local network typically one-hop away from an edge device (Yousefpour et al., 2019). The adage "compute is cheap, storage is cheaper, but data movement is very expensive" (Morgan, 2018) and the fact that edge device performance enhancements have outpaced network performance enhancements (Shi et al., 2016) illustrate the motivation to move storage and computing resources to the edge. Edge computing allows for the better control of data (e.g., privacy, security, etc.), enhanced application performance (e.g., jitter, latency, etc.), increased scalability (e.g., data aggregation, preprocessing, etc.) and improved survivability (e.g., connectivity, reduced power consumption, etc.) (Satyanarayanan, 2017).

Within the current IoT landscape, edge computing is considered a critical computing paradigm (Yousefpour et al., 2019). Edge computing is particularly useful to IoT applications where: 1) low latency is required, 2) connectivity is constrained (i.e., network capacity) or nonexistent, or 3) dense acquisition of, relatively high sample rate, data is occurring (Premsankar et al., 2018; Yousefpour et al., 2019). IoT applications utilizing IoT-based sensor networks to perform remote environmental monitoring (i.e., seismic) typically acquire relatively high sample rate data (i.e., 10s or 100s of samples per second, sps), from one, tens, hundreds, or even thousands of sensors, in locations where connectivity is either constrained or nonexistent.

As geoscientists, we intend to mitigate performance limiters commonly encountered when deploying IoT-based sensor networks for seismic monitoring by utilizing edge computing to collectively process data “in a field setting” without disrupting acquisition and regularly assess the quality of our deployment strategy and results. Essentially, we aim to reduce the cost and risk typically associated with seismic monitoring.

In this paper, we present an edge storage and computing framework for IoT-based sensor networks. The framework uses common embedded systems (i.e., Raspberry Pi[4] and Tinker Board[5]) and IP-based networks to orchestrate general purpose, edge-based, computing services using a popular distributed database (i.e., Apache Cassandra). Our goal was to utilize this framework to automate the acquisition, transmission, storage, and processing of seismic data, in a field setting. The main contributions of this paper are: 1) an architecture and topology supporting IoT-based sensor network edge storage and computing, 2) the selection and review of a distributed database that complements the architecture and topology, 3) recommendations regarding embedded systems to support the acquisition, storage, and processing of sensor data, and 4) details regarding a real-world remote environmental monitoring (i.e., seismic) case study in which approximately 13 million samples were acquired, transmitted, stored, and processed hourly. In a field setting, greater than 99% of the data acquired by edge devices (i.e., seismic stations) was stored, queried, and extracted from edge nodes for seismic processing.

Motivation

There are myriad examples in which WSNs have been proposed to replace cable-based connectivity for seismic monitoring applications (S. Savazzi et al., 2011; Stefano Savazzi, Goratti, et al., 2009; Stefano Savazzi, Rampa, et al., 2009); however, these efforts largely focus on wireless technology itself (e.g., protocols, specification, etc.). In 2018, Jamali-Rad and Campman (2018) proposed a wireless sensing framework, that

[4] https://www.raspberrypi.org/

[5] https://www.asus.com/us/Single-Board-Computer/Tinker-Board/

utilized low-power wide-area network (LPWAN), to prioritize: 1) inherently IoT-compatible, low power, and long range wireless sensors, 2) scalable advanced wireless networking protocols, and 3) cloud storage and computing. The static context header compression initiative (SCHC) for IoT interoperability further strengthens the viability of LPWAN of IoT applications (Sanchez-Gomez et al., 2020). SCHC is a novel compression and fragmentation scheme for transmitting IPv6/UDP packets over LPWANs (Toutain et al., 2018.).

The wireless sensing framework proposed by Jamali-Rad and Campman (2018) and Jamali-Rad et al. (2018) relied upon a cloud paradigm (i.e., a centralized model) for remote data storage and analysis. This centralized model required that acceptable latency, data transmission rates, and data generation rates were considered when identifying applicable scenarios of interest (Jamali-Rad et al., 2018; Jamali-Rad & Campman, 2018). Jamali-Rad and Campman (2018) identified four scenarios of interest (i.e., triggered and/or continuous monitoring): 1) ground motion monitoring, 2) ambient-noise seismic interferometry, 3) microseismic fracture monitoring, and 4) quality control for active land seismic surveys. However, continuous monitoring applications required that an appropriate wireless network was available (Jamali-Rad & Campman, 2018).

Valero et al. (2019) and Clemente et al. (2020) propose an in situ signal processing approach that leverages IoT technologies to develop a real-time system for performing seismic analytics within the sensor network. This approach is ideal for scenarios in which a centralized model is untenable due to constrained or nonexistent backhaul connectivity (Valero et al., 2019). Valero et al. (2019) and Clemente et al. (2020) leverage their respective solutions to successfully perform autonomous, in situ, seismic imaging for thirteen nodes located a few meters apart and six nodes located approximately 15 meters apart, respectively. Valero et al. (2019) and Clemente et al. (2020) both use MySQL[6] to store data on individual sensor network nodes.

The challenges of acquiring, transmitting, storing, and processing seismic data are non-trivial. The seismic methods used extensively in the oil and gas industry are costly and time consuming; seismic surveys require operators to assume substantial cost and risk (Jones, 2018). Likewise, seismic methods employed within the scientific community are typically costly and time consuming. The outlay costs for a single transportable array broadband seismic station (i.e., USArray[7]) was between $30,000 to $50,000 (USD) (“USArray - Adopt a Station - Lower 48,” n.d.). The utilization of edge storage and computing to reduce the cost and mitigate the risk typically associated with seismic methods could have a profound impact on the oil and gas industry and the scientific community. The edge storage and computing framework described below could also prove to be particularly beneficial to the emerging Industrial Internet of Things (IIoT) or other sensor-heavy IoT applications.

Ongoing Information and Communication Technology (ICT) development has resulted in the availability of increased computing resources and widespread connectivity enabling scientists and engineers to streamline research and create practical solutions to real-world problems (Fratu et al., 2016). In 2016, we integrated a commercially available geoscience-related digitizer (i.e., REF TEK 130-01) with an inexpensive and easy-to-use embedded system (i.e., Raspberry Pi) to provide the Raspberry Pi Enhanced REF TEK

[6] https://www.mysql.com/

[7] http://www.usarray.org/

(RaPiER) platform (Sepulveda & Pulliam, 2016). The RaPiER proved to be an effective single-node edge-based solution; however, more complex analysis requires data from multiple nodes to be processed collectively. We built upon our previous effort and utilized easy-to-use and well-established (i.e., within the ICT community) components to develop a novel edge-based solution capable of scaling to hundreds of nodes deployed over thousands of meters.

Framework Overview

Background

We planned to acquire data from approximately 150 seismic stations (i.e., digitizers and sensors) spaced evenly along a line slightly more than two kilometers in length. Each seismic station would acquire 250 sps data from three channels (i.e., a tri-axis geophone); however, only one channel (i.e., the vertical), downsampled to 50 sps, would be processed. It would therefore be necessary to acquire, transmit, store, and collectively process approximately 650 million data samples per day, in a field setting. At 24 bits per sample, this would result in approximately 1.8 gigabytes (GB) of data generated per day. The seismic stations would be deployed in a remote environment, without permanent support infrastructure (e.g., communication, power, etc.), for approximately one week. We intended to utilize commercially available communications infrastructure, digitizers, a distributed database, and embedded systems to minimize the cost and complexity of implementing the edge-based solution described here.

Given the requirements described above, we developed an edge-based solution relying upon an IoT-based sensor network to accomplish our goals as geoscientists. Over the course of multiple deployments, we developed a tiered architecture of embedded systems, arranged in a hub-and-spoke topology, hosting a distributed database allowing for the acquisition, transmission, storage, and hourly processing of seismic data. Thus, allowing for the adjustment, if necessary, of the configuration (e.g., acquisition parameters, geometry, etc.) and modification (i.e., the shortening or lengthening) of the duration of our deployment with high levels of confidence our goals had been achieved. Details regarding our deployment will be provided in the Case Study section of this paper.

The collective processing of sensor network data, at the edge of the network, reduces the cost of individual sensor nodes, increases fault tolerance, and promotes flexible configuration and management of shared sensor network resources (i.e., communication, storage, and computational); however, these resources must be capable of handling the velocity, volume, etc. of sensor network data (Ilyas & Mahgoub, 2006). It is necessary to implement an edge-based solution where the design and arrangement of communication, storage, and computational resources support the processing of sensor network data and mitigate the inevitable network connectivity problems commonly encountered during remote environmental monitoring. The following subsections present information regarding the selection of an appropriate architecture, topology, distributed database, embedded systems, and communication infrastructure to support the edge storage and computing framework.

Architecture

Background. In an effort to maximize energy efficiency, sensor networks in the early 2000s adopted an architectural design that assumed it would be necessary to store and process data, as close to the data source as possible, on nodes with modest resources (Gnawali et al., 2006). This architectural design featured an application-specific and data-centric approach where the sensor networks were customized for specific applications and data was decoupled from the sensors (i.e., nodes) producing it (Estrin et al., 1999). Essentially, an egalitarian collection of sensor nodes, located within an immediate vicinity of each other, coordinate to achieve high-level objectives (Estrin et al., 1999).

Tenet principle. Although this approach was widely adopted, Gnawali et al. (2006) believed it increased system complexity and decreased manageability. Gnawali et al. (2006) expected future large-scale sensor networks would be tiered (i.e., lower and upper). The lower tier would consist of many constrained sensor nodes and the upper tier would consist of fewer less-constrained nodes (Gnawali et al., 2006). The upper tier reduced complexity and increased manageability via the restriction of multi-node storage and processing to the upper tier (i.e., the Tenet principle) (Gnawali et al., 2006). The restriction of multi-node storage and processing to the upper tier could introduce a single-point-of-failure or be less energy efficient (Estrin et al., 1999). Nonetheless, the Tenet architectural principle complements our desire to minimize the complexity of integrating commercially available components into an overarching edge storage and computing framework.

Proposed architecture. Reference Architectures (RA), such as INTEL-SAP RA, Edge Computing RA 2.0, etc., were developed to establish standards regarding the design of edge computing architectures and their integration with ICT (Sittón-Candanedo et al., 2019). Edge computing RA are typically based upon a three-layer model including cloud services as the upper layer (Sittón-Candanedo et al., 2019). Fig 4.1 illustrates a generic edge computing reference architecture.

We utilized general purpose embedded systems to implement an architecture consisting of three, Tenet architectural principle inspired, tiers (i.e., lower, middle, and upper); however, our tiers (i.e., layers) are defined by workload. The complexity of the workload and, in turn, embedded systems (i.e., hardware and software) increases from the lower to upper layers. The lower layer is responsible for the "sensing workload," the middle layer maintains the "transactional workload," and the upper layer supports an "analytic workload." See Fig. 4.2 for our edge storage and computing architecture. The sensing workload consists of edge devices (i.e., digitizers and sensors) and edge gateways (i.e., lower layer embedded systems) responsible for the acquisition of raw sensor data, the pre-processing of sensor data, and its subsequent insertion into middle layer edge nodes. Middle layer edge nodes form a distributed database that stores sensor data from multiple edge devices and replicates the data to the upper layer edge nodes. The upper layer edge nodes form a distributed database that stores sensor data from multiple middle edge nodes. Sensor data within the upper layer edge nodes can be queried and extracted locally for analysis or it can be replicated to other locations (e.g., cloud, edge, etc.).

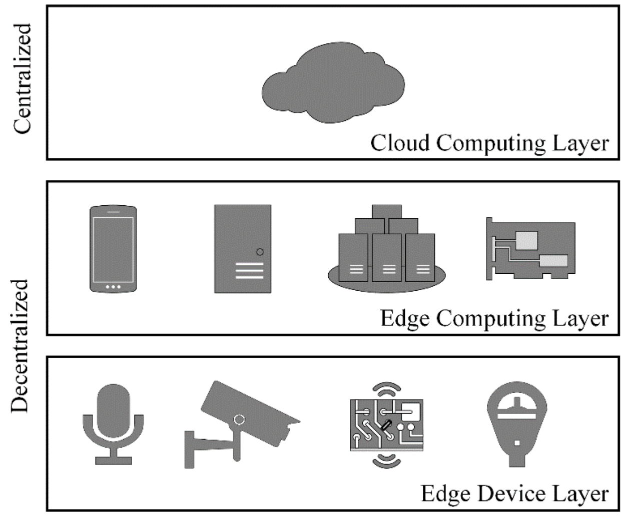


Figure 4.1. Generic edge computing architecture.

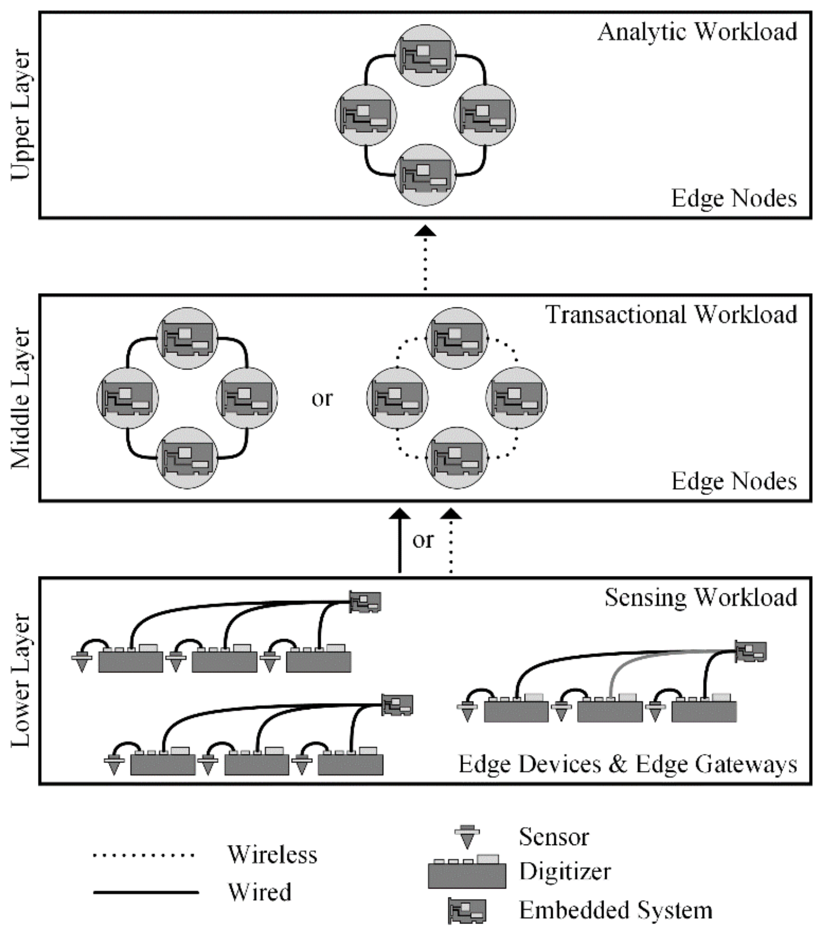


Figure 4.2. Proposed edge computing architecture.

Topology

Background. The distance sensor network data traverses (i.e., wired and wireless) varies from a few meters to thousands of kilometers. Network delay, errors, etc. could negatively impact the quality and timeliness of sensor network performance (Ilyas & Mahgoub, 2006). Considering the inevitable network connectivity problems commonly encountered during remote environmental monitoring, it is necessary to arrange communication, storage, and computational resources in a manner ameliorating the negative effects of data delay, loss, etc. in the collective processing of sensor network data (Ilyas & Mahgoub, 2006).

Hub-and-spoke. In communications networks, the hub-and-spoke topology consists of nodes (i.e., spokes) connected to centralized hubs acting as switching points for network traffic (Klincewicz, 1998). Hubs are interconnected with other hubs via backbone (i.e., backhaul) networks typically carrying larger volumes of network traffic compared to hub-to-spoke network connections (Klincewicz, 1998). The hub-and-spoke topology is commonly used for computer, military, and telecommunication applications (Karatas & Onggo, 2019). Sensor networks often adopt a hub-and-spoke topology to improve system performance by efficiently routing traffic between specific sources and destinations (Karatas & Onggo, 2019).

We adopted a hub-and-spoke topology, consisting of wired and wireless networks, to facilitate the concentration of sensor network data from the lower to upper layers of our edge storage and computing architecture. Fig. 4.3 illustrates the specific hub-and-spoke network (i.e., a tree/star network) used. In the tree/star network, nodes are connected to a hub (i.e., a concentrator) that is, in turn, connected to a central location or other to another intermediary concentrator, in a hierarchical structure (Klincewicz, 1998). Our choice of the tree/star network was influenced by the following three factors: 1) the hierarchical structure allows for the use of concentrators with greater capability (e.g., memory, storage, etc.) as they progress upward in the tree (Klincewicz, 1998), 2) the limits (e.g., network capacity, storage and processing capabilities, etc.) of concentrators can be overcome by adding additional concentrators and redistributing nodes accordingly, and 3) the tree/star networks allows for the continued addition of nodes (i.e., scaling), provided sufficient backhaul network capacity.

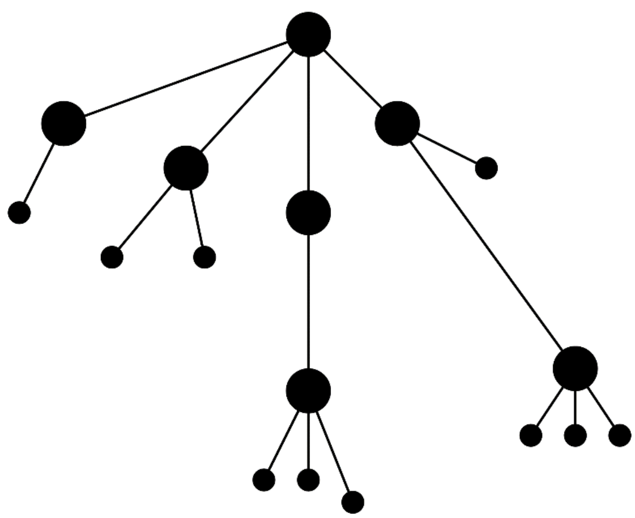


Figure 4.3. An example of a tree/star network.

Distributed Database

Background. A structured collection of data, relating to some modeled real-world phenomena, is known as a database (Özsu & Valduriez, 2011). If the database structure (i.e., model) takes the form tables, it is known as a relational database (Özsu & Valduriez, 2011). The relational model has been used to develop most conventional distributed database technology (Özsu & Valduriez, 2011). A collection of multiple, logically interrelated, databases distributed over a network is known as a distributed database (Özsu & Valduriez, 2011). Distributed database management system (DBMS) is the software used to obfuscate the complexity of distributed data storage and allow for the management of the distributed database (Özsu & Valduriez, 2011). Like DBMS, a relational database management system (RDBMS) afford similar functionality to users. Microsoft Access, MySQL, and Oracle are examples of RDBMS with which readers may be familiar.

As data volumes increase, RDBMS administrators have two available scaling options: 1) the distribution of data across more machines (i.e., horizontal scaling) or 2) increasing the system performance of the existing machine (i.e., vertical scaling) (Carpenter & Hewitt, 2016). Vertical scaling is simple to implement; however, it may not be the most effective scaling method given cost and technology limitations. Horizontal scaling uses relatively inexpensive commodity hardware to distribute the database across multiple systems, thus reducing the overall workload of individual systems. Unfortunately, a distributed RDBMS results in distributed transactions. This requires the implementation of a two-phase commit to prevent new transactions from executing until the prior transaction is complete and a commit response has been returned to the transaction manager (Carpenter & Hewitt, 2016). As the number of transactions (i.e., data velocity) and duration of transaction processing time (i.e., data volume) increase, the RDBMS will likely encounter performance problems resulting from the way RDBMS inherently operate (Carpenter & Hewitt, 2016; Chen & Zhang, 2014).

In 2016, we conducted a literature review to identify RDBMS (i.e., SQL) alternatives ideally suited for remote environmental monitoring applications (Abramova et al., 2014; Confais et al., 2016; Duarte & Bernardino, 2016; Le et al., 2014; van der Veen et al., 2012). Given our need to store and process 100s of millions of samples per day, a "Not only SQL" (i.e., NoSQL) database, specifically Apache Cassandra[8], emerged as our database of choice. Initially created by Facebook to solve their Inbox Search problem, Cassandra leveraged Amazon's Dynamo and Google's Bigtable to meet challenging write-heavy (i.e., billions per day), geographically distributed, reliability, and scalability requirements (Lakshman & Malik, 2010). Cassandra, accepted as an Apache Software Foundation (ASF) top level project in February 2010, is an open source, distributed, decentralized, multi-location (e.g., cloud, on-premises, etc.), operationally simple, nearly linearly scalable (i.e., horizontally scalable), highly available, fault-tolerant, wide-column database (Carpenter & Hewitt, 2016; Ploetz et al., 2018).

CAP theorem. To better illustrate the differences between SQL and NOSQL (i.e., Cassandra) we will elaborate on the Consistency, Availability, and Partition tolerance (CAP) theorem (Carpenter & Hewitt, 2016). In 2000, Eric Brewer conceived that there are three, mutually dependent, requirements present within large-scale distributed systems: consistency, availability, and partition tolerance (Carpenter & Hewitt, 2016). Consistency means each node in the system returns the "correct" response, availability necessitates each request eventually receives a response, and partition tolerance requires the distributed system continue to function even when faulty connectivity has partitioned

[8] https://cassandra.apache.org/

the network (Gilbert & Lynch, 2012). CAP theorem – sometimes referred to as Brewer's theorem – states it is only possible to strongly support two of the three requirements at a time (Carpenter & Hewitt, 2016). The CAP theorem was formally proved to be true by Gilbert and Lynch (2002). Fig. 4.4 was inspired by a graphic presented by Carpenter and Hewitt (2016) illustrating where a variety of datastores align along the CAP continuum. Relational databases (e.g., MySQL, SQL Server, etc.) prioritize availability and consistency and Cassandra prioritizes availability and partition tolerance (Carpenter & Hewitt, 2016).

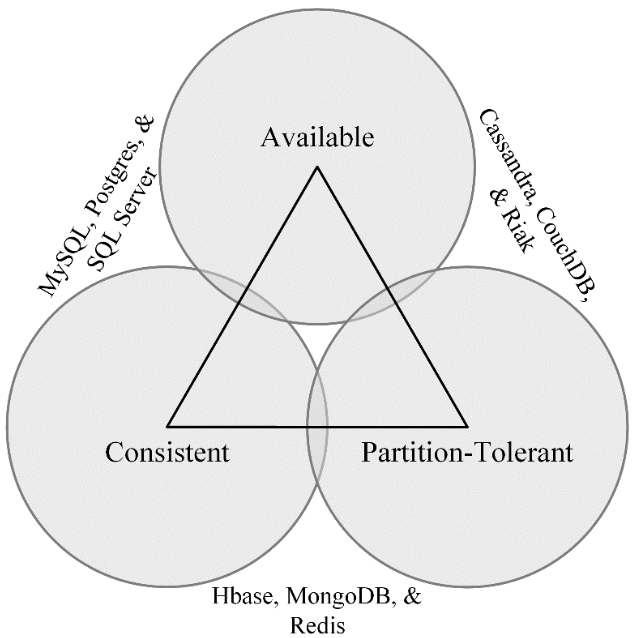


Figure 4.4. CAP Theorem with examples of datastores positioned along CAP continuum.

In 2012, Brewer provided an updated perspective maintaining that CAP theorem's "2 of 3" is misleading because 1) partitions are uncommon, 2) low level choices between availability and consistency occur often, and 3) availability, consistency, and partition tolerance are continuous rather than binary (Brewer, 2012). Brewer's update is germane to enterprise-grade solutions including robust network infrastructure, servers, etc. However, we believe edge-based solutions running on extremely modest hardware, regularly encountering network connectivity problems, require a database with architectural pillars (i.e., mechanisms) supporting a bottom-up approach to partition tolerance. Moreover, edge-based solutions may benefit from a more nuanced approach to partitioning (e.g., data, operational, etc.) and tunable consistency that may significantly improve the solution's robustness to major network connectivity problems without immediately compromising availability (Gilbert & Lynch, 2012).

Apache Cassandra – physical architecture. According to Carpenter and Hewitt (2016), a collection of Cassandra nodes managing a dataset are known as a cluster. A Cassandra cluster is composed of nodes, i.e., a single instance of Cassandra running on a computer, and one or more data centers; a Cassandra data center (DC) is a logical set of nodes, connected via a reliable network, that are relatively close to each other (Carpenter & Hewitt, 2016). Fig. 4.5 illustrates a Cassandra cluster consisting of two DCs, each with four Cassandra nodes. Cassandra clusters can consist of multiple DCs, often geographically distributed, containing one or more Cassandra nodes (Carpenter & Hewitt, 2016). However, a minimum of four Cassandra nodes are typically required to realize the advantages of Cassandra as a distributed database. Refer to Carpenter and Hewitt (2016) for additional information regarding the physical architecture of Cassandra.

Apache Cassandra – ring. The data managed by a Cassandra cluster is known as a ring; each node comprising the ring is assigned a range of data known as its token range (Carpenter & Hewitt, 2016) (see Fig. 4.6). An individual token within a Cassandra node's token range is identified by a 64-bit integer that represents a partition within the ring (Carpenter & Hewitt, 2016). A Cassandra cluster's tokens therefore span the range -2^{63} to $2^{63}-1$ (Carpenter & Hewitt, 2016). When data is written to Cassandra, a hashing function (i.e., a partitioner) determines the data's token value based upon its partition key (Carpenter & Hewitt, 2016). The data's token value is compared with Cassandra nodes' token ranges, its owner-node is identified, and the data is written to the appropriate partition (Carpenter & Hewitt, 2016). Cassandra is able to write data to disk quickly because its design does not require disk reads or seeks (Carpenter & Hewitt, 2016). Essentially, Cassandra writes data to, likewise reads data from, disk sequentially according to the data's partition key; this design is particularly advantageous when working with time series data that has been partitioned (i.e., bucketed) according to anticipated access patterns (e.g., hourly, daily, etc.). For detailed information regarding Casandra's ring or write path refer to Carpenter and Hewitt (2016).

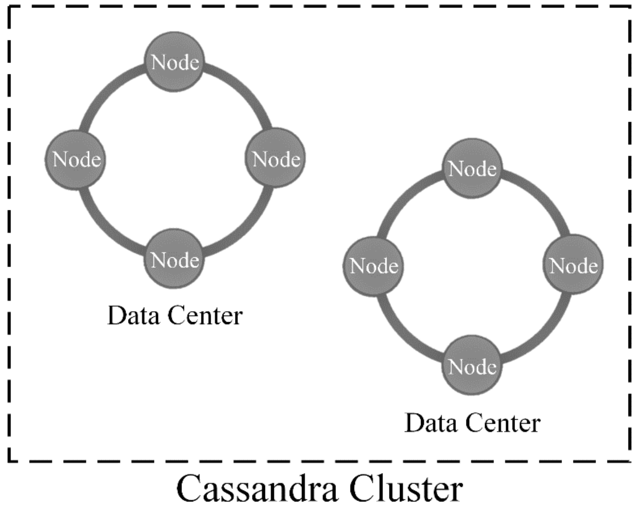


Figure 4.5. Cassandra cluster, with two DCs, and four Cassandra nodes each.

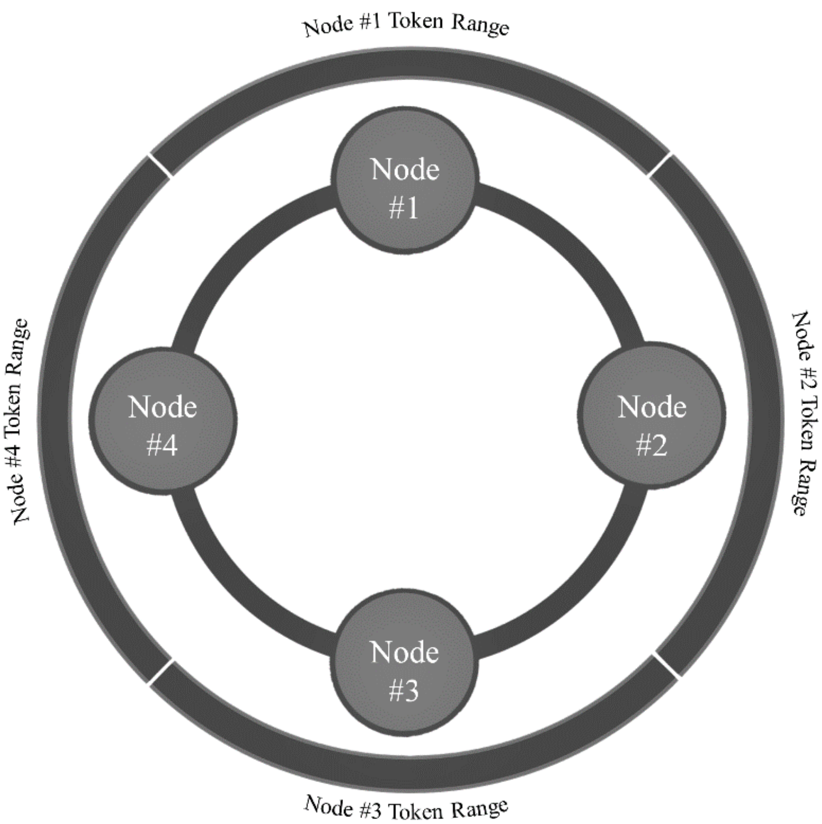


Figure 4.6. Cassandra cluster and ring shown with four Cassandra nodes and their respective token ranges.

Apache Cassandra – replication and consistency. In Cassandra, the database object controlling the replication of data to one or more nodes or DCs within a Cassandra cluster is known as the keyspace; the user defined parameter (i.e., replication factor) determining how data is replicated across Cassandra nodes and DCs is specified in the keyspace (Carpenter & Hewitt, 2016). Read queries or write operations in Cassandra include a user defined consistency level specifying how many nodes must respond before a read or write is considered successfully completed (Carpenter & Hewitt, 2016). Together, replication factor and consistency level allow for tunable consistency supporting Cassandra's prioritization of availability and partition tolerance over the "all or nothing" approach of strict consistency (Carpenter & Hewitt, 2016). It is important to note that any Cassandra node (i.e., coordinator node) or client connected to a coordinator node can coordinate a read or write operation; the coordinator node determines which Cassandra node or nodes own the data (i.e., replicas) and forwards the read or write request accordingly (Carpenter & Hewitt, 2016).

Apache Cassandra – mechanisms. Anti-entropy, gossip, etc. are some of the architectural pillars (i.e., mechanisms) supporting Cassandra's decentralized distributed operations. A review of all these mechanisms is beyond the scope of the current discussion; however, it is important to note that some of these mechanisms are considered essential for decentralized edge-based sensor network solutions in general (Kamath et al., 2016). Below, we will provide a brief introduction to Cassandra's commit log, hinted handoff, gossip protocol, and snitch mechanisms. Our selection of Cassandra for our edge-based solution's distributed database was heavily influenced by its use of these mechanisms. For additional information regarding these mechanisms refer to Carpenter and Hewitt (2016).

When a write operation occurs, Cassandra immediately writes the data to a commit log (i.e., to disk); the commit log is a mechanism supporting Cassandra's durability via crash-recovery (Carpenter & Hewitt, 2016). A write operation is not considered successful unless it is written to the commit log (Carpenter & Hewitt, 2016). If a Cassandra node crashes, the commit log is replayed in order to ensure data is not lost (Carpenter & Hewitt, 2016). If a write operation is sent to a coordinator node and the Cassandra node owning the partition corresponding to the data's partition key is unavailable, Cassandra implements the hinted handoff mechanism (Carpenter & Hewitt, 2016). Hints are saved on the coordinator node and are sent via hinted handoff once the replica node or nodes are back online (Carpenter & Hewitt, 2016). Cassandra utilizes a gossiping protocol to exchange endpoint state information amongst Cassandra nodes (Carpenter & Hewitt, 2016). In addition to the gossip protocol, Cassandra also implements a snitch to gather network topology information; Cassandra uses this information to efficiently route read and write operations by determining the relative proximity of Cassandra nodes (Carpenter & Hewitt, 2016).

Apache Cassandra – DataStax Enterprise. Initially, we planned to replicate data from the middle to upper layer of our architecture by deploying a single Cassandra cluster consisting of two DCs (i.e., a transactional DC and an analytic DC). Our first real-world field deployment (June 2017) consisted of a single Cassandra cluster with a 20 Cassandra node transactional DC and a three Cassandra node analytic DC. Unfortunately, intermittent network connectivity (i.e., wireless backhaul network) between the two DCs resulted in Cassandra nodes being deprecated due to unanswered topological gossip state updates. Ultimately, this resulted in a loss of data replication at the Cluster and DC levels. In order to overcome the real-world network connectivity problems commonly encountered during remote environmental monitoring, we needed a solution allowing for cluster-to-cluster (i.e., middle-to-upper layer) replication that was tolerant of faulty backhaul network connectivity.

For subsequent field deployments, we transitioned from Cassandra (i.e., DataStax Community Edition) to DataStax Enterprise[9] (DSE). DSE is an enterprise-grade version of Cassandra providing commercial confidence and extra capabilities such as automatic management services, advanced security, and advanced functionality. Our primary interest in DSE's advanced functionality was DSE Advanced Replication[10]. DSE Advanced Replication supports the configurable replication of data from source to destination clusters in a manner tolerate of the intermittent loss of backhaul network connectivity. DSE Advanced Replication allows for the configuration of automatic failover, permits, and priority to manage traffic between clusters ("Traffic between the clusters | DSE 6.0 Admin guide," n.d.). Using DSE Advanced Replication, we transitioned from a single cluster with two DCs to a direct cluster-to-cluster implementation. DSE Advanced Replication could be configured to further extend our infrastructure to support additional one-to-one or many-to-one (i.e., cluster(s)-to-cluster)

[9] https://www.datastax.com/products/datastax-enterprise

[10] https://docs.datastax.com/en/dse/6.0/dse-admin/datastax_enterprise/advReplication/advRepTOC.html

implementations. Although not open-source, DSE grants customers a limited no-fee license[11] for non-production purposes, without the right to support.

Apache Cassandra – summary. There are three key takeaways regarding the use of Cassandra (i.e., DSE) as the edge storage solution for our IoT-based sensor network: 1) DSE is ideally suited for time series data because of its sequential (i.e., from disk) read and write operations, 2) mechanisms, such as commit log, gossip, hinted handoff, and snitches, allow DSE to support high availability, fault-tolerant, and geographically distributed implementations, and 3) the shared-nothing architecture of DSE, when coupled with DSE Advanced Replication, enables nearly linear horizonal scalability for our edge storage and computing framework.

Embedded Systems

Background. Embedded systems include, but are not limited to, microcontrollers, embedded computers, system-on-chip, computer-on-module, and system-on-module. Typically, embedded systems are inexpensive, low power, small, and have modest capabilities when compared with desktop or laptop computers. The Raspberry Pi[12] is among the most popular embedded systems. As of December 2018, the Raspberry Pi was the world's third best-selling general purpose computer (Heath, 2019). As a system originally intended to teach children computer science, the Raspberry Pi is inherently easy-to-use and inexpensive (Heath, 2019). Having developed an immense community of

[11] https://www.datastax.com/legal/datastax-enterprise-terms

[12] https://www.raspberrypi.org/

users, Raspberry Pi based industrial and scientific projects are commonplace (Deshmukh & Shinde, 2016; Kumar & Rajasekaran, 2016; Merchant & Ahire, 2017).

Our initial selection of the Raspberry Pi was influenced by the Raspberry Pi's vast community of users and its widespread use within the industrial and scientific communities. We began our development of a multi-node edge-based solution in October of 2016. At the time, the Raspberry Pi 3 B (i.e., 1.2 GHz 64-bit quad core processor with 1 GB of RAM) was available. We installed and configured DSE on the Raspberry Pi 3 B; however, the modest resources of the Raspberry Pi 3 B resulted in frequent downtime (e.g., hangs, reboots, etc.).

In order to improve reliability, we offloaded the Raspberry Pi's DSE workload by replacing the Raspberry Pi 3 B with the Asus Tinker Board[13]. The Tinker is like the Raspberry Pi 3 B, with an additional gigabyte of RAM (i.e., 2 GB of RAM total). We found the additional gigabyte of RAM significantly improved DSE performance and reliability. Although the Tinker performed well as a DSE node, its user community is not as large as the Raspberry Pi's. We spent a disproportionate amount of time configuring the Tinker due to its relatively limited support (e.g., drivers, examples, etc.).

In June 2019, the Raspberry Pi Foundation announced the release of the Raspberry 4. The Raspberry Pi 4 comes in three available configurations (i.e., with 1 GB, 2 GB, and 4 GB of RAM), offers USB 3.0 support, and Gigabit Ethernet connectivity (Upton, 2019). We recently bench tested the Raspberry Pi 4 (i.e., with 2 GB of RAM) and confirmed DSE performance and reliability was equivalent to the Tinker's; however, we have not had an opportunity to test the Raspberry 4 in a field setting. Any future

[13] https://www.asus.com/us/Single-Board-Computer/Tinker-Board/

efforts on our part would utilize the Raspberry Pi 4 (i.e., with 2GB or 4 GB of RAM) as our DSE node's embedded system.

Related work. Cassandra and DSE were established cloud and on-premises NoSQL solution in 2016; however, to the best of our knowledge no one had attempted to deploy Cassandra or DSE, on an embedded system, as an edge-based storage and computing solution supporting remote environmental monitoring. Nonetheless, there have been several publications since 2016 exploring the idea of utilizing the Raspberry Pi and Cassandra for IoT applications (Ferencz & Domokos, 2018; Richardson, 2017; Romero López, 2017). In 2017, Richardson (2017) explored the feasibility of using the Raspberry Pi to host Cassandra in support of IoT applications. Richardson (2017) utilized the Raspberry Pi (i.e., with 1 GB of RAM) and virtual machines (i.e., with 1 GB, 2 GB, and 4 GB of RAM) to assess the feasibility and performance impact of hosting Cassandra on modest platforms; a minimum of 2 GB of RAM was identified by Richardson (2017) as critical for "in-situ IoT storage" using Cassandra. Also in 2017, Romero Lopez (2017) undertook the ambitious endeavor of creating a three node Raspberry Pi (i.e., with 1GB of RAM) Cassandra cluster, deployed via Docker[14], including Apache Spark[15]. Romero Lopez (2017) concluded the Raspberry Pi did not have enough memory (i.e., RAM) for Cassandra or Spark and recommend 4 GB and 8 GB of memory for Cassandra and Spark, respectively.

[14] https://www.docker.com/

[15] https://spark.apache.org/

In 2018, Ferencz and Domokos (2018) introduced a data acquisition and storage system using Cassandra and the Raspberry Pi as an alternative to existing IoT data acquisition and storage solutions. Although their system architecture represented a practical and flexible approach to IoT acquisition and storage, Ferencz and Domokos (2018) did not run Cassandra on the Raspberry Pi. Likewise, Ooi et al. (2016) utilized the Raspberry Pi and Cassandra to effectively acquire and store sensor network data (i.e., seismic); however, Cassandra was not run on the Raspberry Pi.

Communication Infrastructure

There is considerable interest in novel, inherently IoT compatible, wireless technologies (i.e., low-power wide-area networks) for seismic applications (Jamali-Rad & Campman, 2018). However, in order to minimize complexity, we utilized commercially available IP-based wired and wireless components to connect our digitizers, edge devices, edge gateways, and edge nodes. The ports used by Cassandra and DSE for cluster communication and the port used by our digitizers are all IP-based; Cassandra and DSE use TCP and the REF TEK 130-01 uses UDP. See Fig. 4.7 for an overview of the communication infrastructure used for the case study presented in the following section.

An important point to consider, when using embedded systems for remote environmental monitoring, is that their onboard Wi-Fi capabilities are typically inadequate for real-world deployments. Typically, remote environmental monitoring requires that embedded systems and other electronics be placed within enclosures located on or near the ground, in which case the quality of wireless connectivity may degrade. Ideally, embedded systems would connect to a Wi-Fi antenna external to the enclosure.

Unfortunately, the Raspberry Pi required board-level modification to connect an external Wi-Fi antenna. The Tinker did allow for the connection of an external antenna via a MHF4 connector; however, the onboard Wi-Fi of the Raspberry Pi (i.e., Raspberry Pi 3 B) and the Tinker did not support our desire to utilize 802.11ac standard communication infrastructure.

A USB Wi-Fi dongle (i.e., TP-Link Archer T2UH AC600[16]) was used to circumvent embedded system Wi-Fi limitations. The TP-Link Archer T2UH AC600 allowed for the connection of an external Wi-Fi antenna and utilized the 802.11ac standard; however, the Tinker did not support the use of the TP-Link Archer T2UH AC600. Ultimately, external antenna capable, 802.11ac standard, Wi-Fi connectivity was achieved by connecting the Tinker to a radio (i.e., EnGenius ENS500EXT-AC[17]) via its Ethernet port. The Raspberry Pi used the TP-Link Archer T2UH AC600 to achieve external antenna capable, 802.11ac standard, Wi-Fi connectivity.

[16] https://www.tp-link.com/us/home-networking/adapter/archer-t2uh/?utm_medium=select-local

[17] https://www.engeniustech.com/engenius-products/enturbo-outdoor-5-ghz-11ac-wave-2-wireless-access-point/

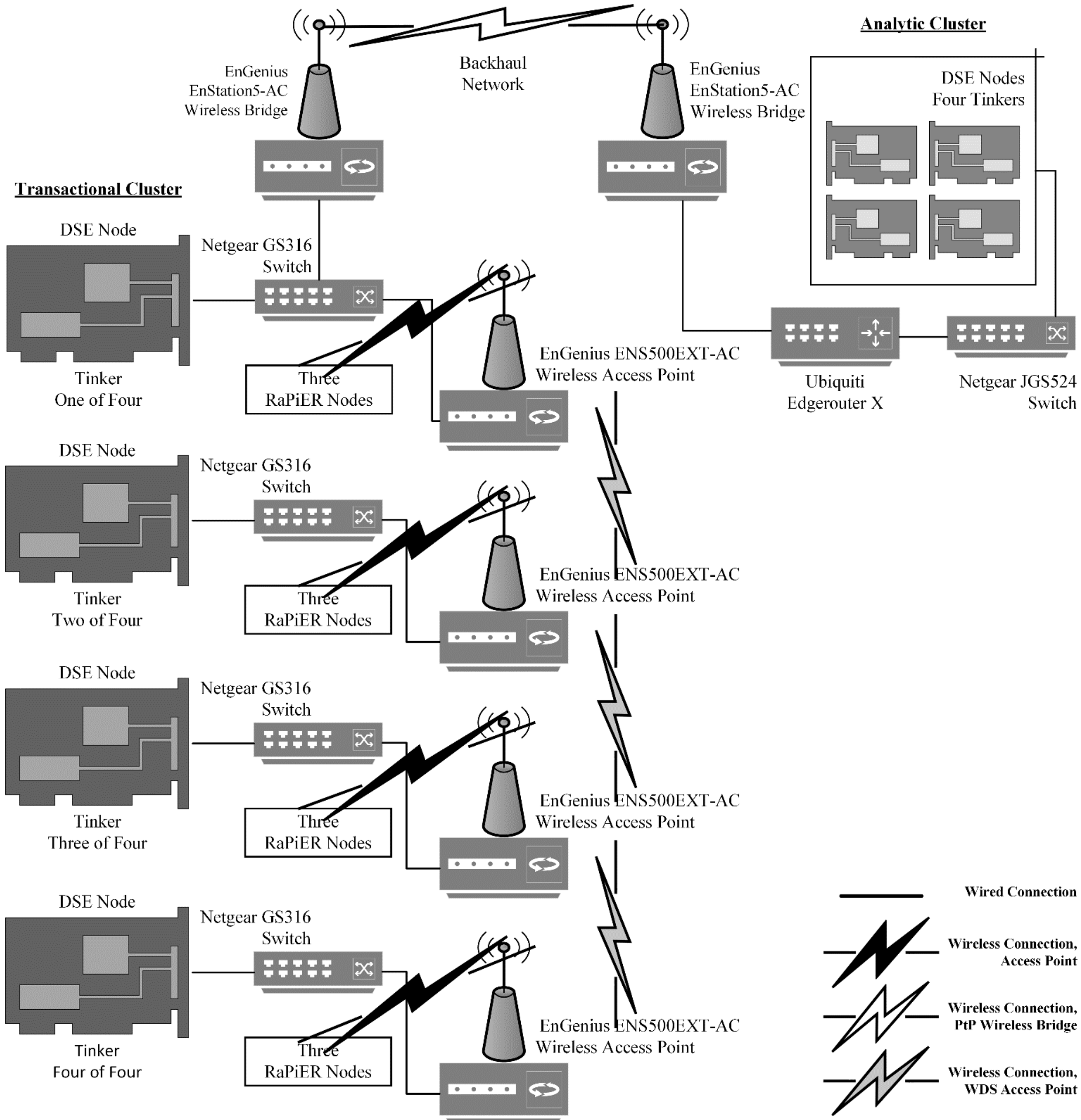


Figure 4.7. Communication infrastructure “wireless” configuration San Emidio Geothermal Field T&E event case study.

Case Study

Background

Our edge storage and computing framework for IoT-based sensor networks was developed over the course of four test and evaluation (T&E) events occurring in May 2017 (Eastland Lakes, Texas), June 2017 (Soda Lake Geothermal Field, Nevada), July 2018 (Baylor Research and Innovation Collaborative, Texas), and May 2019 (San Emidio Geothermal Field, Nevada). Our first deployment to a geothermal field (i.e., Soda Lake Geothermal Field) consisted of 20 seismic stations deployed along a line approximately 575 meters in length and our second deployment to a geothermal field (i.e., San Emidio Geothermal Field) consisted of 144 seismic stations (i.e., planned) deployed along a line approximately 2100 meters in length. The case study described below is specific to the San Emidio Geothermal Field T&E event that occurred in May 2019.

Edge Storage and Computing Workflow

A brief overview of the responsibilities of the layers (i.e., workloads) of the edge storage and computing framework was provided in the Framework Overview section of this paper. What follows is a detailed description of the actions performed by the sensing, transactional, and analytic workloads. The sensing workload is responsible for the acquisition of raw seismic data (i.e., three-channels at 250 sps) and the storage of this data in an archive (i.e., RTPD archive) maintained by the edge gateway. The RTPD archive is maintained "as is" in order to keep an original copy of the raw seismic data. A file watcher running on the edge gateway is used to monitor the RTPD archive. When a RTPD file closes, the file (i.e., five-minute file) is copied to a preprocessing directory.

Every five minutes, files from the preprocessing directory are read, converted, processed, and data for the vertical channel (i.e., a single channel at 50 sps) is saved as a Comma-Separated Value (CSV) file formatted for insertion into the transactional workload DSE cluster (i.e., transactional cluster). The edge gateway connects to the transactional cluster, via a coordinator node, and writes the CSV data into the cluster. The data is then replicated across the transactional cluster according to a user defined replication factor of two. Two copies of the data are saved on the transactional cluster.

DSE Advanced Replication is configured to replicate data from the transactional cluster to the analytic workload DSE cluster (i.e., analytic cluster). If the backhaul network connectivity between the transactional and analytic clusters is down, the transactional cluster maintains the data needing replication until backhaul connectivity is reestablished. With connectivity reestablished, the transactional cluster replicates data to the analytic cluster. The data is then replicated across the analytic cluster according to a user defined replication factor of two. Two copies of the data are saved on the analytic cluster. See Figure 4.8 for an overview of the San Emidio Geothermal Field T&E event edge-based solution workflow.

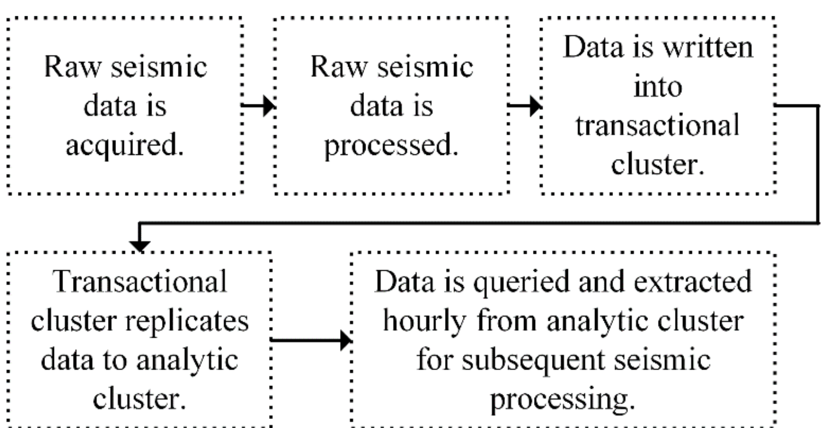


Figure 4.8. San Emidio Geothermal Field T&E event edge-based solution workflow.

Every hour, data from the analytic cluster is queried and extracted for subsequent seismic processing. Query and extract script were executed, against the analytic cluster, on a Mini PC (i.e., Intel NUC[18]) collocated with the analytic cluster. Extracted data was then automatically copied to a second collocated Intel NUC designated for seismic processing.

Implementation

Planned. Prior to the San Emidio Geothermal Field T&E event, we leveraged information obtained from our previous three T&E events to identify performance limiters (e.g., ingest rates, network capacity, etc.) that could not be easily overcome without significant upgrades to solution hardware (i.e., communication infrastructure and embedded systems). Likewise, we considered other physical limiters such as internode spacing of seismic stations, overall length of the seismic line, topography, and operational constraints (e.g., vehicle access, weather, etc.). We considered these limiters in tandem with our geoscientific requirements to organize edge devices, edge gateways, and edge nodes, layer-by-layer (i.e., lower, middle, and upper), into an edge-based solution allowing us to acquire, transmit, store, and process seismic data hourly, in a field setting. The edge-based solution was then replicated and scaled up until it totaled 144 seismic stations. Fig. 4.9 illustrates a "headquarters" consisting of four upper layer edge nodes (i.e., analytic cluster) connected wirelessly to a "squadron" consisting of four middle layer edge nodes (i.e., transactional cluster), in turn, connected (i.e., wired or wirelessly) to twelve lower layer edge gateways, in turn, connected (i.e., wired) to 36

[18] https://www.intel.com/content/www/us/en/products/boards-kits/nuc.html

edge devices. Each squadron was responsible for acquiring approximately 778 million samples per day; however, only approximately 156 million samples per day were inserted into the transactional cluster (i.e., squadron) and subsequently replicated to the analytic cluster (i.e., headquarters).

Actual. We intended to deploy four squadrons and four headquarters totaling 144 lower layer edge devices (i.e., seismic stations), 48 lower layer edge gateways, 16 middle layer edge nodes, and 16 upper layer edge nodes. Ultimately, we only deployed 142 seismic stations due to broken or missing REF TEK 130-01 components and middle layer and upper layer components for two (i.e., squadron #1 and squadron #2) of the four planned squadrons. Fig. 4.10 shows a lower layer station consisting of an edge device and edge gateway and Fig. 4.11 shows a middle layer edge node. Weather-related delays and unplanned troubleshooting (i.e., digitizers and communication infrastructure) were primarily responsible for our inability to deploy all four squadrons. However, the organization of squadrons and headquarters into groups, operating independent of each other, provided an opportunity to assess the performance and suitability of our edge-based solution regardless of the total number of squadrons deployed.

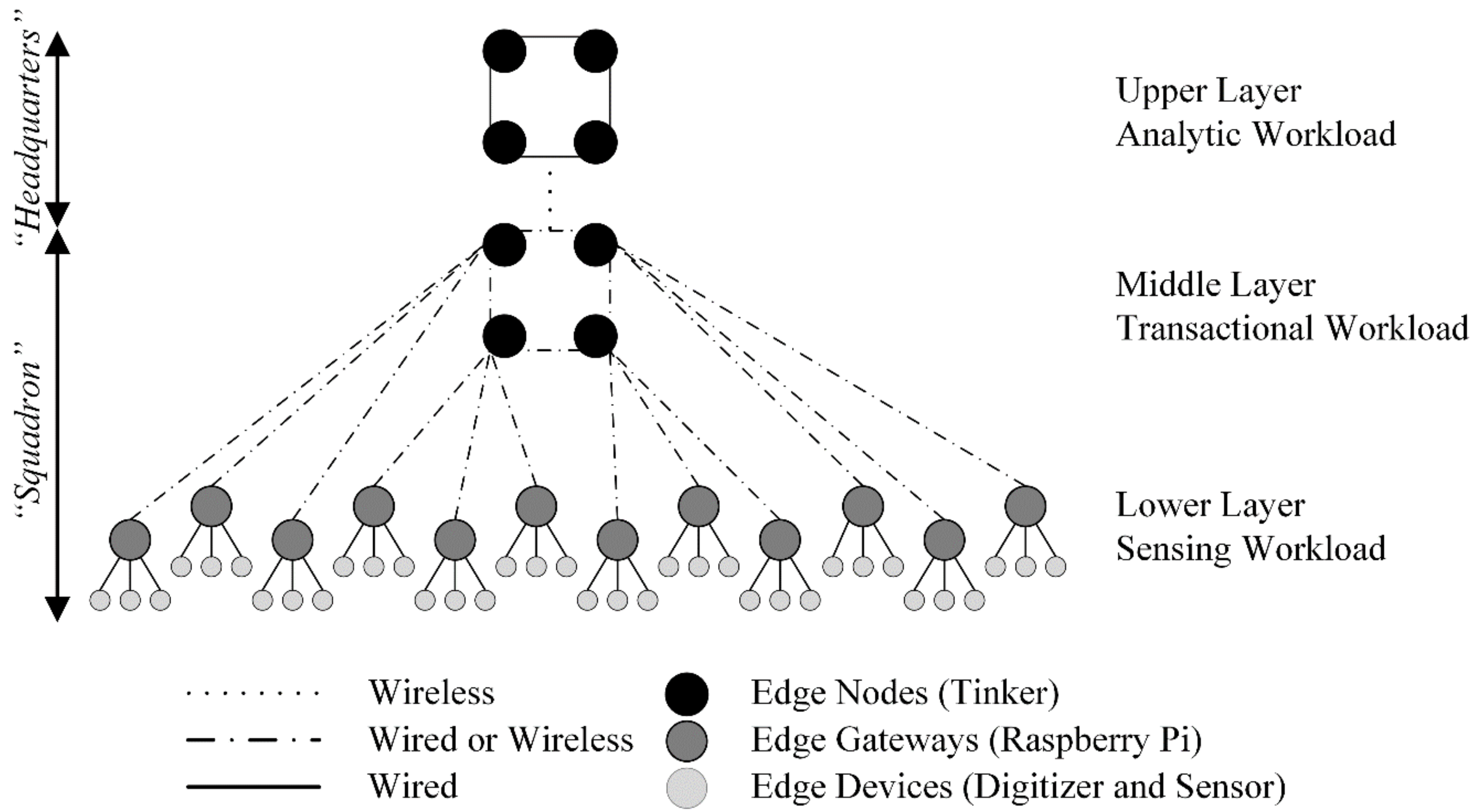


Figure 4.9. San Emidio Geothermal Field T&E event single squadron and headquarters layout pair.

Figure 4.10. San Emidio Geothermal Field T&E event lower layer edge device and edge gateway.

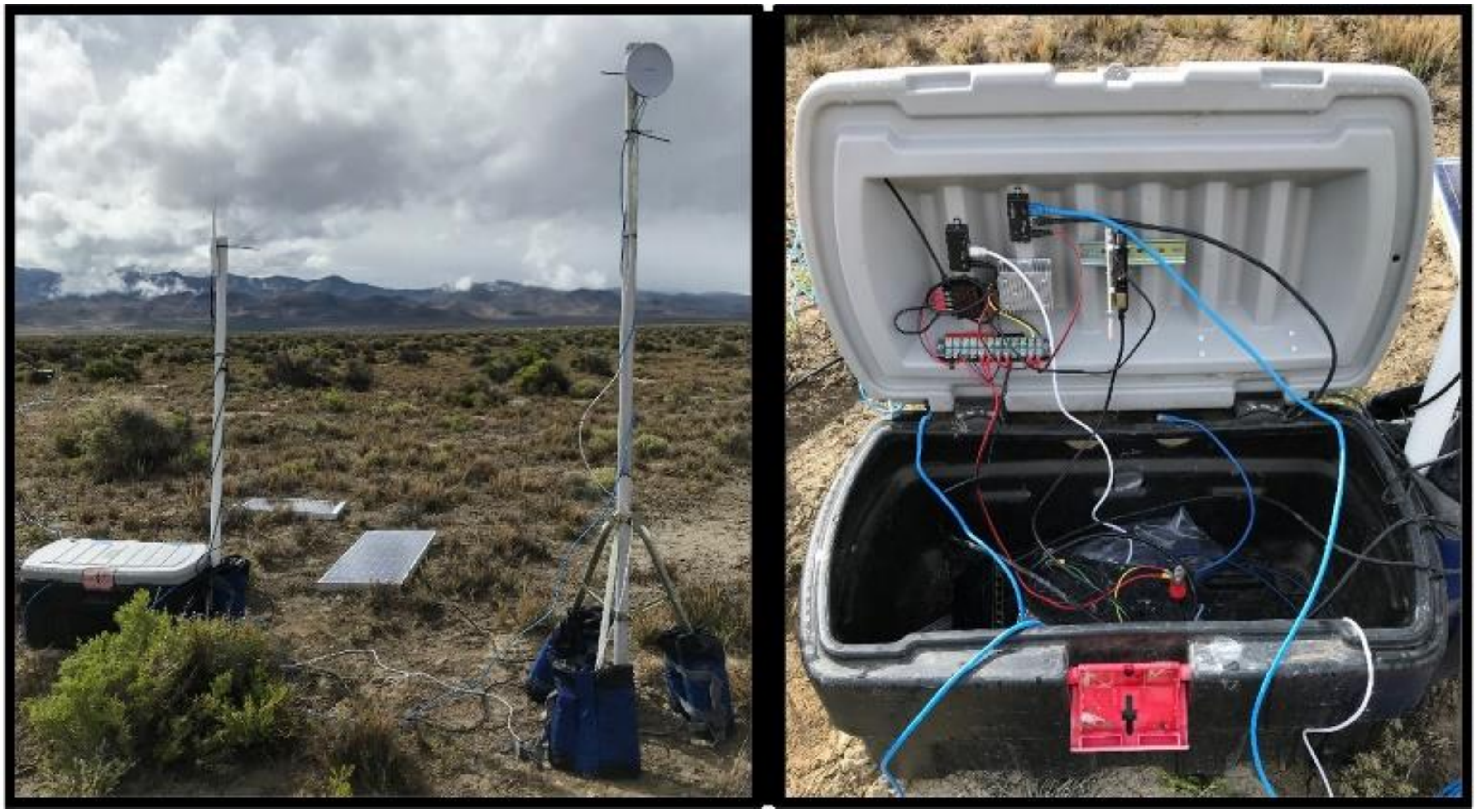


Figure 4.11. San Emidio Geothermal Field T&E event exterior view of middle layer edge node (left) and interior view edge node (right).

Communication infrastructure. During the San Emidio Geothermal Field T&E event, we deployed our edge-solution in three different communication infrastructure configurations (i.e., “wired,” “hybrid,” and “wireless”) corresponding to T&E event test blocks. The three different configurations allowed us to assess system (e.g., distributed database, embedded system, etc.) performance “layer-by-layer” as we transitioned from predominantly wired to predominantly wireless infrastructure. All three configurations utilized wireless (i.e., a point-to-point wireless bridge) for backhaul network cluster-to-cluster (i.e., middle-to-upper layer) replication; likewise, all three configurations utilized wired (i.e., Ethernet cables) for lower layer edge device to edge gateway connectivity. Headquarters edge nodes (i.e., analytic cluster) were collocated and always connected to each other using wired (i.e., Ethernet cable) connections.

The “wired” configuration utilized Ethernet cables to connect lower layer edge gateways to middle layer edge nodes and middle layer edge nodes (i.e., the transactional cluster) to each other. The “hybrid” configuration continued to use Ethernet cables to connect lower layer edge gateways to middle layer edge nodes; however, middle layer edge nodes (i.e., the transactional cluster) were connected to each other wirelessly (i.e., a WDS access point). Lastly, the “wireless” configuration connected middle layer edge nodes (i.e., the transactional cluster) to each other wirelessly (i.e., a WDS access point) and lower layer edge gateways were also connected to middle layer edge nodes wirelessly (i.e., a wireless access point). Table 4.1 provides an overview of communication infrastructure configuration. “Wired” and “hybrid” test blocks were conducted for squadron #1 and “wired,” “hybrid,” and “wireless” test blocks were conducted for squadron #2.

Table 4.1. Communication Infrastructure Configuration.

Communication Infrastructure Configuration	Lower Layer to Lower Layer Edge Device to Edge Gateway	Lower Layer to Middle Layer Edge Gateway to Edge Node	Middle Layer to Middle Layer Edge Node to Edge Node	Middle Layer to Upper Layer Transactional Cluster to Analytic Cluster
"Wired"	Ethernet Cable	Ethernet Cable	Ethernet Cable	Point-to-Point Wireless Bridge
"Hybrid"	Ethernet Cable	Ethernet Cable	Wireless Distribution System Access Point	Point-to-Point Wireless Bridge
"Wireless"	Ethernet Cable	Wireless Access Point	Wireless Distribution System Access Point	Point-to-Point Wireless Bridge

Layer performance – lower layer. Although weather-related delays were primarily responsible, unplanned troubleshooting also impacted our ability to deploy middle layer and upper layer components for the planned four squadrons. Digitizer (i.e., the REF TEK 130-01) problems (i.e., GPS week number rollover (Vincent, 2019) and bad backup battery problems) were relatively easy to solve; however, they were difficult to diagnose. Without the ability to remotely configure the REF TEK 130-01 we were forced to visit seismic stations multiple times before the lower layer components were fully functional. The REF TEK 130-01 can be configured remotely using software provided by the vendor; however, we had not configured the REF TEK 130-01 and the Raspberry Pi (i.e., the edge gateway) to allow remote access to REF TEK 130-01.

Once faulty components were replaced (i.e., the GPS antenna and backup batteries) and the REF TEK 130-01s reconfigured, the lower layer performed its sensing workload as expected. No edge gateway hardware (i.e., the Raspberry Pi) or software (e.g., operating system, Python script, etc.) failures were observed; however, there were instances in which faulty lower layer components required troubleshooting or needed to be replaced (e.g., Ethernet cables, Ethernet switches, etc.).

Layer performance – lower layer to middle layer. For the “wired” configuration approximately 99.7% and 98.9% of the data acquired by the edge devices was inserted, by the edge gateways, into the edge nodes (i.e., the transactional cluster), for squadron #1 and #2 respectively. Approximately 99.2% and 99.9% of “hybrid” configuration data acquired by the edge devices was inserted, by the edge gateways, into the edge nodes (i.e., transactional cluster), for squadron #1 and #2 respectively. Lastly, the “wireless” configuration resulted in approximately 85.0% of the data acquired by the edge devices being inserted, by the edge gateways, into the edge nodes (i.e., transactional cluster) for squadron #2.

We transitioned from 802.11n to 802.11ac standard communication infrastructure for the San Emidio Geothermal Field T&E event. Unfortunately, compatibility problems with the edge gateway’s external antenna and USB Wi-Fi dongle prevented us from deploying our wireless configuration as planned. Spare radios (i.e., the EnGenius ENS500EXT-AC), from our two undeployed squadrons, were used to connect edge gateways wirelessly, via their Ethernet port; however, modifications to the edge gateway “wireless” communication infrastructure were performed in a field setting and required troubleshooting that we believe negatively impacted overall “wireless” configuration performance.

During the San Emidio Geothermal Field T&E event, each edge gateway was responsible for inserting 45,000 samples of seismic data into DSE every five minutes. A minimum rate of 150 sps (i.e., per edge gateway) was required to ingest data into DSE faster than it was created by edge devices. However, data processing overhead, distributed database-related mechanisms, and variations in network capacity could affect ingest rates. The ratio of edge devices, edge gateways, and edge nodes (i.e., 36:12:4) was adjusted, prior to the T&E event, to allow for a least five times (i.e., 750 sps) the required minimum ingest rate.

Ultimately, our use of 802.11ac standard communication infrastructure supported edge gateway ingest rates ranging from approximately 1200 to 1900 sps, depending upon the communication infrastructure configuration. Approximately eight to twelve times the required minimum ingest rate was available during the San Emidio Geothermal Field T&E event. This provided adequate network capacity to support increasing the number of devices per edge gateway, the ingestion of additional device channels (i.e., the horizontal channels), or increasing the sampling rate of data ingested into DSE.

Layer performance – middle layer to upper layer. For squadron #1 and squadron #2, 100% of "wired" test block data inserted into the transactional cluster was replicated to the analytic cluster (i.e., headquarters #1 and headquarters #2), via DSE Advanced Replication. Approximately 99.9% of squadron #1 and 100% of squadron #2 "hybrid" test block data were replicated to their respective analytic cluster (i.e., headquarters #1 and headquarters #2), via DSE Advanced Replication. Lastly, 99.5% of squadron #2 "wireless" test block data was replicated to its analytic cluster (i.e., headquarters #2), via DSE Advanced Replication. The DSE Advanced Replication backlog was monitored during the "wired," "hybrid," and "wireless" test blocks; the DSE Advanced Replication backlog never exceeded more than a few 1000 writes (i.e., samples).

Note that only 85.0% of squadron #2 “wireless” test block data acquired by edge devices was inserted into its transactional cluster; however, 99.5% of squadron #2 “wireless” test block data was replicated from its transactional cluster to analytic cluster. This represents an anomaly where more edge device data resided within an analytic cluster than its corresponding transactional cluster. Although the exact cause of this anomaly remains unknown, we believe the anomaly is a result of the independent communication links (i.e., point-to-point wireless versus WDS access point) and the different mechanisms (e.g., commit log, hinted handoff, etc.) used by DSE versus DSE Advanced Replication (i.e., change-data-capture).

Layer performance – upper layer. An automated script (i.e., Python) was used to query and extract seismic data hourly from the analytic clusters. Leveraging the Python Cassandra driver[19], seismic data was queried from the two analytic clusters (i.e., headquarters #1 and headquarters #2) in parallel and CSV files were extracted for subsequent seismic processing. The query and extract scripts were executed on an Intel NUC and the extracted CSV files were then automatically copied to a second Intel NUC designated for seismic processing. The query and extract of one hour’s worth of squadron data (i.e., 36 seismic stations or approximately 6.5 million samples) took approximately ten minutes, at a rate of approximately 11,000 sps. This provided up to 50 minutes to perform seismic processing before the next one hour’s worth of data was available for query and extract.

[19] https://docs.datastax.com/en/developer/python-driver/3.18/

Results

The overall effectiveness of the edge storage and computing framework for IoT-based sensor networks can be assessed by considering its performance and suitability. The solution's performance refers to quantifiable metrics associated with its ability to function as intended and its suitability refers to its ability to operate in its intended environment.

Performance

Hardware and software. In the previous section, a layer-by-layer assessment of the edge storage and computing framework's performance was provided for the San Emidio Geothermal Field T&E event. This assessment indicates: 1) lower layer and lower-to-middle layer implementations somewhat effectively (i.e., 85% or greater) acquired, processed, transmitted, and stored seismic data into DSE, 2) the middle to upper layer effectively (i.e., 99% or greater) replicated seismic data wirelessly from one DSE cluster to another via DSE Advanced Replication, and 3) the upper layer supported the timely query and extract of seismic data (i.e., approximately 6.5 million samples in 10 minutes) from DSE for subsequent seismic processing.

Architecture and topology. Assessing the performance of the edge-based solution's architecture and topology quantitatively is challenging. Our selection of a Tenet principle inspired architecture and the tree/star hub-and-spoke topology was not driven by specific performance requirements; rather, our choice of architecture and topology evolved over the course of our four T&E events. Nonetheless, we believe the architecture and topology of the edge-based solution support the implementation non-application specific solutions, allowing for tunable scalability, that complement the constraint driven nature of remote environmental monitoring.

Suitability

From a suitability perspective, we are confident the San Emidio Geothermal Field T&E event represented a real-world remote environmental monitoring use case. It was necessary for us to deploy temporary infrastructure supporting the operation of edge devices, edge gateways, and edge nodes (i.e., edge components). Antenna masts, batteries, enclosures, and solar panels were deployed to support the continuous acquisition, transmission, storage, and processing of data, without the need to service edge components.

Power. Our power related support infrastructure relied on one 60 Amp-hour battery and a 20 W solar panel for each edge device (i.e., REF TEK 130-01) not collocated (i.e., not sharing a battery) with an edge gateway (i.e., Raspberry Pi), one 60 Amp-hour battery and 20 W solar panel for each edge device and edge gateway pair (i.e., sharing a battery), and two 60 Amp-hour batteries and a 60 W solar panel for each edge node (i.e., Tinker).

We estimated the overall power draw, via bench testing, of an edge device not collocated with an edge gateway at approximately 2 W, an edge device and edge gateway pair at approximately 4 W, and an edge node at approximately 8W. The variability in power draw is a result of the various components, configurations, and workloads of the edge components. Without considering solar charging, we estimated a minimum of six days’ worth of available power for edge devices not collocated with an edge gateway, three days’ worth of power for edge device and edge gateway pairs, and three days’ worth of power for edge nodes.

Our power estimates proved to be accurate. During the San Emidio Geothermal Field T&E event we experienced more than three days of continuous cloud coverage that limited solar charging. We observed low voltage conditions that triggered solar charge controller power cycling (i.e., load off) until battery voltage was restored. This resulted in the temporary loss of a few edge components, typically in the early morning, until power was restored later that morning.

Environmental. Over the course of our four T&E events, we have had ample opportunity to assess the environmental suitability of the edge-based solution. We deployed the equipment in temperatures that ranged from approximately 1°C to 48°C and in weather that included dry, dusty, rainy, sleeting, and windy conditions.

Provided they are protected from moisture, commercially available components can usually operate across a wide range of temperatures and environmental conditions. The EnGenius ENS500EXT-AC and the Raspberry Pi’s operating temperatures are -20°C to 60°C and -25°C to 80°C, respectively. However, we did experience temperature-related failures (i.e., overheating) when deploying other commercially available components (i.e., home or lab use) such as DC-to-DC converters, Ethernet switches, etc. during our first two T&E events. Ultimately, we transitioned to industrial-use components with operating temperatures more closely aligned with the Raspberry Pi’s operating temperature. We did not experience any problems related to environmental conditions during our last two T&E events.

Discussion

The edge storage and computing framework utilizes easy-to-use, inexpensive, and well-established commercially available components and a popular distributed database to orchestrate an edge-based solution for IoT-based sensor networks. Moreover, our use of an architecture inspired by the Tenet principle and the tree/start hub-and-spoke topology supports highly configurable, general purpose solutions that meet the demands of constraint-driven applications such as remote environmental monitoring. Metrics acquired during the San Emidio Geothermal Field T&E event indicate that the solution supported the in situ acquisition, transmission, storage, and processing of seismic data. As a result, we believe the use of embedded systems (i.e., the Raspberry Pi and Tinker), Mini PCs (i.e., the Intel NUC), DSE, and DSE Advanced Replication to implement an edge-based solution that reduces the cost and risk associated with seismic methods is tenable.

As geoscientists, our need to design, develop, and deploy an edge-based solution to acquire and process seismic data in a field setting was strongly influenced by our method of seismic monitoring. We planned to utilize a cost effective and non-invasive exploration method using ambient (i.e., passive) seismic noise to characterize the subsurface. One of the primary challenges of using passive (i.e., anthropogenic or natural) seismic noise sources is not knowing the characteristics of the noise sources in advance. As a result, it is impossible to know when you have acquired enough data to successfully characterize the subsurface without first processing and analyzing the data.

The edge-based solution described here minimizes the cost and mitigates the risk typically associated with passive methods of seismic exploration. With data in hand, in a field setting, myriad possibilities are available to leverage conventional and bleeding edge methods to generate higher quality data products. For a summary of framework features see Table 4.2.

Table 4.2. Summary of Edge Storage and Computing Framework Features.

	Framework Features
Architecture	Utilizes a tiered architecture supporting workloads of varying complexity.
Topology	Utilizes a hub-and-spoke topology supporting the addition and/or redistribution of edge nodes to overcome common edge-based performance limiters.
Distributed Database	Uses a datastore based upon an open-source solution that: 1) is ideally suited for time series data, 2) supports high availability, fault-tolerant, and geographically distributed implementations, and 3) offers nearly linear horizontal scalability.
Embedded Systems	Uses easy-to-use, inexpensive, and well-established commercially available components.
Communication Infrastructure	Uses commercially available IP-based wired and wireless components.

Although we successfully demonstrated an effective edge storage and computing framework for IoT-based sensor networks, this edge-based solution is not without limitations. For instance, the deployment of conventional sensor networks (i.e., wired or wireless) is often logistically challenging. The effort required to prepare, mobilize, and deploy 142 seismic stations for the San Emidio Geothermal Field T&E event was significant. The cables, digitizers, enclosures, power systems, and sensors required for seismic monitoring are expensive, sizeable, and often require specialized knowledge to configure, deploy, and maintain. Commercially available embedded systems and communication infrastructure are relatively easy-to-use and inexpensive; however, they add to the overall logistical burden of sensor network deployments. The value of edge storage and computing must be weighed carefully against its logistical impact.

Commercially available digitizers used in geoscience applications, such as the REF TEK 130-01, are typically very reliable and do not require a lot of supervision. Although Cassandra and DSE support high availability and fault tolerant implementations, our use of embedded systems to host DSE at the edge represents a novel implementation that required continuous oversight during the San Emidio Geothermal Field T&E event. DSE OpsCenter[20] is an enterprise-grade management and monitoring solution for DSE clusters; however, the OpsCenter client is not available for embedded systems (i.e., ARM architecture processors). In order to monitor our edge-based solution, we used Ansible[21] and custom Python code to log performance metrics; however, our performance monitoring did not include an overview dashboard. Instead, we were forced to manually review log files throughout our T&E event. We recommend using an overview dashboard, such as OpsCenter, to monitor the overall status of the edge-based solution.

The San Emidio Geothermal Field T&E event provided an opportunity to assess, at hourly intervals, our edge-based solution and the quality of our deployment strategy (i.e., process) in a geoscience application. Unfortunately, an abundance of data can often result in "analysis paralysis" that stifles the decision-making process. When confronted with large data rates, we learned that we needed a strategy that automated the assessment edge-based products and process.

---

[20] https://www.datastax.com/products/datastax-enterprise/dse-opscenter

[21] https://www.ansible.com/

Typically, geoscience-related products are generated using “human-in-the-loop” systems that exploit the domain expertise of geoscientists. We believe the automatic generation of geoscience-related products, using an edge-based solution, require specialized methods to objectively assess the overall quality of the products. These specialized methods (e.g., artificial intelligence, statistical, etc.) are necessary to support the relatively rapid operational tempo afforded by an edge-based solution. Likewise, we believe an automated edge-based (i.e., decentralized) version of a seismic quality control program, similar to the program put forth by Ringler et al. (2015), allowing for the timely identification and communication of data quality problems would benefit the edge-based solution.

Conclusion

In this paper, we presented an edge storage and computing framework that leverages commercially available communication infrastructure, digitizers, and embedded systems. The framework is organized in a tiered architecture, arranged in a hub-and-spoke topology, and hosts a popular distributed database to support the acquisition, transmission, storage, and processing of IoT-based sensor network data. We provided details regarding the selection of the architecture, distributed database, embedded systems, and topology used to implement the solution. Lastly, a real-world (i.e., geoscience) case study was presented that leveraged the edge storage and computing framework to acquire, transmit, store, and process millions of samples of seismic data per hour. More than 99% of the data acquired by edge devices (i.e., seismic stations) was stored, queried, and extracted from edge nodes for subsequent seismic processing, in a field setting.

The June 2019 release of the Raspberry Pi 4, in three available configurations (i.e., with 1 GB, 2 GB, and 4 GB of RAM), further complements the architecture, inspired by the Tenet principle, and tree/star hub-and-spoke topology of the solution. The availability of three inherently compatible Raspberry Pi 4 versions, with differing capabilities (i.e., RAM), eliminates the need to use different types of embedded systems and Mini PCs for different layers (i.e., lower, middle, and upper), thus simplifying the overall effort required to design, develop, and deploy an edge-based solution.

More importantly, the 4 GB version of the Raspberry Pi 4 provides an easy-to-use, inexpensive, and well-established embedded system to support an edge-based implementation of Apache Spark. Spark, accepted as an ASF top level project in February 2014, is the most actively developed, open source, unified computing engine for the parallel processing of data on a computer cluster (Chambers & Zaharia, 2018). Spark manages and coordinates the execution of tasks across a cluster of computers (Chambers & Zaharia, 2018). Leveraging the pooled resources of a computer cluster, often in conjunction with a distributed datastore, Spark can process data that a single computer typically cannot (Chambers & Zaharia, 2018).

Our edge-based solution is already capable of implementing Spark. DSE provides additional out-of-the-box capabilities, via DSE Analytics[22], that include Spark integration. Using the Raspberry Pi's quad-core processor and 4 GB of RAM, upper layer edge nodes could be configured for an analytic workload that leverages the DSE cluster to support Spark (i.e., distributed processing). We have used the Raspberry Pi 4 (i.e., with 4 GB of RAM) to host a four-node, i.e., Spark-enabled, analytic workload DSE cluster

[22] https://docs.datastax.com/en/dse/6.0/dse-dev/datastax_enterprise/analytics/analyticsTOC.html

and have performed a series of bench tests to assess the feasibility of edge-based distributed processing. We believe the utilization of multiple Raspberry Pi 4s to host a Spark-enabled DSE cluster is feasible and warrants further investigation.

Funding

This research was funded by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, under FOA DE-FOA-0001445.

Acknowledgments

The development and testing described here was supported by funding from the U.S. Department of Energy (Energy Efficiency and Renewable Energy Agency, Geothermal Technologies Program, Award Number: DE-EE0007699). Instrumentation and data archiving were provided by the Incorporated Research Institutions for Seismology through their PASSCAL and Data Management Centers, respectively. Both centers provided technical support.

All data and associated metadata from our T&E events at Eastland Lakes, Soda Lake Geothermal Field, the Baylor Research and Industrial Collaborative, and San Emidio Geothermal Field have been archived at the Data Management Center (DMC) operated by the Incorporated Research Institutions for Seismology (IRIS). The open-source algorithms and codes used to derive processed and interpreted seismic results, including detailed descriptions of data processing methodology, are available with MSNoise distribution: www.msnoise.org.

We want to thank Joe Iovenitti, Tim Meredith, John Queen, Marge Queen, Diego Ugalde Quiros, and Joseph Thangraj for their programmatic and technical support of this project. We also want to thank the San Emidio Geothermal Field team for their support of the May 2019 T&E event. Lastly, F. Sepulveda wants to thank Jon Haddad (The Last Pickle) and Patrick McFadin (DataStax) for their support regarding Cassandra and DSE.

Disclaimer

Conflict of Interest

The final report for this research was submitted to the U.S. Department of Energy in December 2019. In May 2020, F. Sepulveda began working with DataStax as a Data Architect. DataStax had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

Abramova, V., Bernardino, J., & Furtado, P. (2014). Which NoSQL Database? A Performance Overview, *1*(2), 8.

Alcaraz, C., Najera, P., Lopez, J., & Roman, R. (2010). Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50. https://doi.org/10.1145/1721654.1721672

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684–700. https://doi.org/10.1016/j.future.2015.09.021

Brewer, E. (2012). CAP twelve years later: How the “rules” have changed. *Computer*, *45*(2), 23–29. https://doi.org/10.1109/MC.2012.37

Carpenter, J., & Hewitt, E. (2016). *Cassandra: the definitive guide* (Second edition). Sebastopol, CA: O’Reilly Media, Inc.

Challen, G., and M. Welsh (2010). Volcano monitoring: Addressing data quality through iterative deployment, in *Wireless Sensor Networks*, E. Gaura (Editor), Springer, New York, New York, 71–113.

Chambers, B., & Zaharia, M. (2018). *Spark: the definitive guide: big data processing made simple* (First edition). Sebastapol, CA: O’Reilly Media.

Chen, C. L. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*(Supplement C), 314–347. https://doi.org/10.1016/j.ins.2014.01.015

Chung, M., & Kim, J. (2016). The Internet Information and Technology Research Directions based on the Fourth Industrial Revolution. *KSII Transactions on Internet and Information Systems*, *10*, 1311–1320. https://doi.org/10.3837/tiis.2016.03.020

Clemente, J., Li, F., Valero, M., Chen, A., & Song, W. (2020). ASIS: Autonomous Seismic Imaging System With In Situ Data Analytics and Renewable Energy. *IEEE Systems Journal*, *14*(1), 1277–1284. https://doi.org/10.1109/JSYST.2019.2920073

Confais, B., Lebre, A., & Parrein, B. (2016). Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 294–301). https://doi.org/10.1109/CloudCom.2016.0055

Corke, P., Wark, T., Jurdak, R., Hu, W., Valencia, P., & Moore, D. (2010). Environmental Wireless Sensor Networks. *Proceedings of the IEEE*, *98*(11), 1903–1917. https://doi.org/10.1109/JPROC.2010.2068530

Deshmukh, A. D., & Shinde, U. B. (2016). A low cost environment monitoring system using raspberry Pi and arduino with Zigbee. In *2016 International Conference on Inventive Computation Technologies (ICICT)* (Vol. 3, pp. 1–6). https://doi.org/10.1109/INVENTIVE.2016.7830096

Duarte, A., & Bernardino, J. (2016). Cassandra for Internet of Things: An Experimental Evaluation. In *IoTBD*. https://doi.org/10.5220/0005846400490056

Estrin, D., Govindan, R., Heidemann, J., & Kumar, S. (1999). Next century challenges: scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99* (pp. 263–270). Seattle, Washington, United States: ACM Press. https://doi.org/10.1145/313451.313556

Ferencz, K., & Domokos, J. (2018). IoT Sensor Data Acquisition and Storage System Using Raspberry Pi and Apache Cassandra. In *2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)* (pp. 000143–000146). https://doi.org/10.1109/CANDO-EPE.2018.8601139

Fratu, O., Pejanovic-Djurisic, M., Poulkov, V., & Gavrilovska, L. (2016). Introduction to Special Issue "Current Trends in Information and Communications Technology." *Wireless Personal Communications*, *87*(3), 615–617. https://doi.org/10.1007/s11277-016-3228-6

Gaber, M. M., Aneiba, A., Basurra, S., Batty, O., Elmisery, A. M., Kovalchuk, Y., & Rehman, M. H. U. (2019). Internet of Things and data mining: From applications to techniques and systems. *WIREs Data Mining and Knowledge Discovery*, *9*(3), e1292. https://doi.org/10.1002/widm.1292

Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., et al. (2015). Edge-centric Computing: Vision and Challenges. *SIGCOMM Comput. Commun. Rev.*, *45*(5), 37–42. https://doi.org/10.1145/2831347.2831354

Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, *33*(2), 51–59. https://doi.org/10.1145/564585.564601

Gilbert, S., & Lynch, N. (2012). Perspectives on the CAP Theorem. *Computer*, *45*(2), 30–36. https://doi.org/10.1109/MC.2011.389

Gnawali, O., Jang, K.-Y., Paek, J., Vieira, M., Govindan, R., Greenstein, B., et al. (2006). The Tenet Architecture for Tiered Sensor Networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems* (pp. 153–166). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1182807.1182823

Greengard, S. (2015). *The internet of things*. Cambridge, Massachusetts: MIT Press.

Heath, N. (2019, December). How the Raspberry Pi was created: A visual history of the $35 board. Retrieved August 6, 2019, from https://www.techrepublic.com/pictures/how-the-raspberry-pi-was-created-a-visual-history-of-the-35-board/

Ilyas, M., & Mahgoub, I. (Eds.). (2006). *Sensor network protocols*. Boca Raton: CRC/Talor & Francis.

Jamali-Rad, H., & Campman, X. (2018). Internet of Things-based wireless networking for seismic applications. *Geophysical Prospecting*, *66*(4), 833–853. https://doi.org/10.1111/1365-2478.12617

Jamali-Rad, H., Campman, X., MacKay, I., Walk, W., Beker, M., van den Brand, J., et al. (2018). IoT-based wireless seismic quality control. *The Leading Edge*, *37*(3), 214–221. https://doi.org/10.1190/tle37030214.1

Jan, M. A., Khan, F., & Alam, M. (2019). *Recent Trends and Advances in Wireless and IoT-enabled Networks*. Retrieved from https://doi.org/10.1007/978-3-319-99966-1

Jones, C. M. (2018). The oil and gas industry must break the paradigm of the current exploration model. *Journal of Petroleum Exploration and Production Technology*, *8*(1), 131–142. https://doi.org/10.1007/s13202-017-0395-2

Kamath, G., Agnihotri, P., Valero, M., Sarker, K., & Song, W.-Z. (2016). Pushing Analytics to the Edge. In *2016 IEEE Global Communications Conference (GLOBECOM)* (pp. 1–6). https://doi.org/10.1109/GLOCOM.2016.7842181

Karatas, M., & Onggo, B. S. (2019). Optimising the barrier coverage of a wireless sensor network with hub-and-spoke topology using mathematical and simulation models. *Computers & Operations Research*, *106*, 36–48. https://doi.org/10.1016/j.cor.2019.02.007

Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, *6*(1), 307–335. https://doi.org/10.1016/S0966-8349(98)00042-4

Kocakulak, M., & Butun, I. (2017). An overview of Wireless Sensor Networks towards internet of things. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 1–6). https://doi.org/10.1109/CCWC.2017.7868374

Kumar, R., & Rajasekaran, M. P. (2016). An IoT based patient monitoring system using raspberry Pi. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)* (pp. 1–4). https://doi.org/10.1109/ICCTIDE.2016.7725378

Lakshman, A., & Malik, P. (2010). Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, *44*(2), 35. https://doi.org/10.1145/1773912.1773922

Lazarescu, M. T. (2013). Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *3*(1), 45–54. https://doi.org/10.1109/JETCAS.2013.2243032

Le, T. D., Kim, S. H., Nguyen, M. H., Kim, D., Shin, S. Y., Lee, K. E., & Righi, R. da R. (2014). EPC information services with No-SQL datastore for the Internet of Things. In *2014 IEEE International Conference on RFID (IEEE RFID)* (pp. 47–54). https://doi.org/10.1109/RFID.2014.6810711

Mainetti, L., Patrono, L., & Vilei, A. (2011). Evolution of wireless sensor networks towards the Internet of Things: A survey. In *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks* (pp. 1–6).

Martinez, K., Padhy, P., Elsaify, A., Zou, G., Riddoch, A., Hart, J. K., & Ong, H. L. R. (2006). Deploying a sensor network in an extreme environment. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)* (Vol. 1, p. 8 pp.-). https://doi.org/10.1109/SUTC.2006.1636175

Merchant, H., & Ahire, D. D. (2017). Industrial Automation using IoT with Raspberry Pi.

Morgan, T. P. (2018, February 19). Pushed To The Edge. Retrieved December 28, 2019, from https://www.nextplatform.com/2018/02/19/pushed-to-the-edge/

Neves, P., & Rodrigues, J. (2010). Internet Protocol over Wireless Sensor Networks, from Myth to Reality. *Journal of Communications*, *5*. https://doi.org/10.4304/jcm.5.3.189-196

Nordrum, A. (2016, August). Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated - IEEE Spectrum. Retrieved January 7, 2020, from https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

Ooi, G. L., Tan, P. S., Lin, M.-L., Wang, K.-L., Zhang, Q., & Wang, Y.-H. (2016). Near real-time landslide monitoring with the smart soil particles. *Japanese Geotechnical Society Special Publication*, *2*, 1031–1034. https://doi.org/10.3208/jgssp.HKG-05

Özsu, M. T., & Valduriez, P. (2011). *Principles of distributed database systems* (3rd ed). New York: Springer Science+Business Media.

Ploetz, A., Neeraj, N., & Malepati, T. (2018). *Mastering Apache Cassandra 3.x: an expert guide to improving database scalability and availability without compromising performance*. Retrieved from http://proquest.safaribooksonline.com/9781789131499

Premsankar, G., Di Francesco, M., & Taleb, T. (2018). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

Richardson, D. P. (2017). *Cloud Benchmark Testing of Cassandra on Raspberry Pi for Internet of Things Capability* (No. AFIT-ENG-MS-17-M-065). AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH WRIGHT-PATTERSON AFB United States. Retrieved from https://apps.dtic.mil/docs/citations/AD1054746

Ringler, A. T., Hagerty, M. T., Holland, J., Gonzales, A., Gee, L. S., Edwards, J. D., et al. (2015). The data quality analyzer: A quality control program for seismic data. *Computers & Geosciences*, *76*, 96–111. https://doi.org/10.1016/j.cageo.2014.12.006

Romero López, E. (2017). Creación de un cluster formado por tres Raspberry PI-3 que alojarán una base de datos NoSQL (Apache Cassandra) y motor de cálculo distribuido (Apache Spark), desplegado en contenedores (Docker). Retrieved from http://openaccess.uoc.edu/webapps/o2/handle/10609/67505

Sanchez-Gomez, J., Gallego-Madrid, J., Sanchez-Iborra, R., Santa, J., & Skarmeta, A. F. (2020). Impact of SCHC Compression and Fragmentation in LPWAN: A Case Study with LoRaWAN. *Sensors*, *20*(1), 280. https://doi.org/10.3390/s20010280

Satyanarayanan, M. (2017). The Emergence of Edge Computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/MC.2017.9

Savazzi, S., Goratti, L., Fontanella, D., Nicoli, M., & Spagnolini, U. (2011). Pervasive UWB sensor networks for oil exploration. In *2011 IEEE International Conference on Ultra-Wideband (ICUWB)* (pp. 225–229). https://doi.org/10.1109/ICUWB.2011.6058833

Savazzi, Stefano, Rampa, V., & Spagnolini, U. (2009). High-Density Wireless Geophone Networks for Oil and Gas Monitoring and Exploration. *ERCIM News*.

Savazzi, Stefano, Goratti, L., Spagnolini, U., & Latva-aho, M. (2009). Short-range wireless sensor networks for high density seismic monitoring. Retrieved August 5, 2020, from /paper/Short-range-wireless-sensor-networks-for-high-Savazzi-Goratti/00ca7e20a8f216b9078950d66d6fa4137dd528fe

Sepulveda, F., & Pulliam, J. (2016). The Internet of Geophysical Things: Raspberry Pi Enhanced REF TEK (RaPiER) System Integration and Evaluation. *Seismological Research Letters*, *87*(2A), 345–357. https://doi.org/10.1785/0220150234

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. *Future Generation Computer Systems*, *99*, 278–294. https://doi.org/10.1016/j.future.2019.04.016

Talzi, I., Hasler, A., Gruber, S., & Tschudin, C. (2007). PermaSense: Investigating permafrost with a WSN in the Swiss Alps (pp. 8–12). Presented at the Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007. https://doi.org/10.1145/1278972.1278974

Toutain, L., Minaburo, A., Barthel, D., Zúñiga, J.-C., & Gomez, C. (n.d.). Static Context Header Compression (SCHC) and fragmentation for LPWAN, application to UDP/IPv6. Retrieved August 5, 2020, from https://tools.ietf.org/html/draft-ietf-lpwan-ipv6-static-context-hc-10

Traffic between the clusters | DSE 6.0 Admin guide. (n.d.). Retrieved August 3, 2020, from https://docs.datastax.com/en/dse/6.0/dse-admin/datastax_enterprise/advReplication/advRepClusterTraffic.html

Upton, E. (2019, June). Raspberry Pi 4 on sale now from $35 - Raspberry Pi. Retrieved August 26, 2019, from https://www.raspberrypi.org/blog/raspberry-pi-4-on-sale-now-from-35/

USArray - Adopt a Station - Lower 48. (n.d.). Retrieved January 10, 2020, from http://www.usarray.org/researchers/adopt/lower48

Valero, M., Li, F., & Song, W. (2019). Smart Seismic Network for Shallow Subsurface Imaging and Infrastructure Security, 13.

Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016). Challenges and Opportunities in Edge Computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 20–26). https://doi.org/10.1109/SmartCloud.2016.18

Vasseur, J.-P., & Dunkels, A. (2010). *Interconnecting smart objects with IP: the next Internet*. Burlington, MA: Morgan Kaufmann Publishers/Elsevier.

van der Veen, J. S., van der Waaij, B., & Meijer, R. J. (2012). Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 431–438). Honolulu, HI, USA: IEEE. https://doi.org/10.1109/CLOUD.2012.18

Vincent, J. (2019, March 8). Older GPS devices are facing their own mini Y2K bug next month. Retrieved August 12, 2019, from https://www.theverge.com/2019/3/8/18255847/gps-week-rollover-issue-2019-garmin-tomtom-devices-affected

Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., & Welsh, M. (2005). Monitoring volcanic eruptions with a wireless sensor network. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005* (pp. 108–120). https://doi.org/10.1109/EWSN.2005.1462003

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., et al. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, *98*, 289–330. https://doi.org/10.1016/j.sysarc.2019.02.009

Zorzi, M., Gluhak, A., Lange, S., & Bassi, A. (2010). From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view. *IEEE Wireless Communications*, *17*(6), 44–51. https://doi.org/10.1109/MWC.2010.5675777

CHAPTER FIVE

Conclusion

We demonstrated the feasibility of leveraging an IoT-based sensor array to orchestrate edge-based storage and computing resources capable of characterizing the subsurface, using ambient seismic noise, in near real-time. Our edge-based solution utilized commercially available communication infrastructure, digitizers, embedded systems, and DSE (powered by Apache Cassandra) to store and process sensor array data, in a field setting. The scalability and reliability provided by our novel use of DSE, as an edge-based distributed database, was also validated.

Although the REF TEK 130-01 is readily available and in wide use by the geoscience community, its end-of-life status motivates us to consider other seismic digitizers. Cableless (i.e., nodal) seismic digitizers, like those used in oil and gas exploration, may represent a more cost-effective and rugged alternative to the REF TEK 130-01. Future implementations of the edge-based solution described herein should consider other nodal seismic digitizers. Coordination with their respective vendors is likely necessary to discuss the porting of existing software to the ARM architecture common to embedded systems.

We struggled in early 2018 to identify an inexpensive and widely used embedded system, with 2 GB of RAM, to support our edge-based implementation of DSE. By May 2020, three variants of the immensely popular Raspberry Pi 4 are currently available with 2, 4, and 8 GB of RAM. The incredible pace of edge device performance enhancements (e.g., embedded systems, mobile devices, etc.) provides new opportunities to leverage unused device resources, computing and memory, to implement edge-based distributed processing (i.e., Apache Spark[1]).

Spark is the most actively developed, open source, unified computing engine for the parallel processing of data on a computer cluster (Chambers & Zaharia, 2018). Spark leverages the pooled resources of a computer cluster, often in conjunction with a distributed datastore, to process data that a single computer cannot (Chambers & Zaharia, 2018). Spark can connect to a wide variety of distributed data stores and messages buses, supports traditional bulk extract, load, transform (ETL) operations, and provides libraries for common data analysis tasks such as graph analytics (GraphX), machine learning (MLib), streaming processing (Spark Streaming), and working with structured data (Spark SQL) (Chambers & Zaharia, 2018).

Given the variety, velocity, and volume of data (i.e., "Big Data") commonly generated during seismic exploration and monitoring, the excess computing and memory resources of IoT-based sensor array nodes could be used in conjunction with edge-based distributed data storage and processing to extend seismic exploration and monitoring functionality and performance.

DSE supports Spark via DSE Analytics[2] (i.e., integrated Spark). We have performed limited DSE Analytics bench testing with four Raspberry Pi 4s (i.e., 4 GB) and believe distributed edge-based processing is feasible and warrants further investigation. In additional to the seismic application area (i.e., ambient seismic noise)

[1] https://spark.apache.org/

[2] https://docs.datastax.com/en/dse/6.0/dse-dev/datastax_enterprise/analytics/analyticsTOC.html

described herein, we believe our framework could be implemented, with relative ease, to enhance and optimize active source seismic and engineering seismology methods.

Typically, management and monitoring solutions are used to keep track of distributed databases, virtual machines, etc. within the cloud or on-premises data centers. These solutions often use a graphical user interface (GUI) to provide operators an easy way to observe the status of their resources. Unfortunately, we did not implement a GUI during our field T&E events. We relied on logs to assess the status and performance of our edge-based solution. The manual review of logs proved labor intensive and we recommend the adoption of a GUI approach to better monitor an edge-based solution.

The creation of geoscience-related analysis products is typically a “human-in-the-loop” process. However, an abundance of data, available in near real-time, can overwhelm geoscientists performing analysis in a field setting. We believe the continued development of specialized methods (e.g., artificial intelligence, statistical, etc.) are necessary to objectively assess the overall quality of analysis products created using a near real-time edge-based solution. Moreover, the implementation of an automated edge-based seismic quality control program, like the centralized program put forth by Ringer et al. (2015), is necessary for the timely identification of data quality problems.

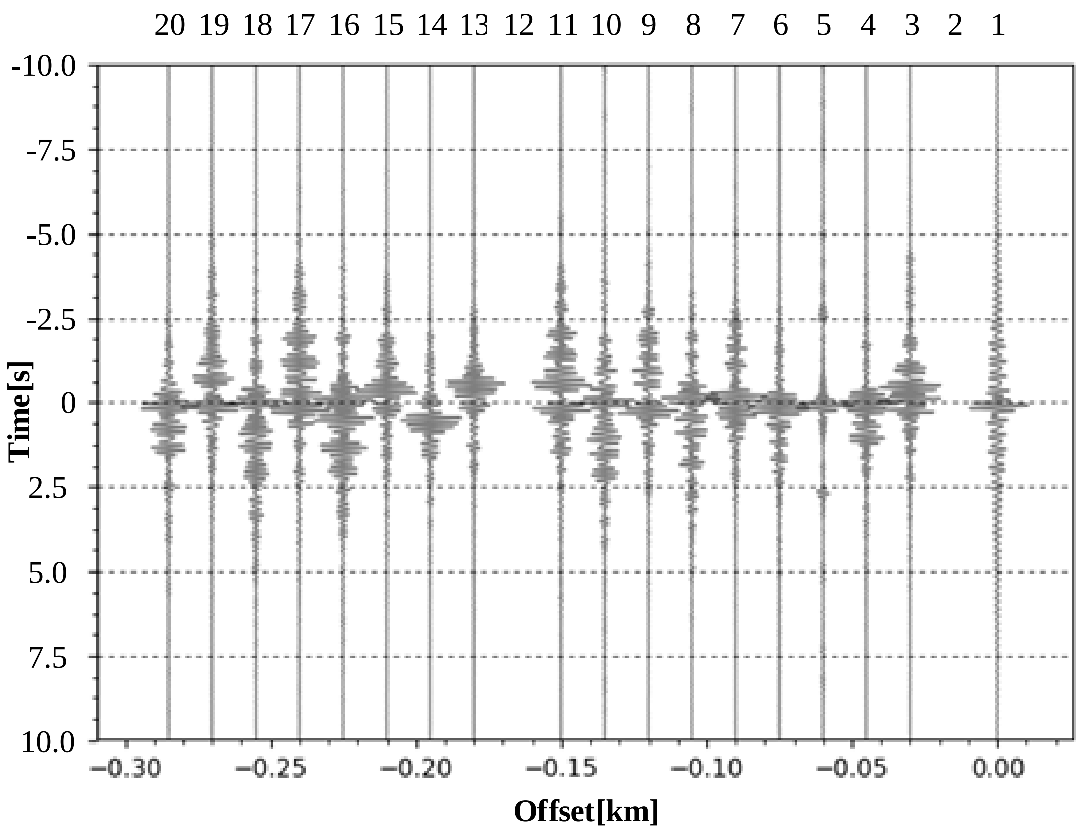
APPENDIX

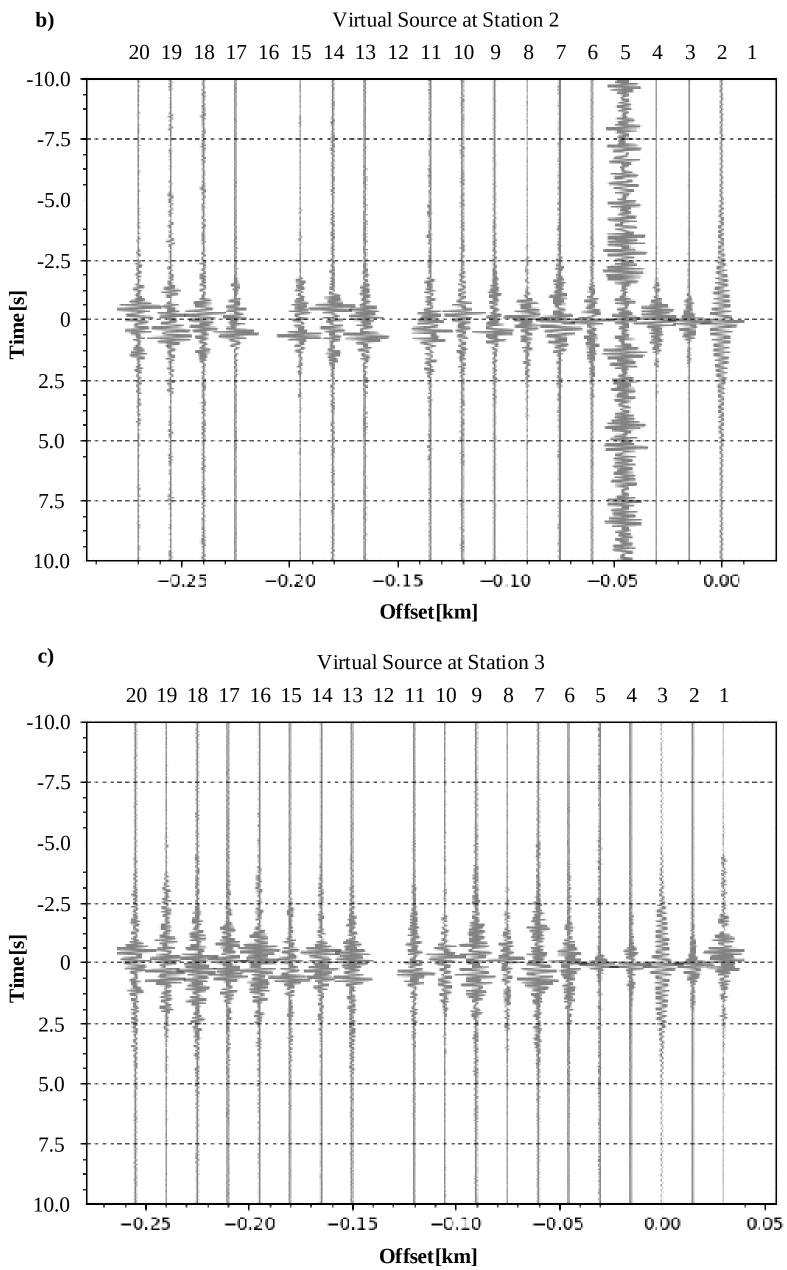
APPENDIX

Supplemental Material to Chapter Three

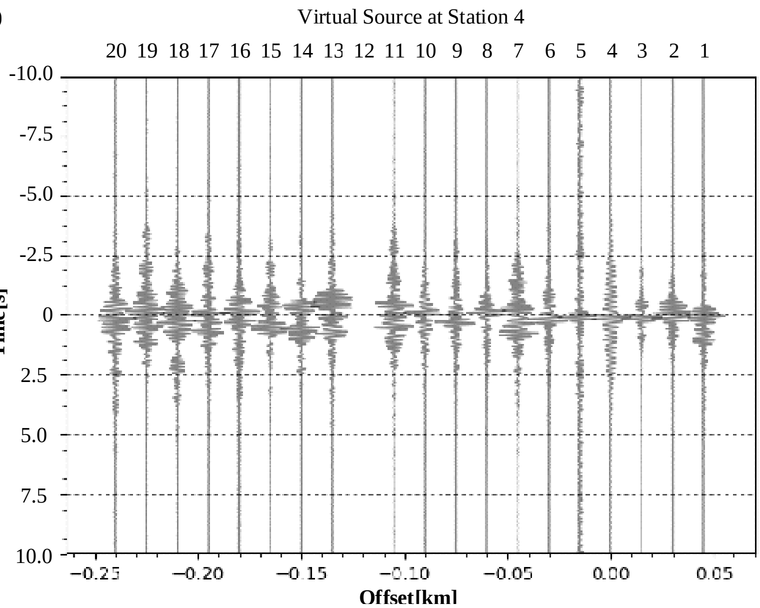
Figure A.1 (a-q). The following 17 final VSGs were produced automatically after 25 hours of data acquisition by the RaPiER array, Eastland Lakes. The source location is indicated at the top of the figure and by the 0.0 location on the x-axis. The distance of each station from the virtual source location is indicated on the x-axis as “offset.” The y-axis refers to time lag.

a) Virtual Source at Station 1

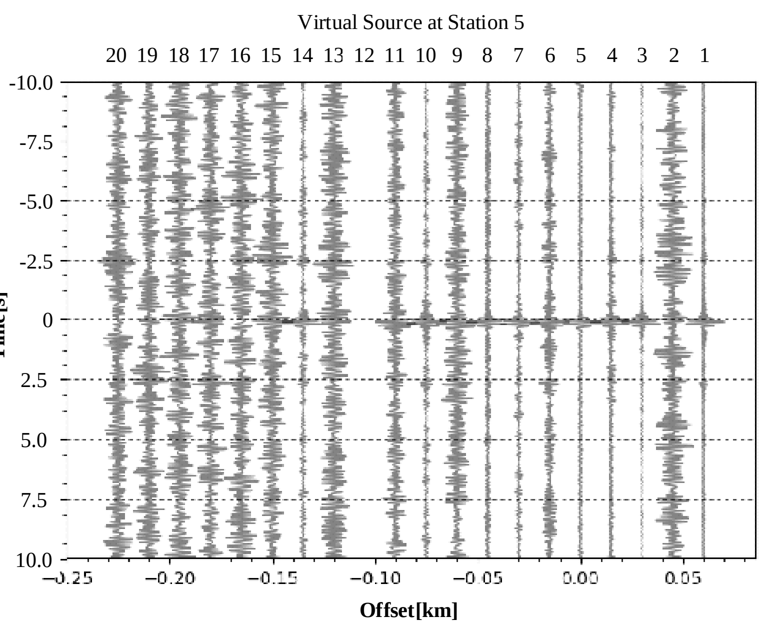



b)
Virtual Source at Station 2
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Time[s]
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
−0.25 −0.20 −0.15 −0.10 −0.05 0.00
Offset[km]
c)
Virtual Source at Station 3
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Time[s]
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
−0.25 −0.20 −0.15 −0.10 −0.05 0.00 0.05
Offset[km]

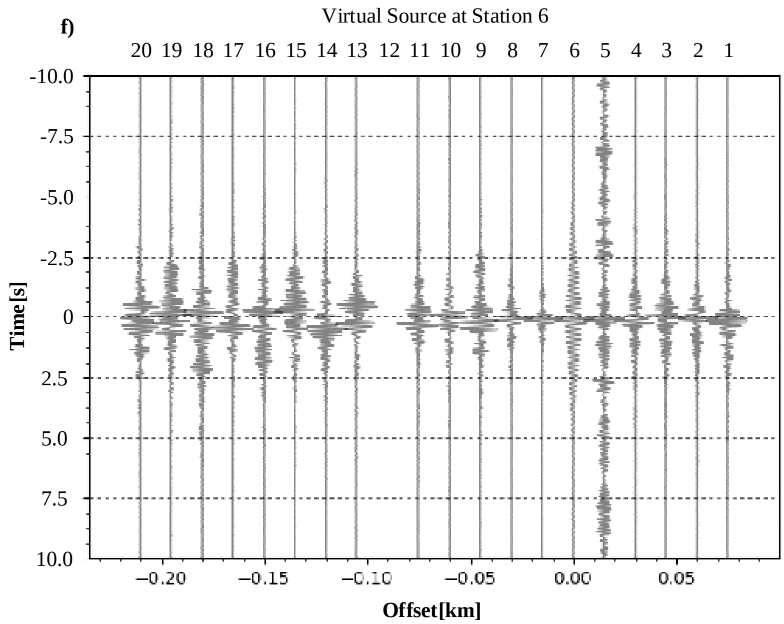
d)


Virtual Source at Station 4
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.25 −0.20 −0.15 −0.10 −0.05 0.00 0.05
Offset[km]

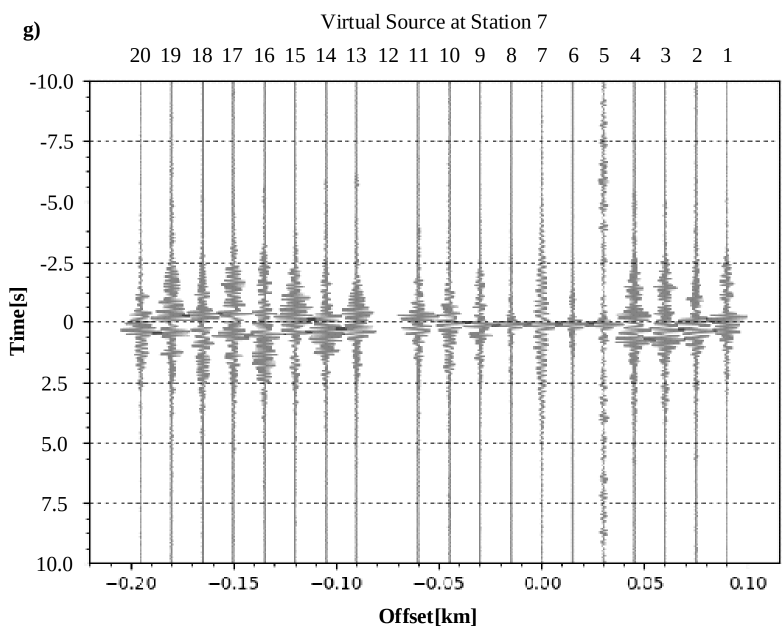
e)


Virtual Source at Station 5
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.25 −0.20 −0.15 −0.10 −0.05 0.00 0.05
Offset[km]

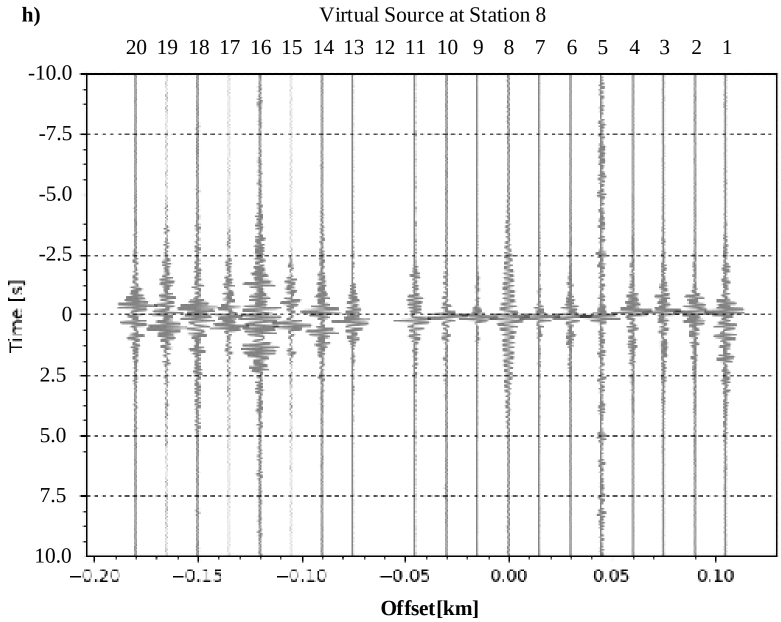
f)


Virtual Source at Station 6
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.20 −0.15 −0.10 −0.05 0.00 0.05
Offset[km]

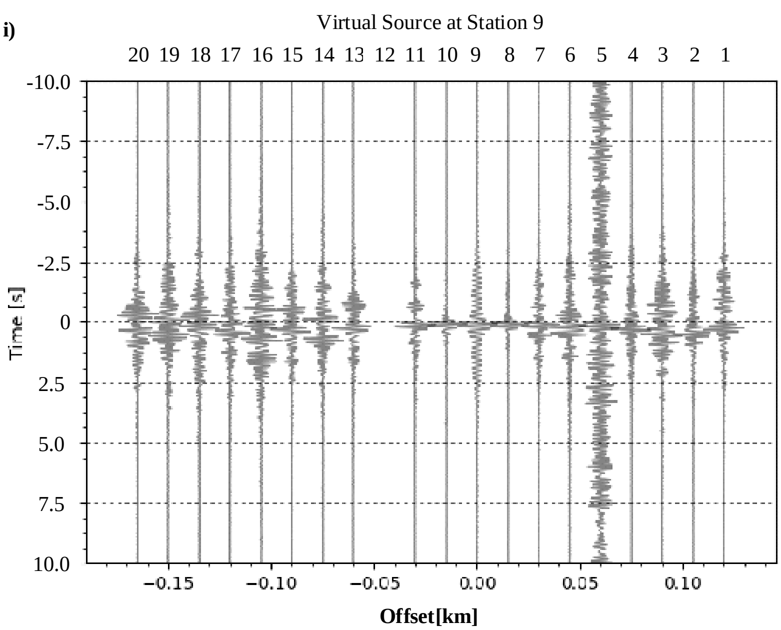
g)


Virtual Source at Station 7
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.20 −0.15 −0.10 −0.05 0.00 0.05 0.10
Offset[km]

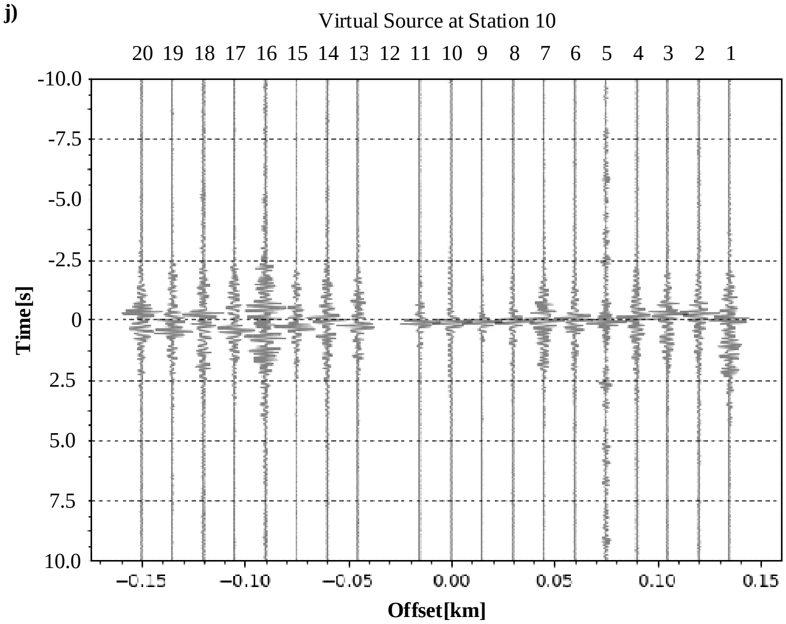
h)


Virtual Source at Station 8
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time [s]
−0.20 −0.15 −0.10 −0.05 0.00 0.05 0.10
Offset[km]

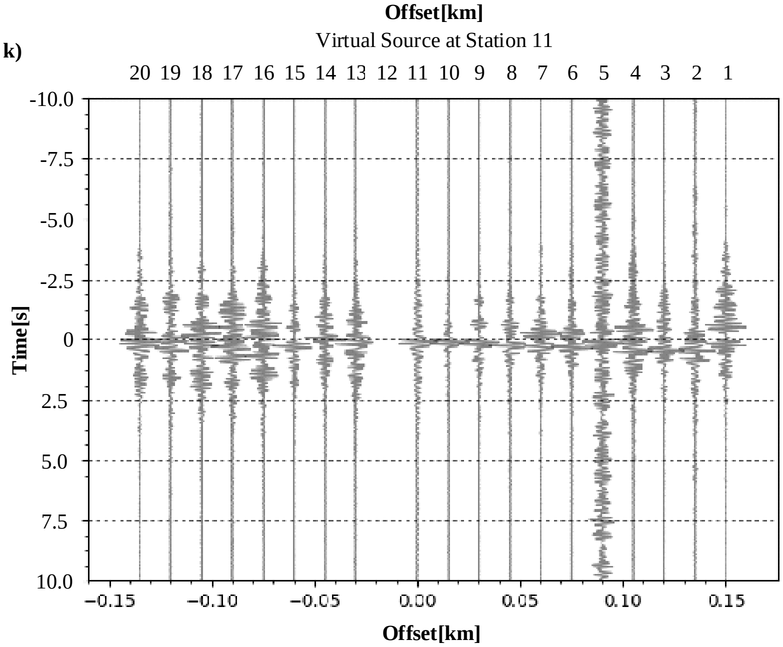
i)


Virtual Source at Station 9
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time [s]
−0.15 −0.10 −0.05 0.00 0.05 0.10
Offset[km]

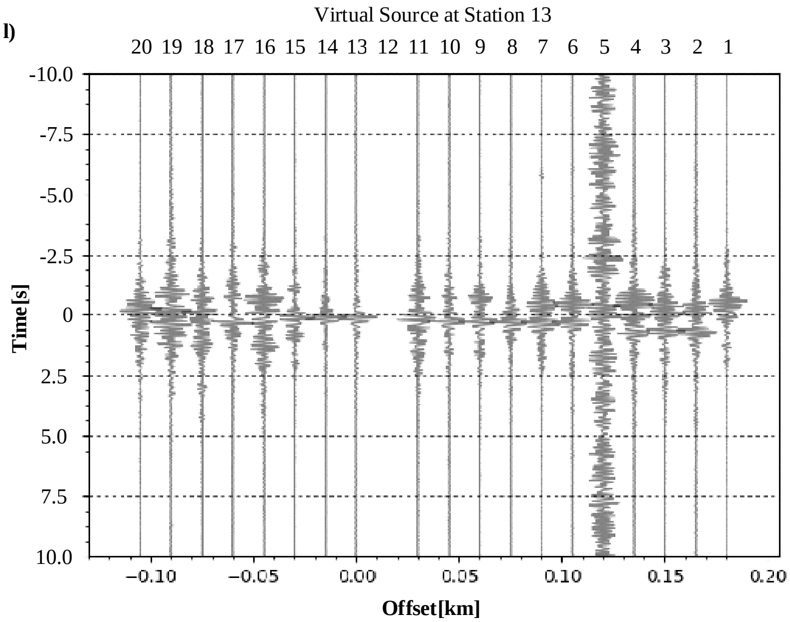
j)


Virtual Source at Station 10
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.15 −0.10 −0.05 0.00 0.05 0.10 0.15
Offset[km]

k)

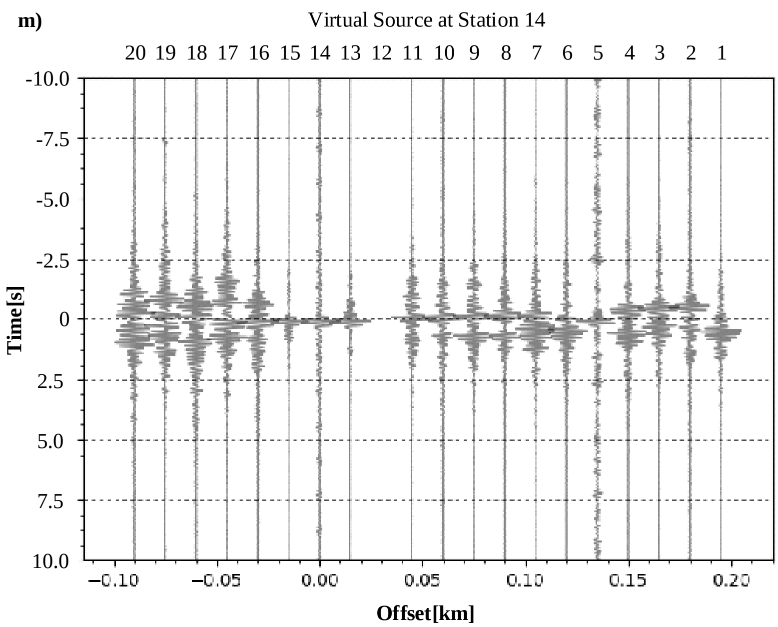

Virtual Source at Station 11
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.15 −0.10 −0.05 0.00 0.05 0.10 0.15
Offset[km]

l)



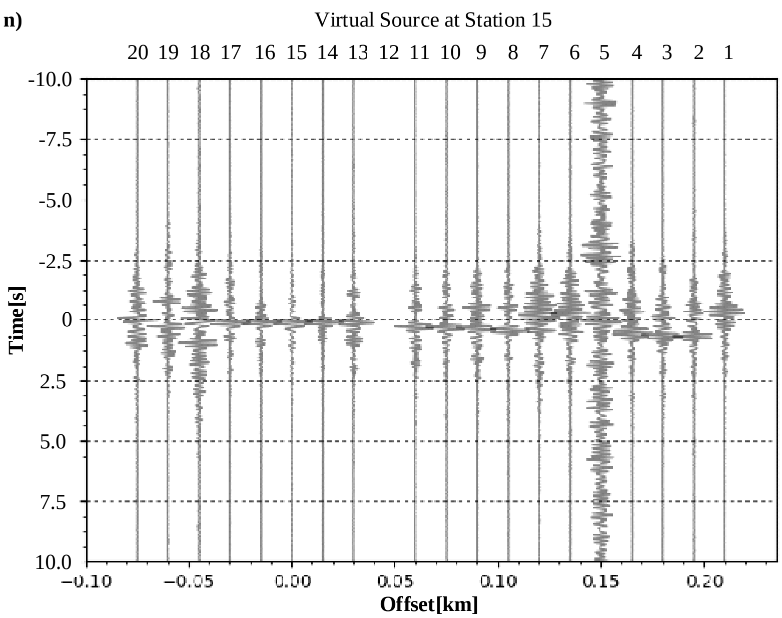
Virtual Source at Station 13
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
Offset[km]

m)

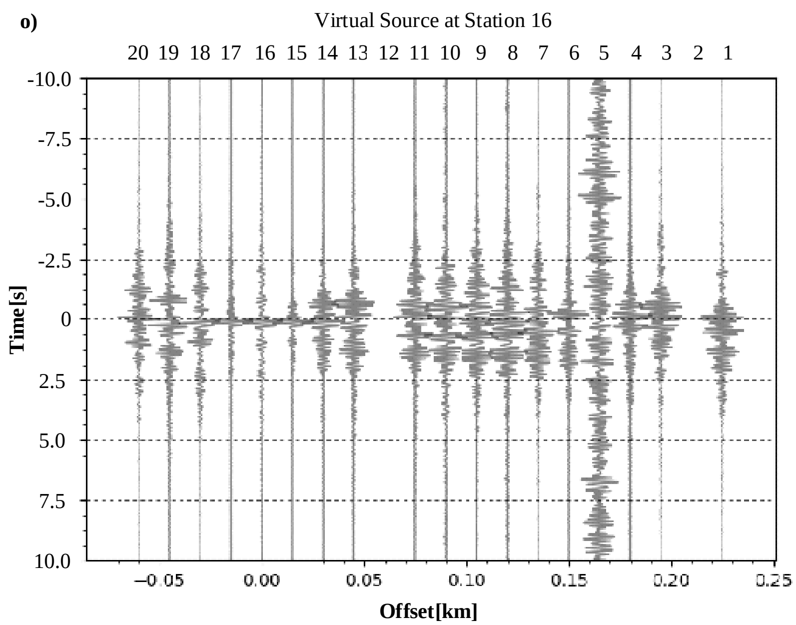


Virtual Source at Station 14
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
Offset[km]

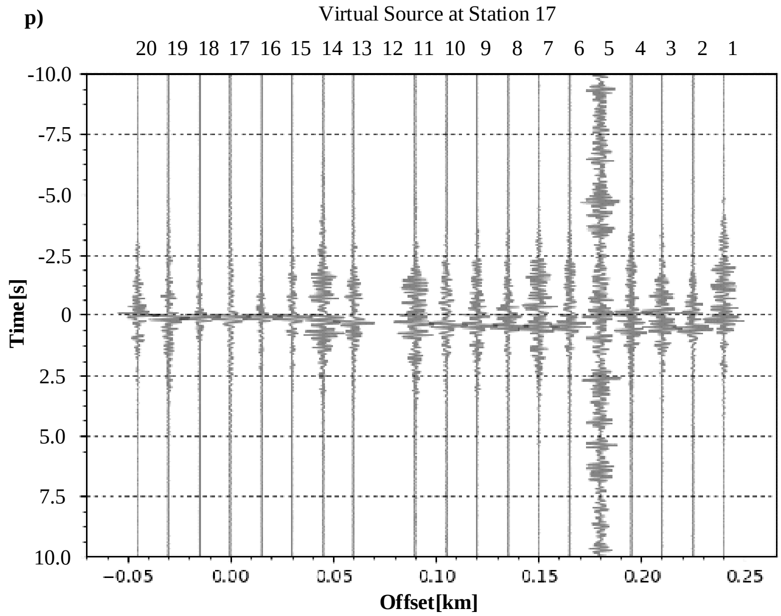
n)


Virtual Source at Station 15
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
Offset[km]

o)


Virtual Source at Station 16
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
Offset[km]

p)


Virtual Source at Station 17
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05 0.00 0.05 0.10 0.15 0.20 0.25
Offset[km]

q)

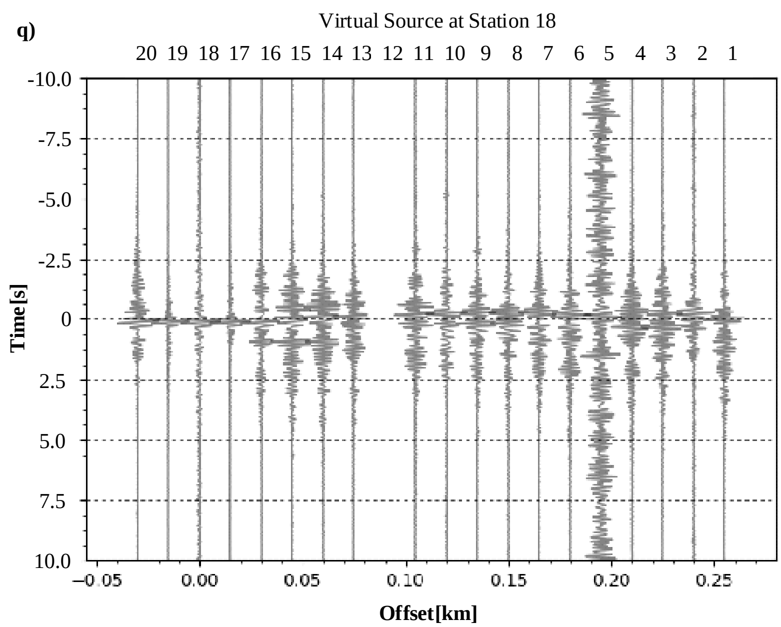
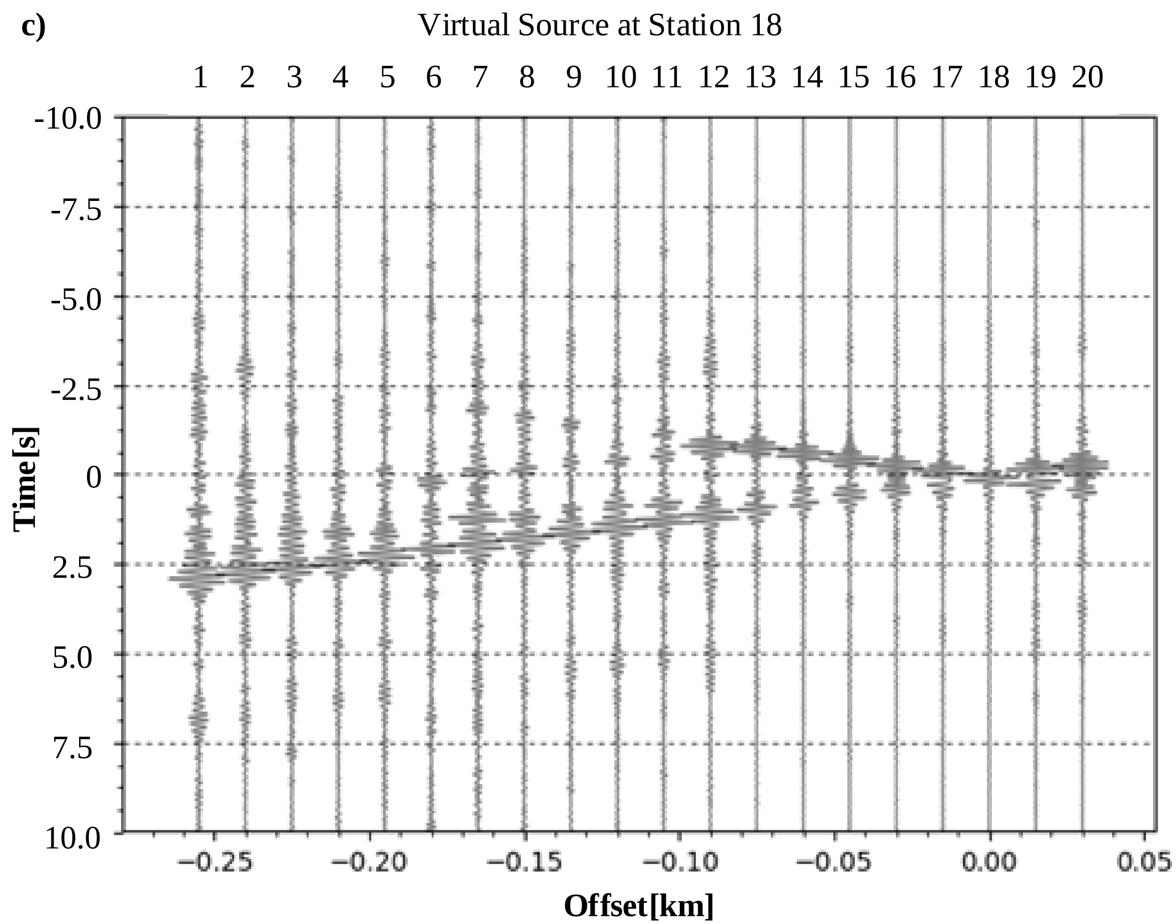
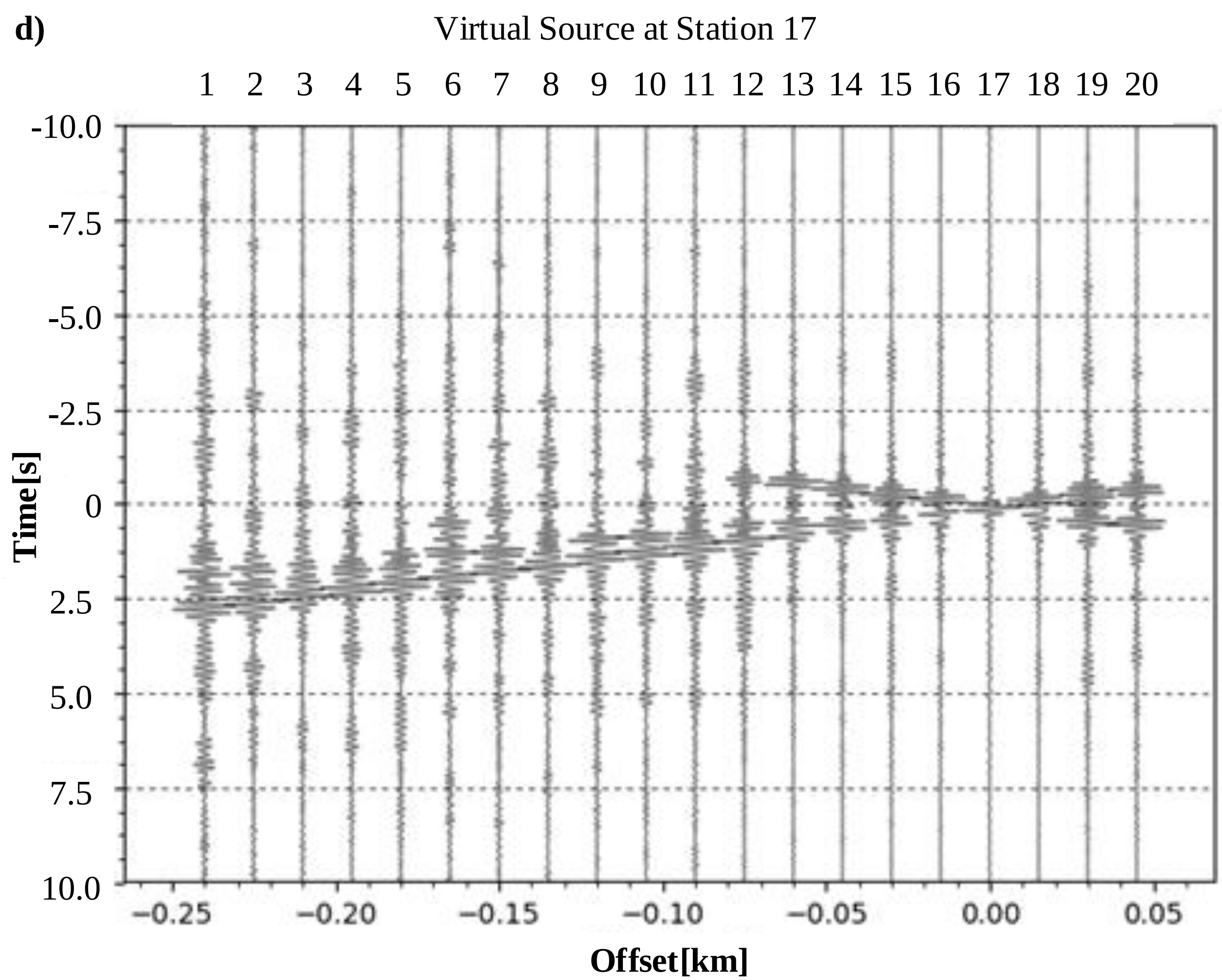
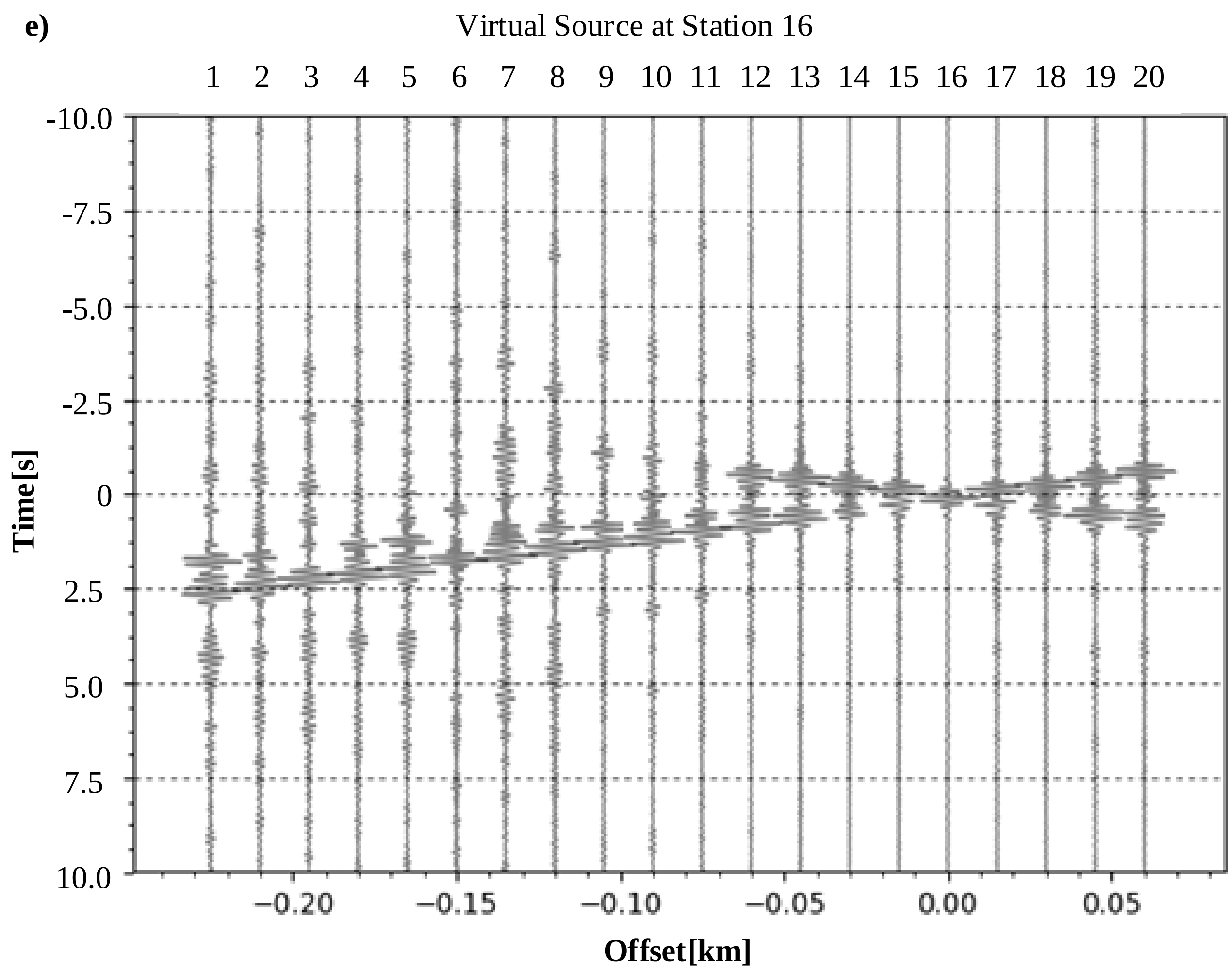

Virtual Source at Station 18
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05 0.00 0.05 0.10 0.15 0.20 0.25
Offset[km]

Figure A.2 (c-t). The following 18 final VSGs were produced automatically after 45 hours of data acquisition by the RaPiER array at Soda Lake Geothermal Field (Fallon, NV).

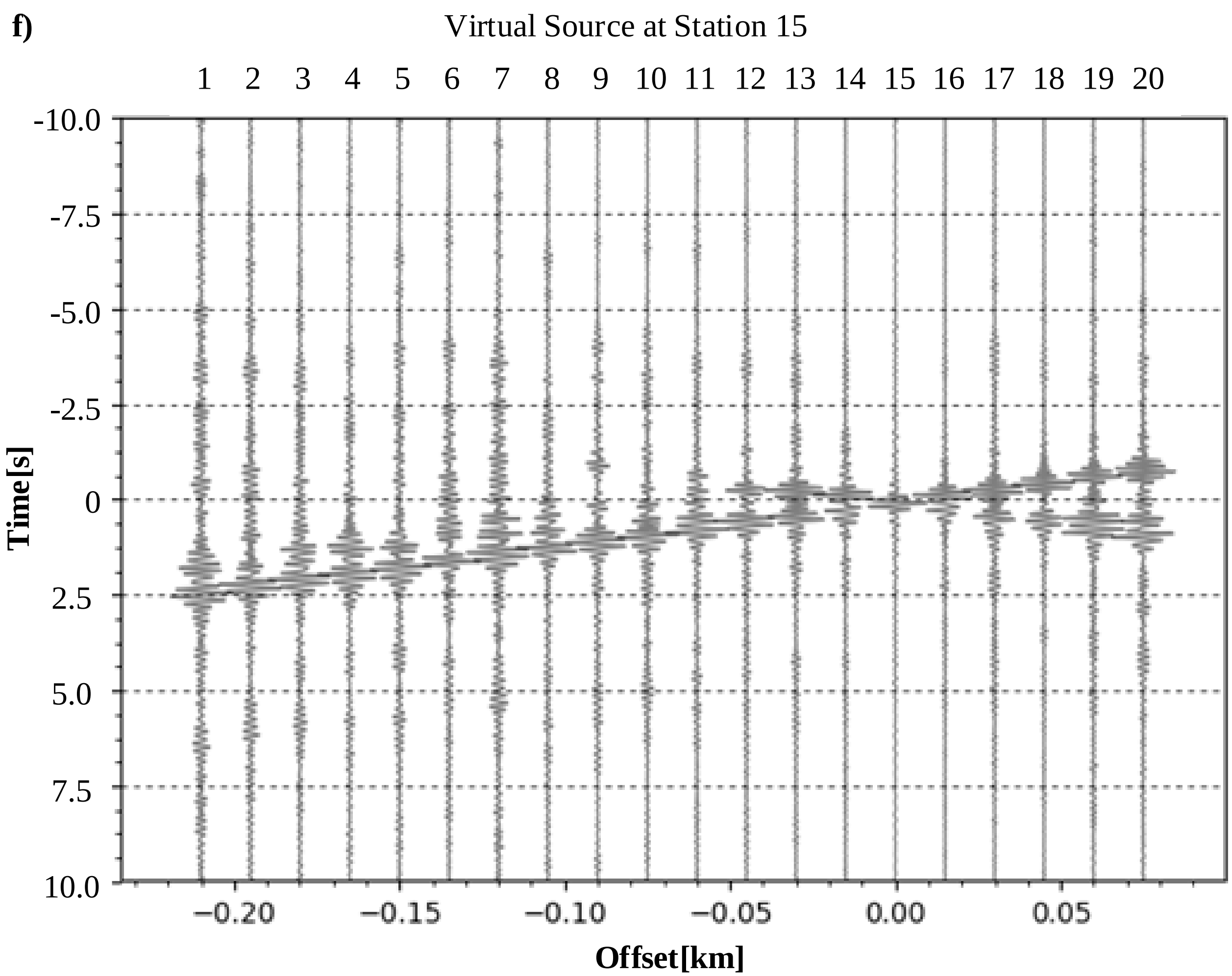



d)
Virtual Source at Station 17
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.25 −0.20 −0.15 −0.10 −0.05 0.00 0.05
Offset[km]

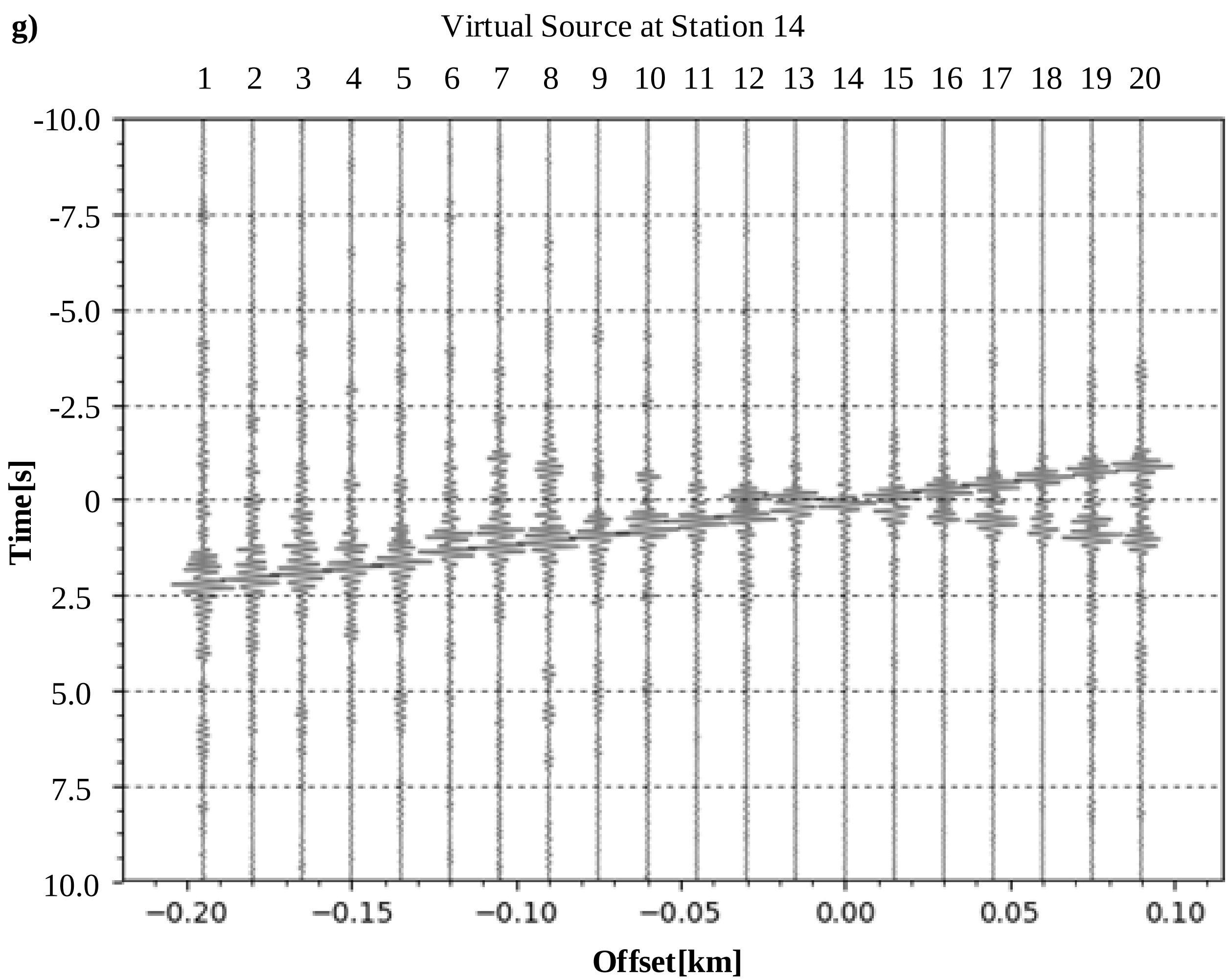
e)



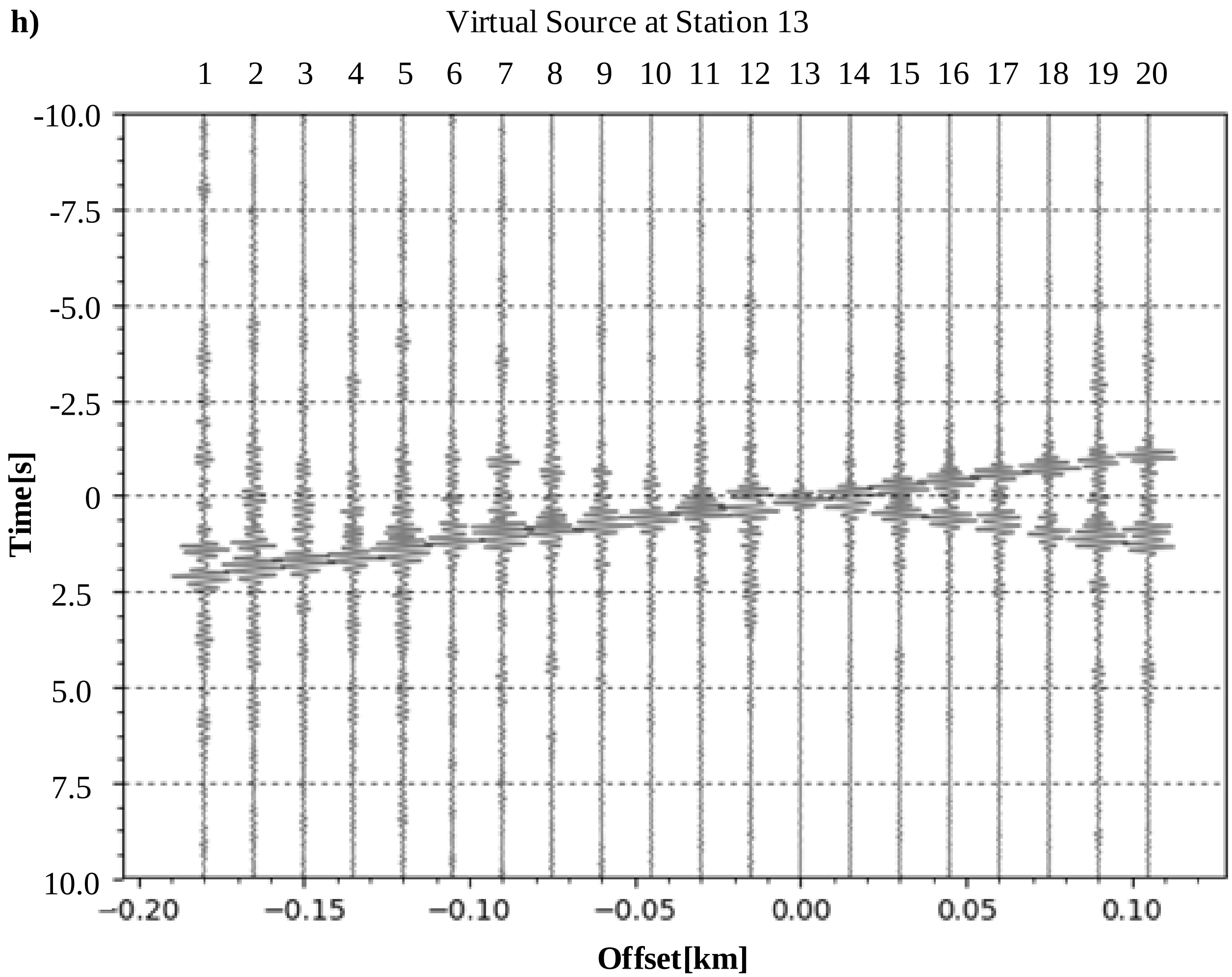
Virtual Source at Station 16
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.20
−0.15
−0.10
−0.05
0.00
0.05
Offset[km]


f)
Virtual Source at Station 15
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.20
−0.15
−0.10
−0.05
0.00
0.05
Offset[km]

g)

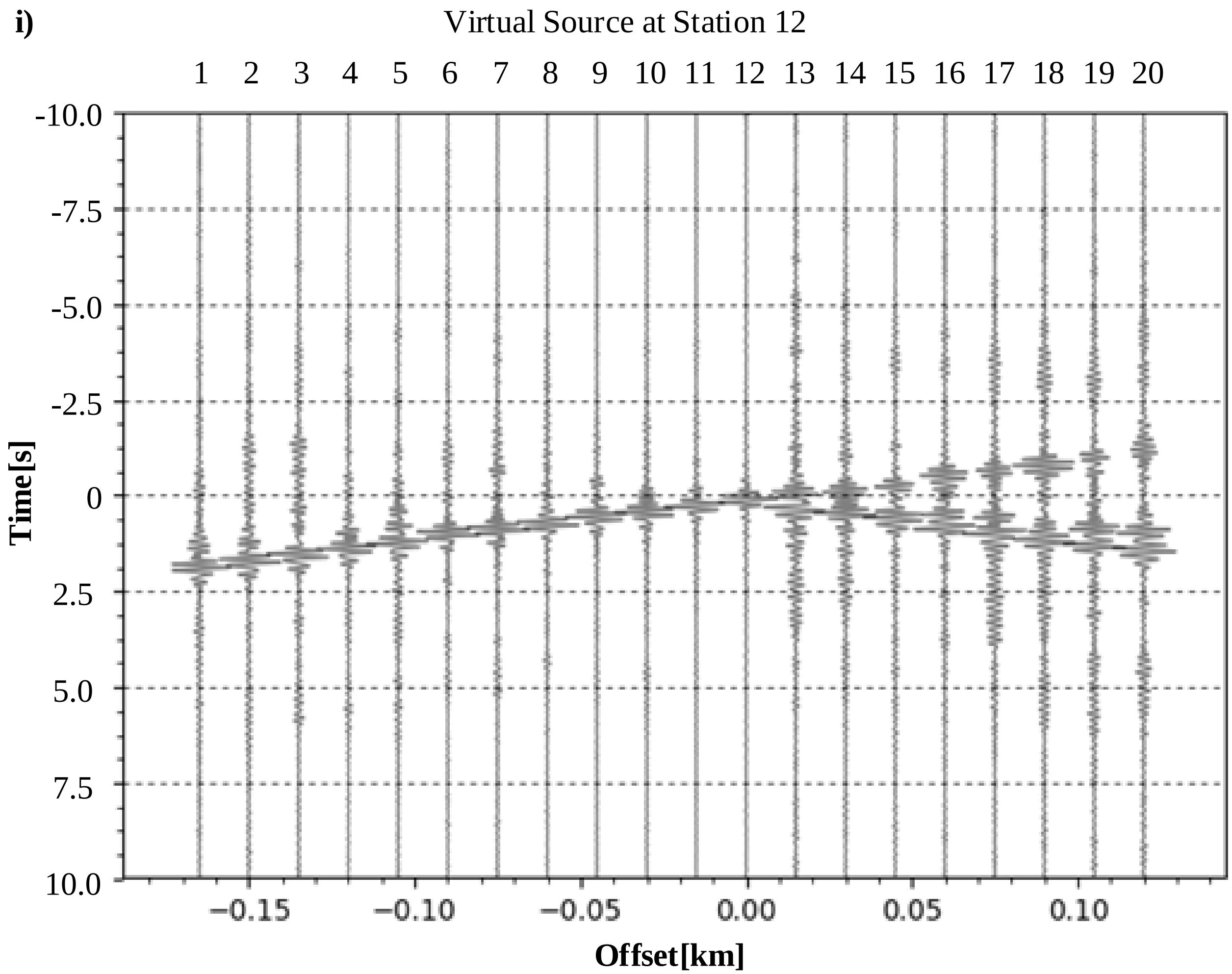

Virtual Source at Station 14
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.20 −0.15 −0.10 −0.05 0.00 0.05 0.10
Offset[km]

h)



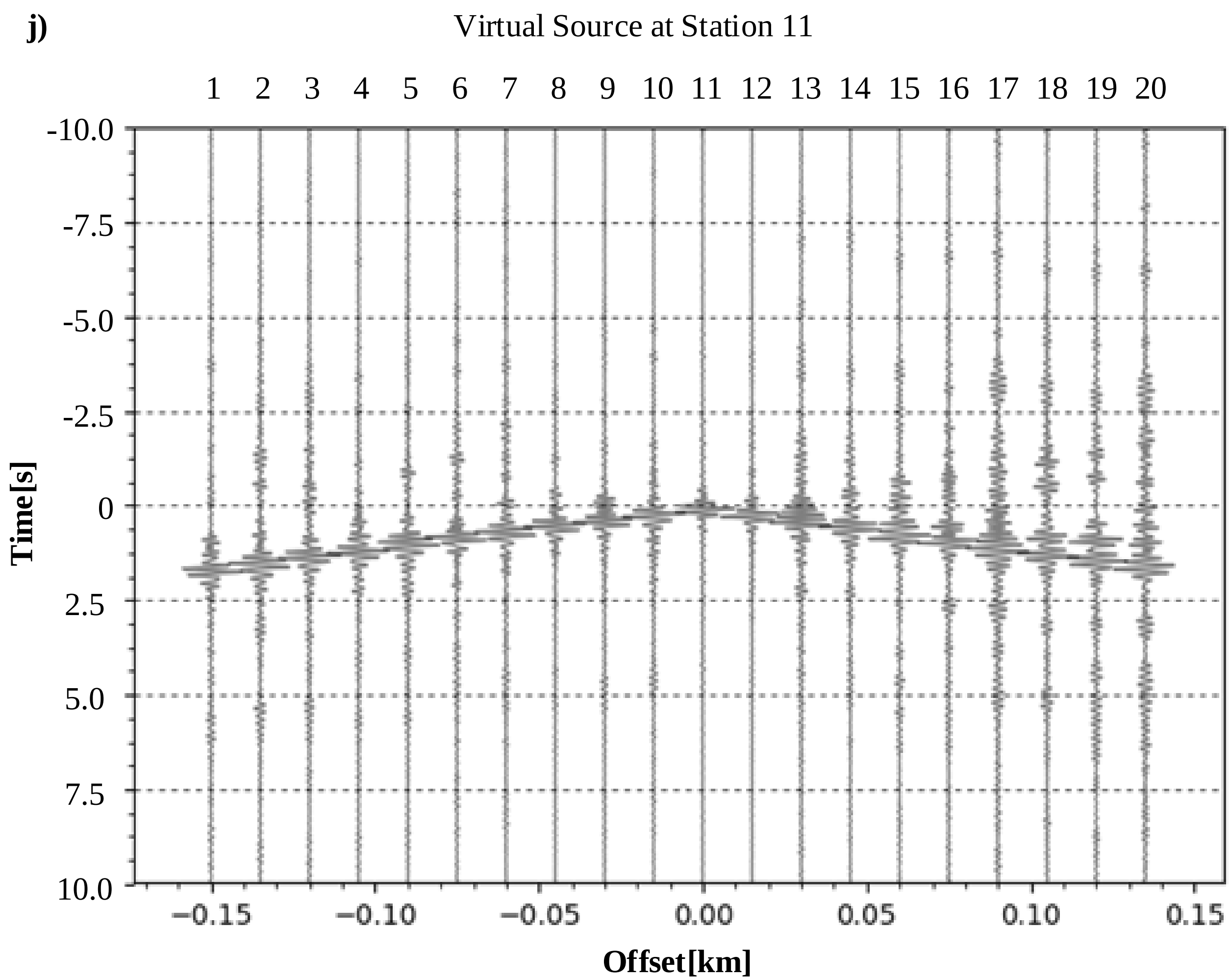
Virtual Source at Station 13
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
Offset[km]

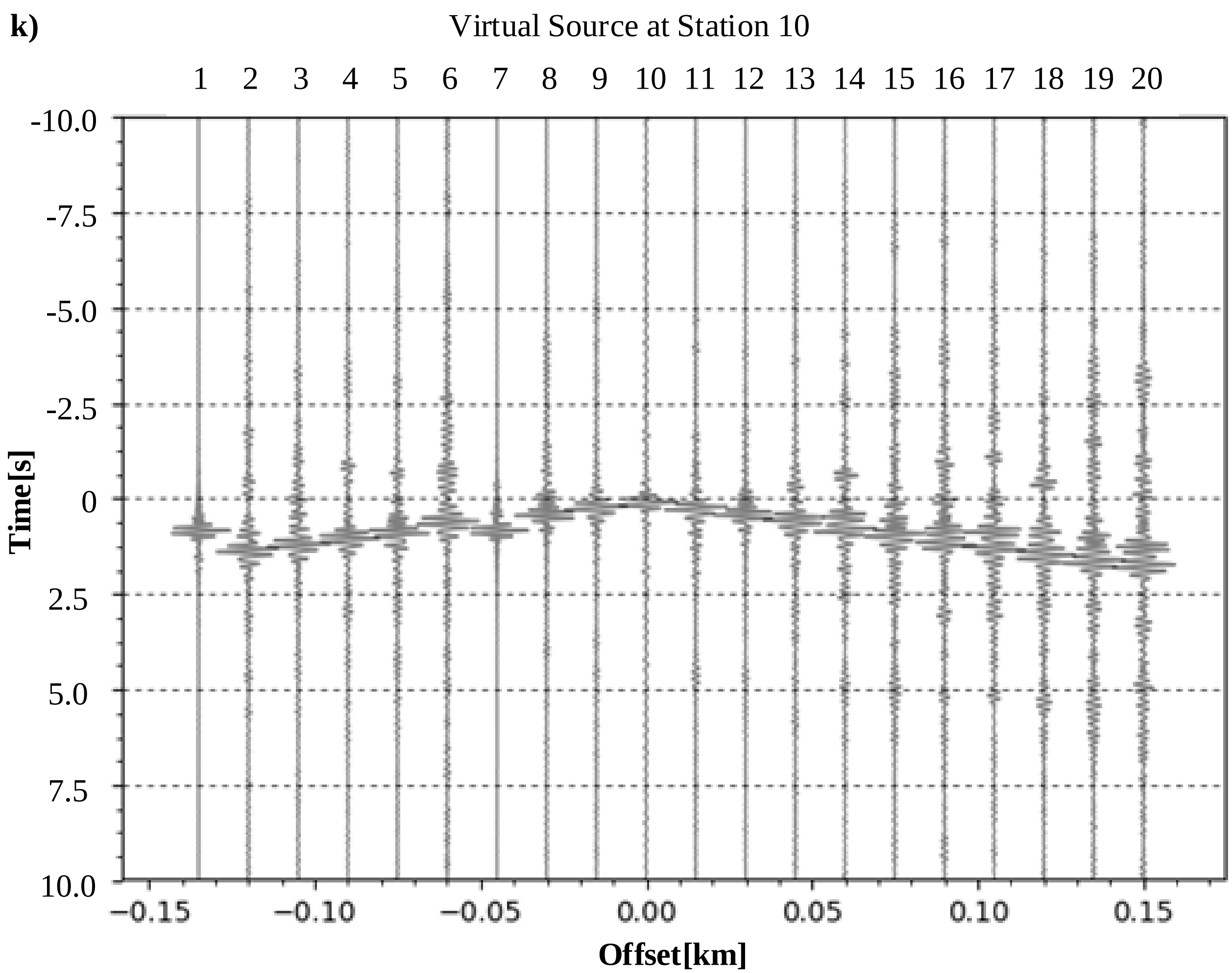
i)



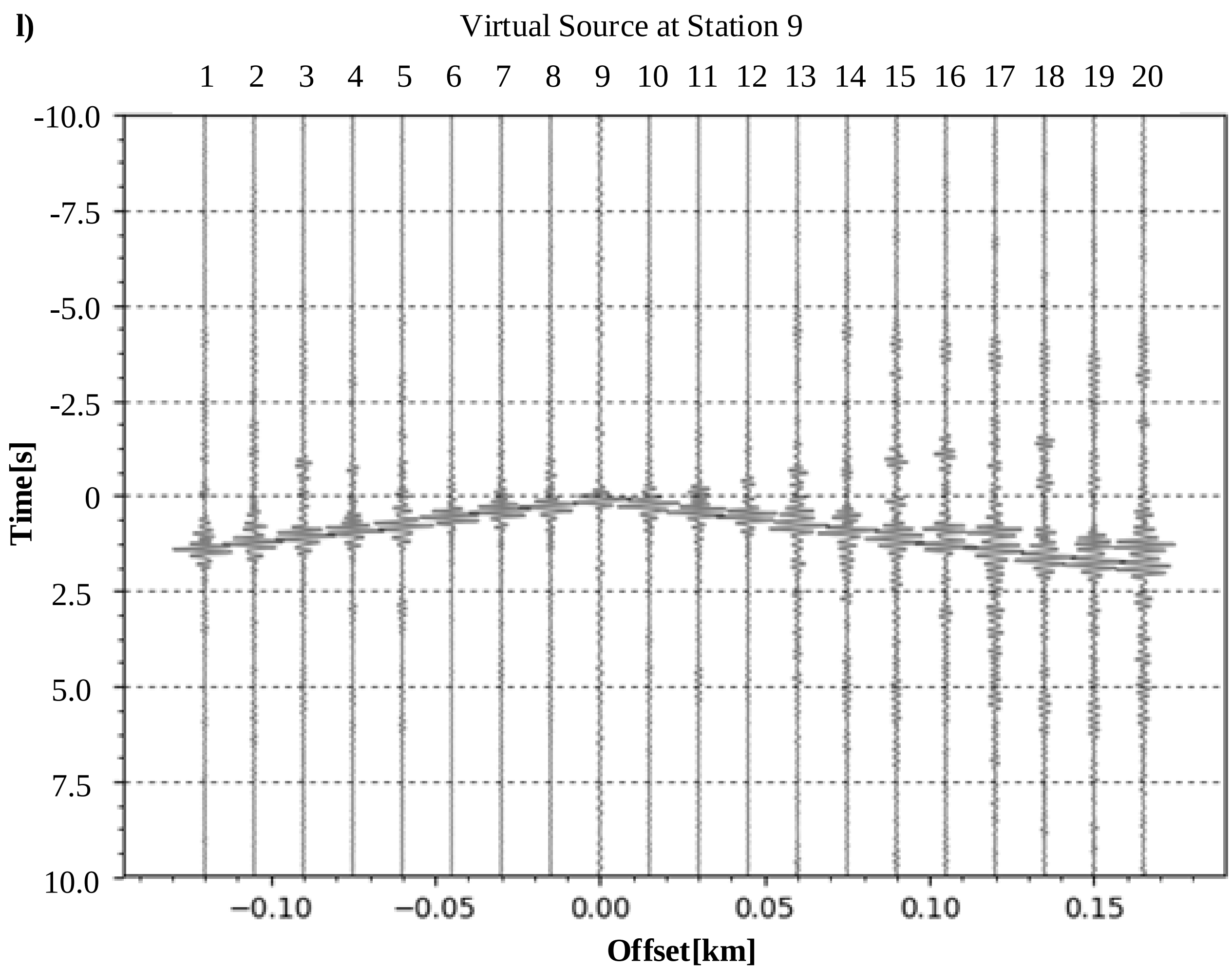
Virtual Source at Station 12
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.15
−0.10
−0.05
0.00
0.05
0.10
Offset[km]

j)

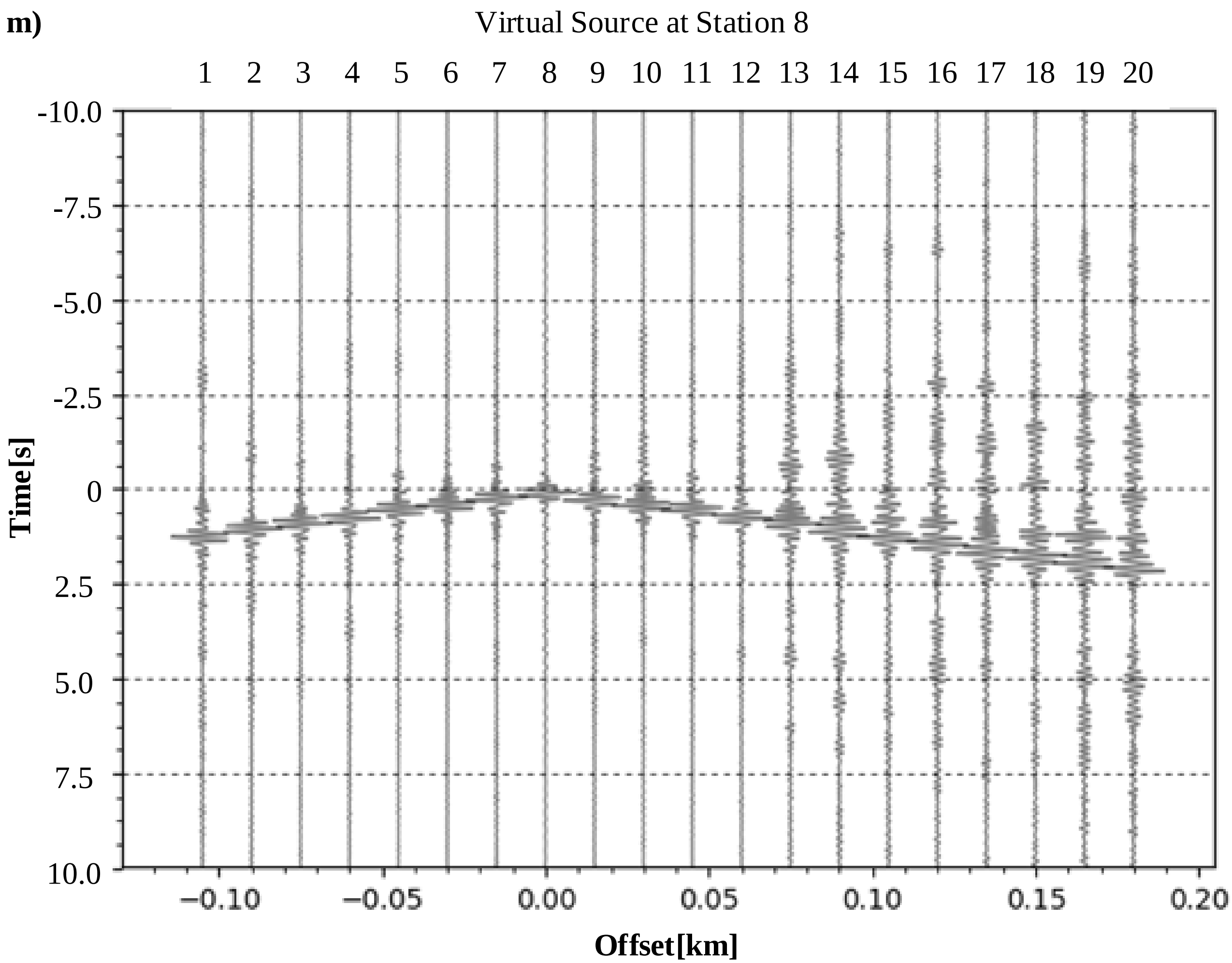

Virtual Source at Station 11
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.15 −0.10 −0.05 0.00 0.05 0.10 0.15
Offset[km]


k)
Virtual Source at Station 10
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.15 −0.10 −0.05 0.00 0.05 0.10 0.15
Offset[km]

l)

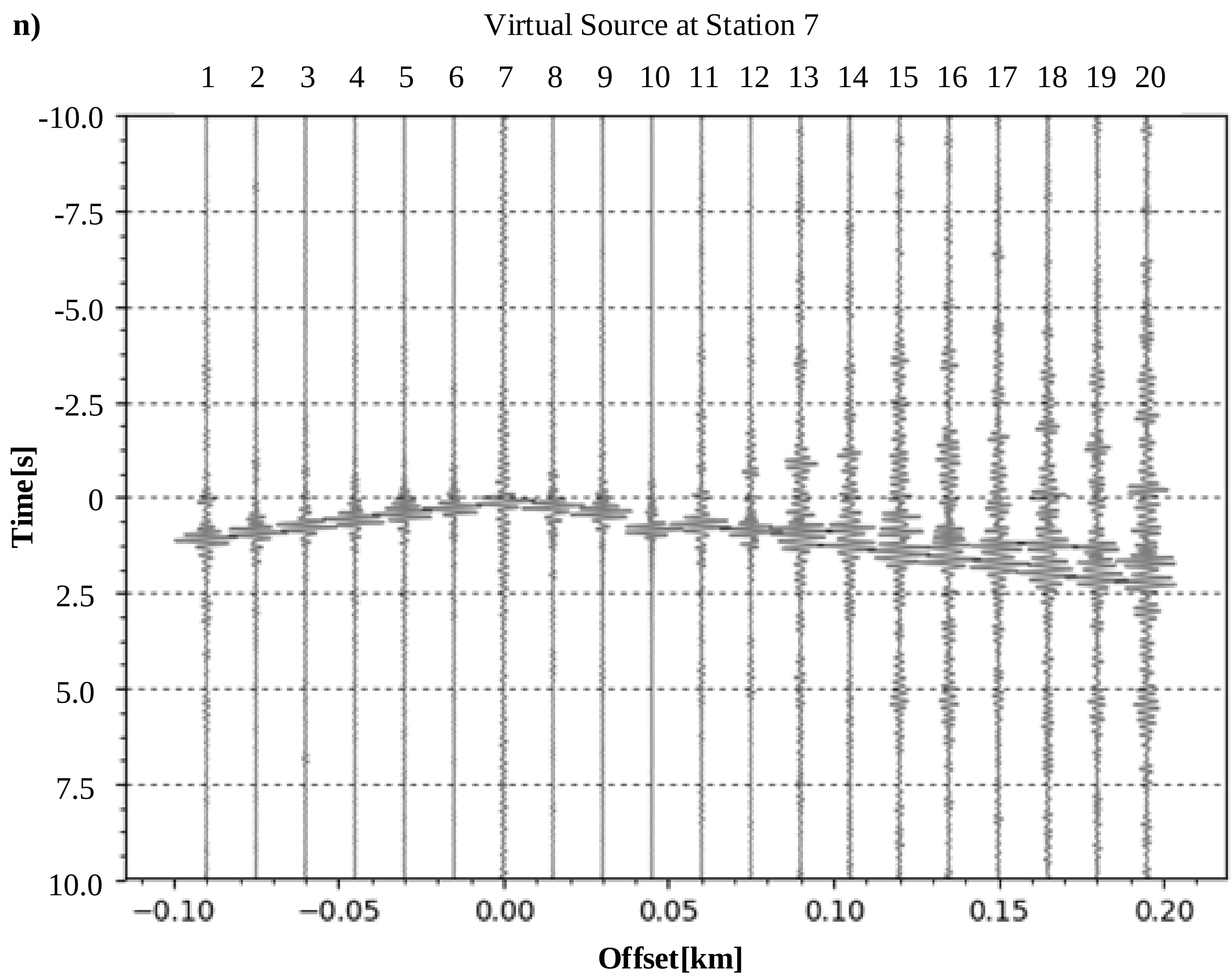

Virtual Source at Station 9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.10
−0.05
0.00
0.05
0.10
0.15
Offset[km]

m)

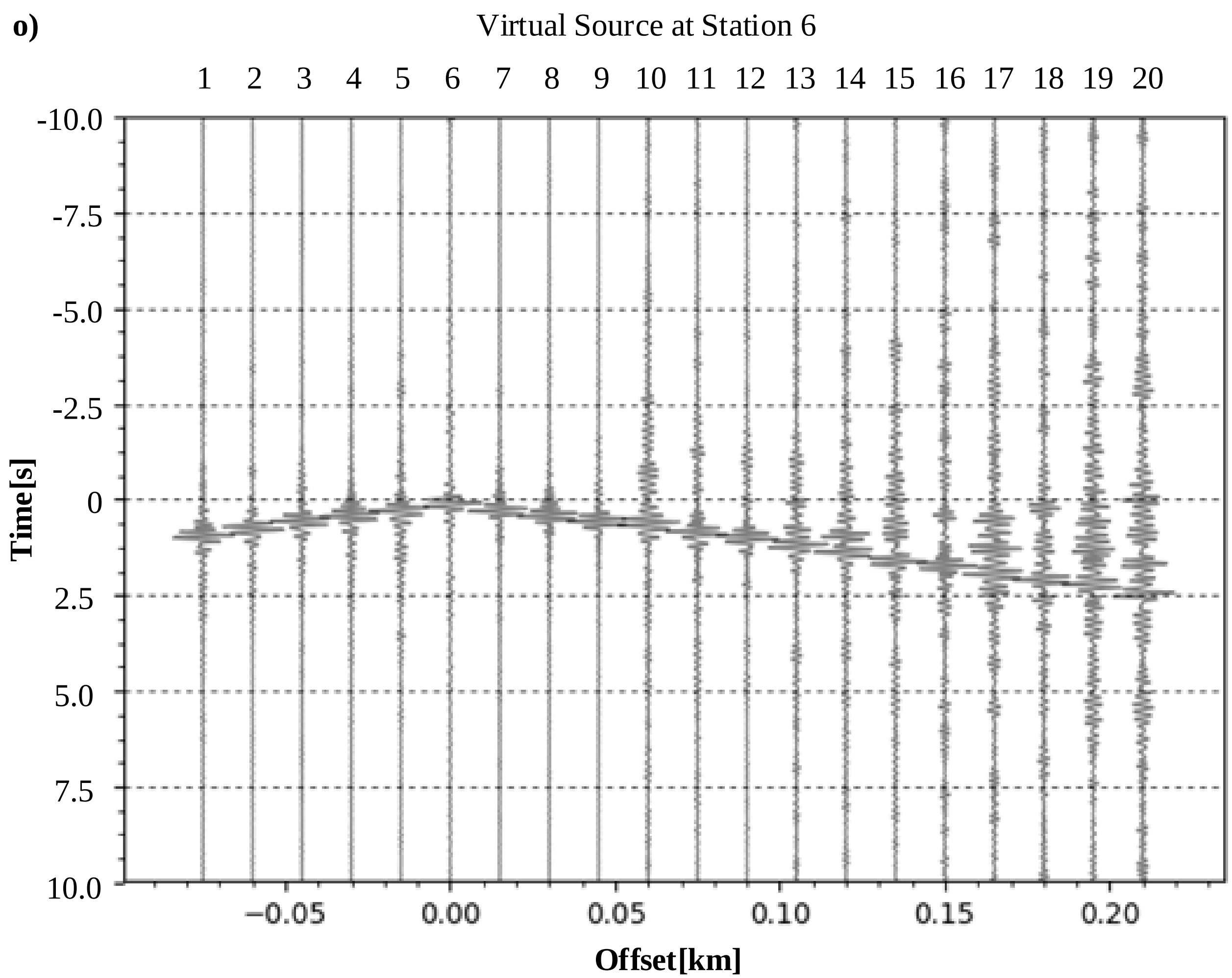


Virtual Source at Station 8
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.10 −0.05 0.00 0.05 0.10 0.15 0.20
Offset[km]

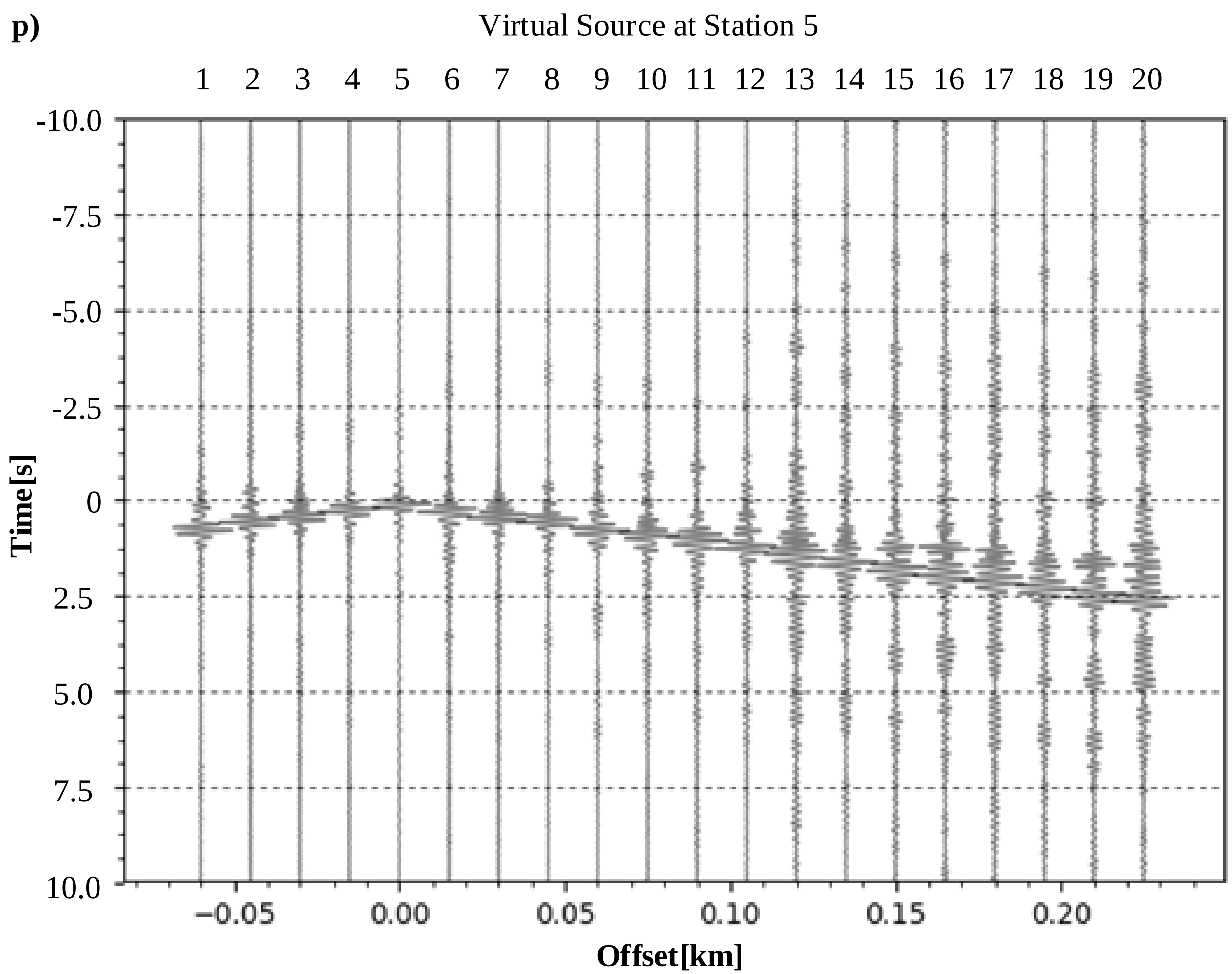
n)


Virtual Source at Station 7
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.10 −0.05 0.00 0.05 0.10 0.15 0.20
Offset[km]

o)

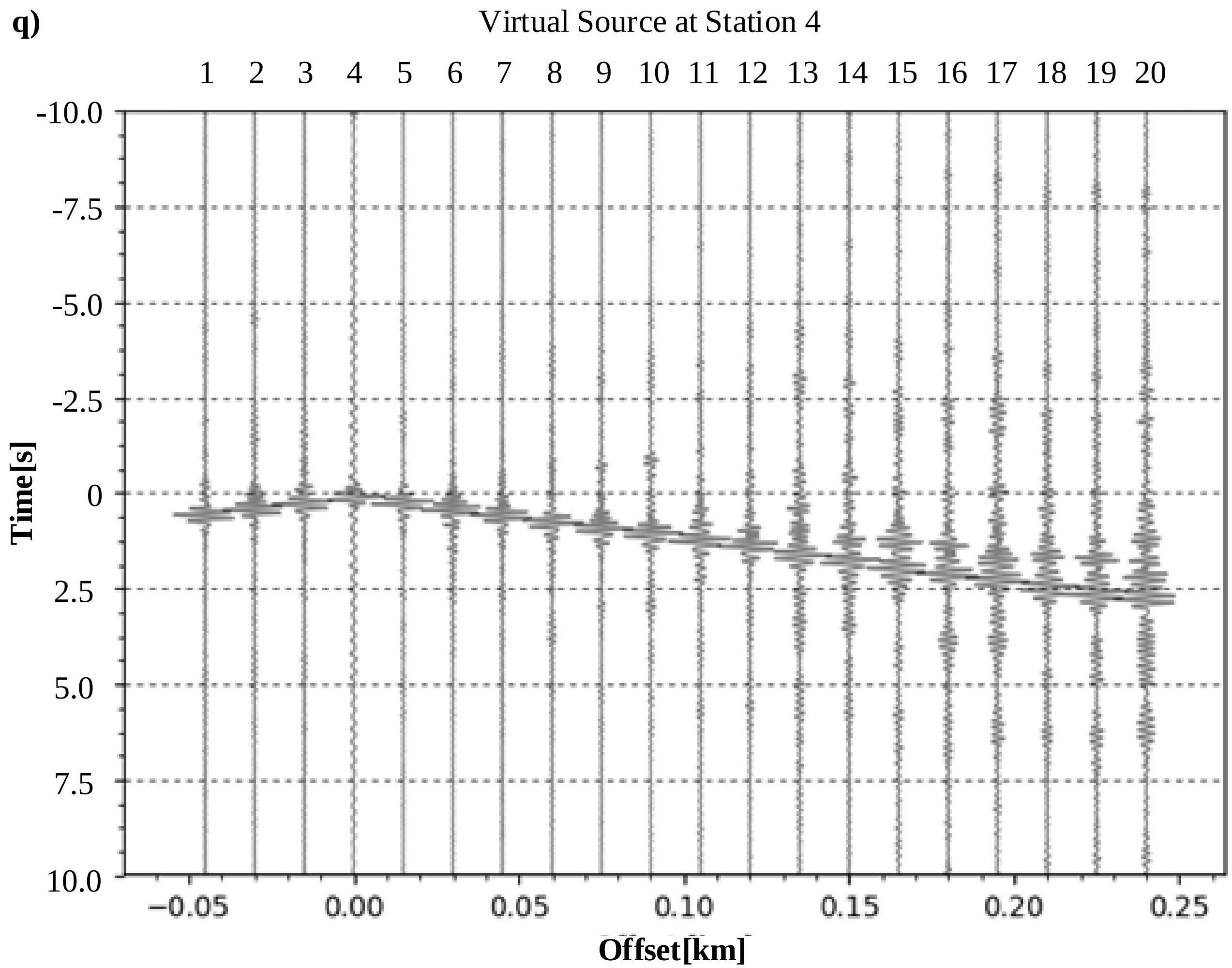

Virtual Source at Station 6
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05
0.00
0.05
0.10
0.15
0.20
Offset[km]

p)



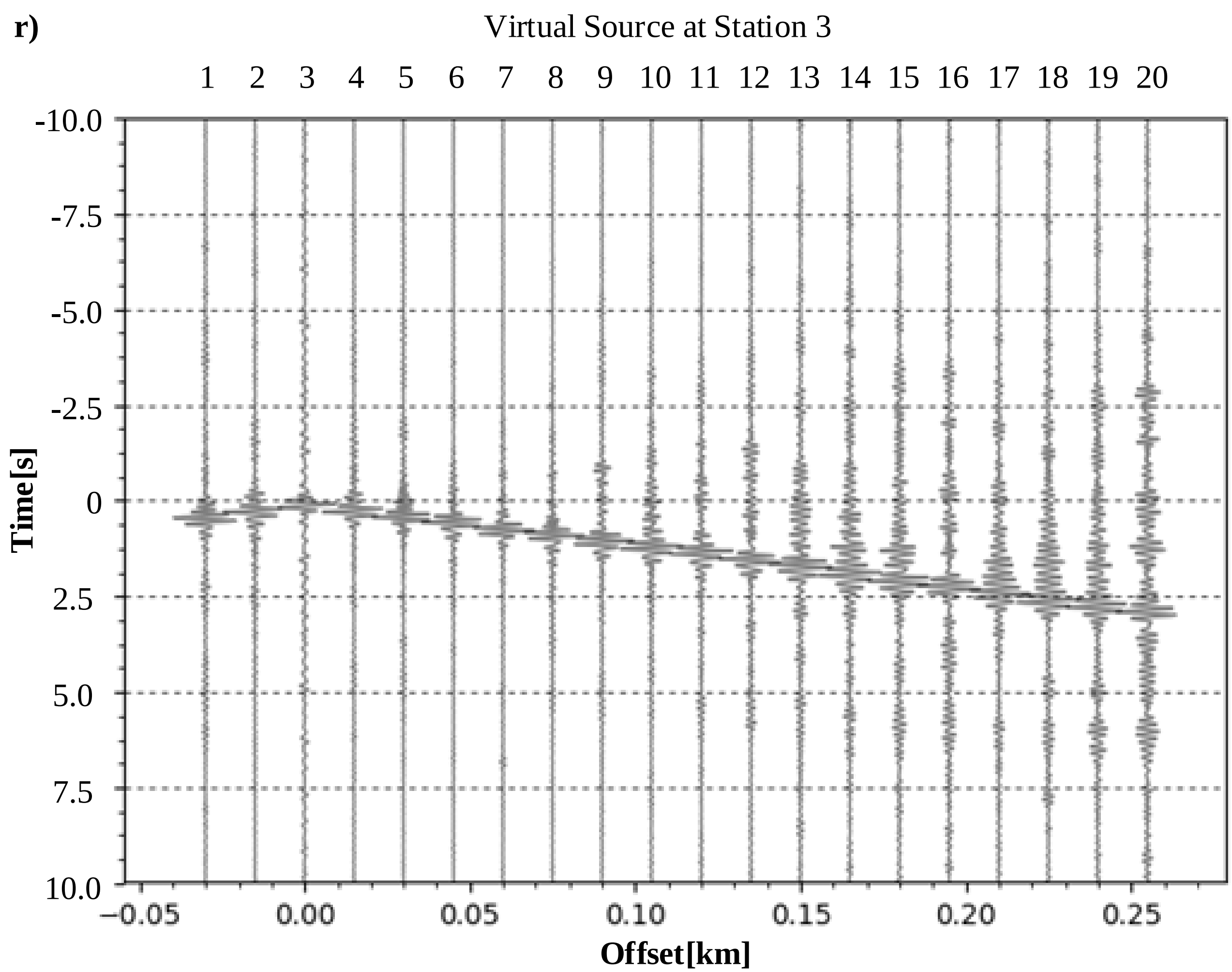
Virtual Source at Station 5
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05
0.00
0.05
0.10
0.15
0.20
Offset[km]

q)

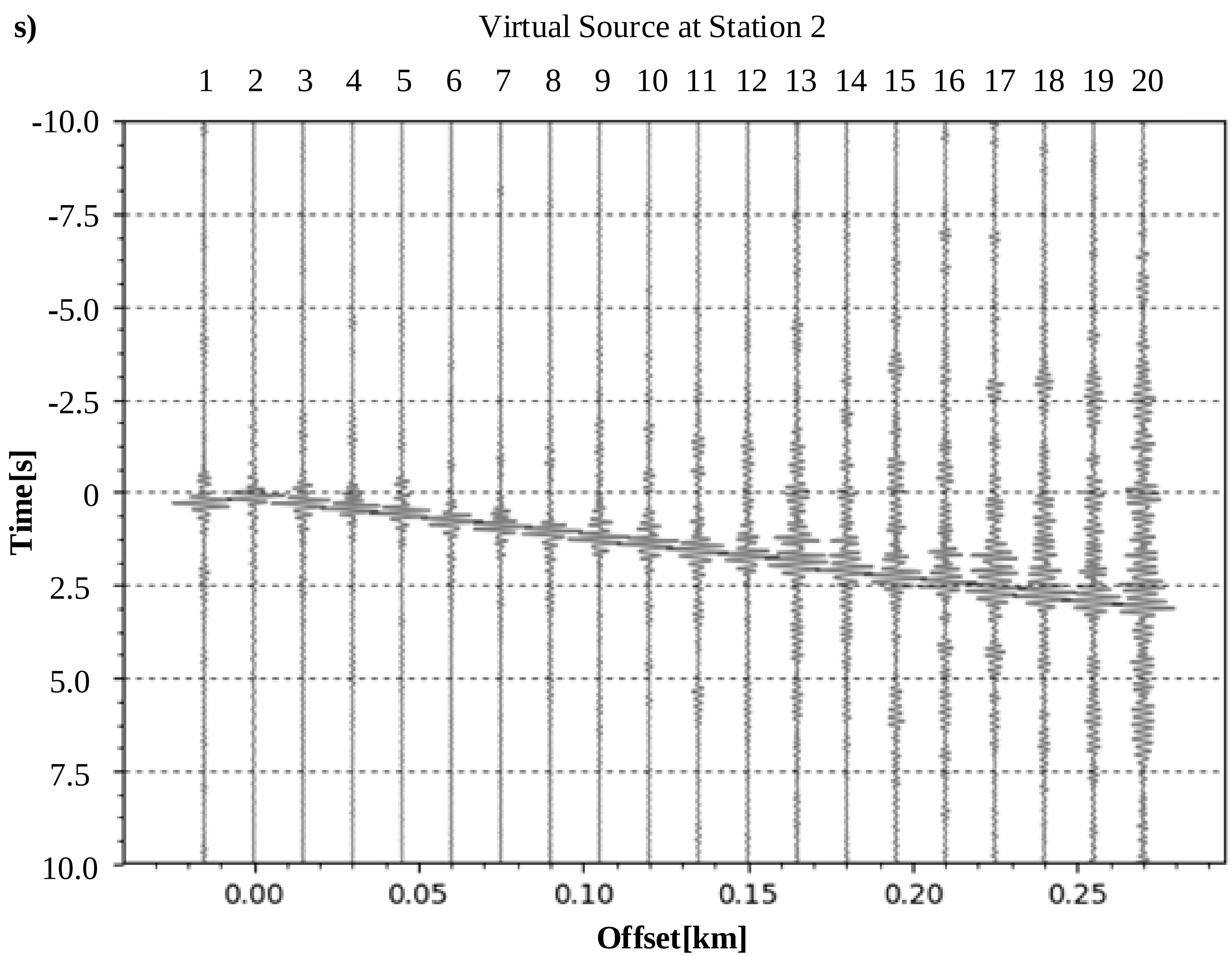


Virtual Source at Station 4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05 0.00 0.05 0.10 0.15 0.20 0.25
Offset[km]

r)


Virtual Source at Station 3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
Offset[km]

s)


Virtual Source at Station 2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
0.00
0.05
0.10
0.15
0.20
0.25
Offset[km]

t)

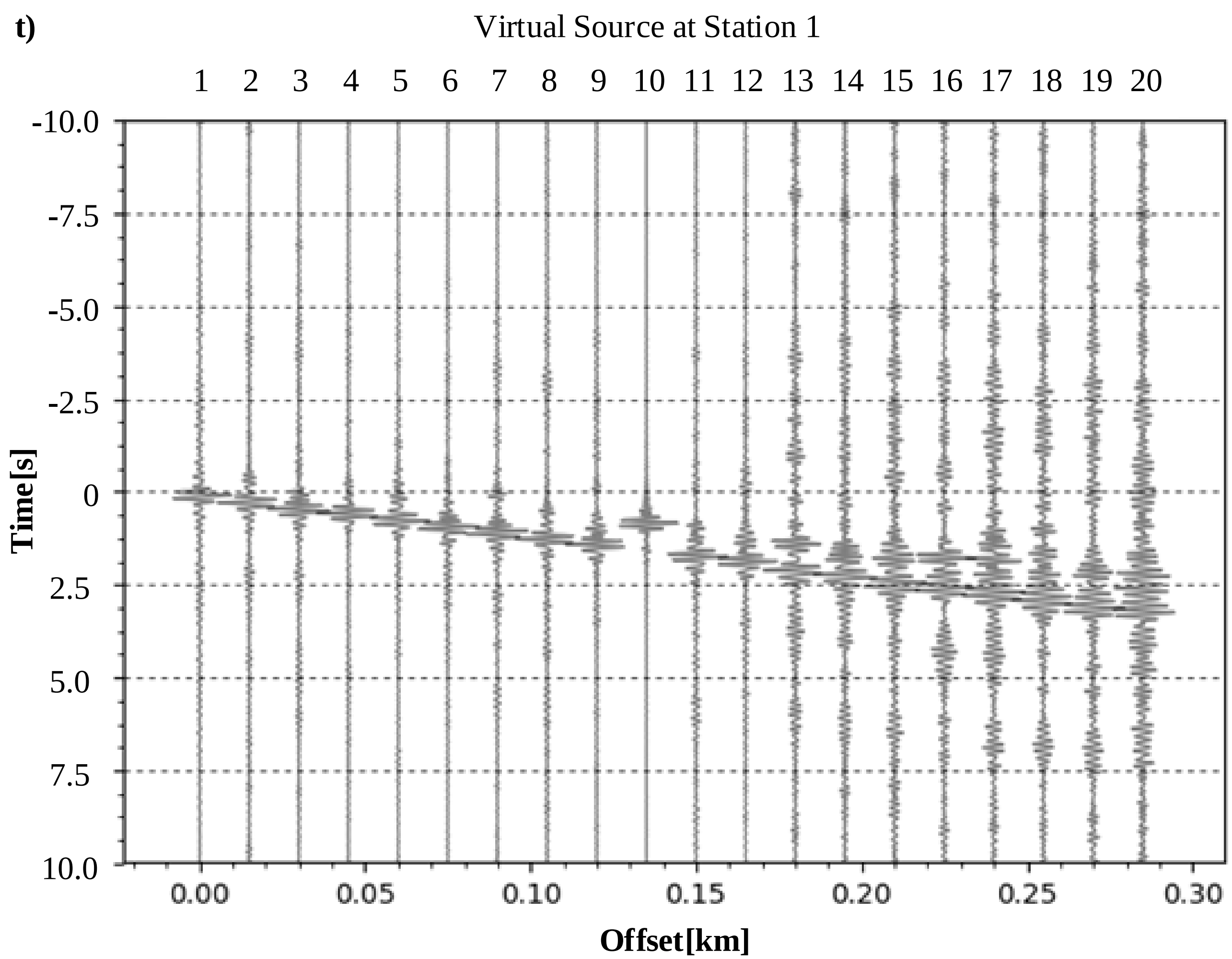
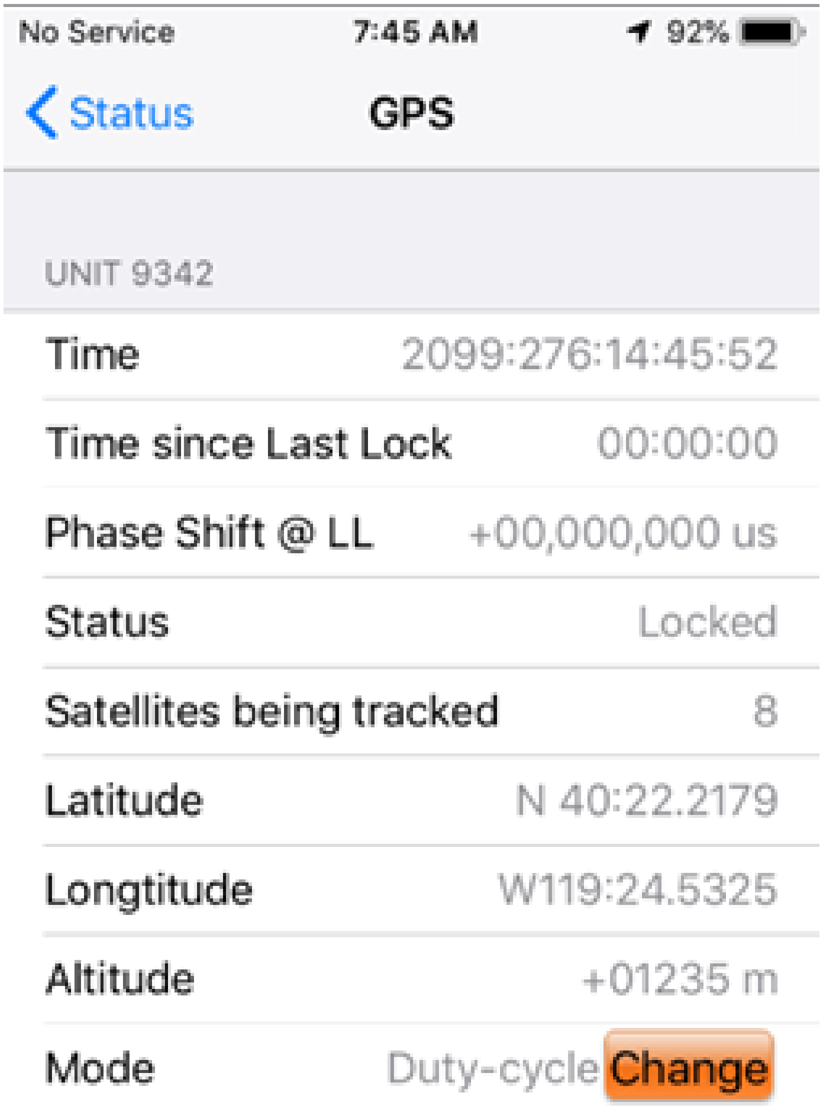

Virtual Source at Station 1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
-10.0
-7.5
-5.0
-2.5
0
2.5
5.0
7.5
10.0
Time[s]
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Offset[km]

Figure A.3. San Emidio T&E event GPS week number rollover issue.



REFERENCES

Abramova, V., Bernardino, J., & Furtado, P. (2014). Which NoSQL Database? A Performance Overview, *1*(2), 8.

Alcaraz, C., Najera, P., Lopez, J., & Roman, R. (2010). Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50. https://doi.org/10.1145/1721654.1721672

Ashton, K. (2009). That "Internet of Things" thing, *RFID J.*, available from http://www.rfidjournal.com/articles/view?4986 (last accessed April 2015).

Baldwin, H. (2014). A match made somewhere: Big Data and the Internet of Things, *Forbes*, available from http://www.forbes.com/sites/howardbaldwin/2014/11/24/a-match-made-somewhere-big-data-and-the-internet-of-things/ (last accessed June 2015).

Bertolucci, J. (2014). Internet of things "overhyped," say IT execs, *InformationWeek*, available from http://www.informationweek.com/big-data/big-data-analytics/internet-of-things-overhyped-say-it-execs/d/d-id/1317518 (last accessed July 2015).

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A Python Toolbox for Seismology, *Seismol. Res. Lett.* **81,** no. 3, 530–533, doi:10.1785/gssrl.81.3.530.

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, *56*, 684–700. https://doi.org/10.1016/j.future.2015.09.021

Bourdeau, J. (2012). Setup VSFTPD with custom multiple directories and (virtual) users accounts on Ubuntu (no database required), *Sigerr*., available from http://www.sigerr.org/linux/setup-vsftpd-custom-multiple-directories-users-accounts-ubuntu-step-by-step/ (last accessed July 2015).

Brewer, E. (2012). CAP twelve years later: How the "rules" have changed. *Computer*, *45*(2), 23–29. https://doi.org/10.1109/MC.2012.37

Carpenter, J., & Hewitt, E. (2016). *Cassandra: the definitive guide* (Second edition). Sebastopol, CA: O'Reilly Media, Inc.

Challen, G., and M. Welsh (2010). Volcano monitoring: Addressing data quality through iterative deployment, in *Wireless Sensor Networks*, E. Gaura (Editor), Springer, New York, New York, 71–113.

Chambers, B., & Zaharia, M. (2018). *Spark: the definitive guide: big data processing made simple* (First edition). Sebastapol, CA: O'Reilly Media.

Chen, C. L. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, *275*(Supplement C), 314–347. https://doi.org/10.1016/j.ins.2014.01.015

Chung, M., & Kim, J. (2016). The Internet Information and Technology Research Directions based on the Fourth Industrial Revolution. *KSII Transactions on Internet and Information Systems*, *10*, 1311–1320. https://doi.org/10.3837/tiis.2016.03.020

Clemente, J., Li, F., Valero, M., Chen, A., & Song, W. (2020). ASIS: Autonomous Seismic Imaging System With In Situ Data Analytics and Renewable Energy. *IEEE Systems Journal*, *14*(1), 1277–1284. https://doi.org/10.1109/JSYST.2019.2920073

Confais, B., Lebre, A., & Parrein, B. (2016). Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 294–301). https://doi.org/10.1109/CloudCom.2016.0055

Corke, P., Wark, T., Jurdak, R., Hu, W., Valencia, P., & Moore, D. (2010). Environmental Wireless Sensor Networks. *Proceedings of the IEEE*, *98*(11), 1903–1917. https://doi.org/10.1109/JPROC.2010.2068530

Cossick, S. (2019, July 16). Here's what you need to know about a Wi-Fi extender. Retrieved July 31, 2019, from https://www.allconnect.com/blog/3-reasons-house-needs-wi-fi-extender/

Cukier, K. N., and V. Mayer-Schoenberger (2013). The rise of big data, *Foreign Aff.*, available from https://www.foreignaffairs.com/articles/2013-04-03/rise-big-data (last accessed June 2015).

Deshmukh, A. D., & Shinde, U. B. (2016). A low cost environment monitoring system using raspberry Pi and arduino with Zigbee. In *2016 International Conference on Inventive Computation Technologies (ICICT)* (Vol. 3, pp. 1–6). https://doi.org/10.1109/INVENTIVE.2016.7830096

Duarte, A., & Bernardino, J. (2016). Cassandra for Internet of Things: An Experimental Evaluation. In *IoTBD*. https://doi.org/10.5220/0005846400490056

Eames, A. (2015). Raspberry Pi2: Power and performance measurement, *RasPi.TV*, available from http://raspi.tv/2015/raspberry-pi2-power-and-performance-measurement (last accessed August 2015).

Estrin, D., Govindan, R., Heidemann, J., & Kumar, S. (1999). Next century challenges: scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99* (pp. 263–270). Seattle, Washington, United States: ACM Press. https://doi.org/10.1145/313451.313556

Ferencz, K., & Domokos, J. (2018). IoT Sensor Data Acquisition and Storage System Using Raspberry Pi and Apache Cassandra. In *2018 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)* (pp. 000143–000146). https://doi.org/10.1109/CANDO-EPE.2018.8601139

Fratu, O., Pejanovic-Djurisic, M., Poulkov, V., & Gavrilovska, L. (2016). Introduction to Special Issue "Current Trends in Information and Communications Technology." *Wireless Personal Communications*, *87*(3), 615–617. https://doi.org/10.1007/s11277-016-3228-6

Gabbai, A. (2015). Kevin Ashton describes, *Smithson. Mag.*, available from http://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/ (last accessed April 2015).

Gaber, M. M., Aneiba, A., Basurra, S., Batty, O., Elmisery, A. M., Kovalchuk, Y., & Rehman, M. H. U. (2019). Internet of Things and data mining: From applications to techniques and systems. *WIREs Data Mining and Knowledge Discovery*, *9*(3), e1292. https://doi.org/10.1002/widm.1292

Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., et al. (2015). Edge-centric Computing: Vision and Challenges. *SIGCOMM Comput. Commun. Rev.*, *45*(5), 37–42. https://doi.org/10.1145/2831347.2831354

Gibbs, M. (2015). Ten reasons why the new Raspberry Pi 2 model B SBC is a killer product, *Networkworld*, available from http://www.networkworld.com/article/2887224/tech-primers/10-reasons-why-the-raspberry-pi-2-model-b-is-a-killer-product.html (last accessed July 2015).

Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, *33*(2), 51–59. https://doi.org/10.1145/564585.564601

———. (2012). Perspectives on the CAP Theorem. *Computer*, *45*(2), 30–36. https://doi.org/10.1109/MC.2011.389

Gnawali, O., Jang, K.-Y., Paek, J., Vieira, M., Govindan, R., Greenstein, B., et al. (2006). The Tenet Architecture for Tiered Sensor Networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems* (pp. 153–166). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1182807.1182823

Greengard, S. (2015). *The internet of things*. Cambridge, Massachusetts: MIT Press.

Hall, M., and E. Bianco (2012). *52 Tthings You Should Know about Geophysics*, Agile Libre, Mahone Bay, Nova Scotia.

Hamilton, E. (2018, December 31). What is Edge Computing: The Network Edge Explained. Retrieved July 16, 2019, from https://www.cloudwards.net/what-is-edge-computing/

Heath, N. (2019, December). How the Raspberry Pi was created: A visual history of the $35 board. Retrieved August 6, 2019, from https://www.techrepublic.com/pictures/how-the-raspberry-pi-was-created-a-visual-history-of-the-35-board/

Ilyas, M., & Mahgoub, I. (Eds.). (2006). *Sensor network protocols*. Boca Raton: CRC/Talor & Francis.

Jamali-Rad, H., & Campman, X. (2018). Internet of Things-based wireless networking for seismic applications. *Geophysical Prospecting*, *66*(4), 833–853. https://doi.org/10.1111/1365-2478.12617

Jamali-Rad, H., Campman, X., MacKay, I., Walk, W., Beker, M., van den Brand, J., et al. (2018). IoT-based wireless seismic quality control. *The Leading Edge*, *37*(3), 214–221. https://doi.org/10.1190/tle37030214.1

Jan, M. A., Khan, F., & Alam, M. (2019). *Recent Trends and Advances in Wireless and IoT-enabled Networks*. Retrieved from https://doi.org/10.1007/978-3-319-99966-1

Jones, C. M. (2018). The oil and gas industry must break the paradigm of the current exploration model. *Journal of Petroleum Exploration and Production Technology*, *8*(1), 131–142. https://doi.org/10.1007/s13202-017-0395-2

Kalinich, S. (2010, January 4). Wireless Distribution System - Is Wireless ready to replace the Wired network? Retrieved August 4, 2019, from https://www.tweaktown.com/articles/3085/wireless_distribution_system_is_wireless_ready_to_replace_the_wired_network/index2.html

Kamath, G., Agnihotri, P., Valero, M., Sarker, K., & Song, W.-Z. (2016). Pushing Analytics to the Edge. In *2016 IEEE Global Communications Conference (GLOBECOM)* (pp. 1–6). https://doi.org/10.1109/GLOCOM.2016.7842181

Karatas, M., & Onggo, B. S. (2019). Optimising the barrier coverage of a wireless sensor network with hub-and-spoke topology using mathematical and simulation models. *Computers & Operations Research*, *106*, 36–48. https://doi.org/10.1016/j.cor.2019.02.007

Kelly, G. (2014, December). 802.11ac vs 802.11n WiFi: What's The Difference? Retrieved July 31, 2019, from https://www.forbes.com/sites/gordonkelly/2014/12/30/802-11ac-vs-802-11n-wifi-whats-the-difference/

Klincewicz, J. G. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, *6*(1), 307–335. https://doi.org/10.1016/S0966-8349(98)00042-4

Kocakulak, M., & Butun, I. (2017). An overview of Wireless Sensor Networks towards internet of things. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 1–6). https://doi.org/10.1109/CCWC.2017.7868374

Koegler, S. (2019, February 7). 3 IoT Edge Storage Considerations [Text]. Retrieved August 7, 2019, from https://www.iotplaybook.com/article/3-iot-edge-storage-considerations-0

Kumar, R., & Rajasekaran, M. P. (2016). An IoT based patient monitoring system using raspberry Pi. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)* (pp. 1–4). https://doi.org/10.1109/ICCTIDE.2016.7725378

Lakshman, A., & Malik, P. (2010). Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, *44*(2), 35. https://doi.org/10.1145/1773912.1773922

Langner, C. (2015). Monitoring Raspberry Pi temperatures, *Raspberry Pi Geek*, no. 10, 16–19.

Lazarescu, M. T. (2013). Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *3*(1), 45–54. https://doi.org/10.1109/JETCAS.2013.2243032

Le, T. D., Kim, S. H., Nguyen, M. H., Kim, D., Shin, S. Y., Lee, K. E., & Righi, R. da R. (2014). EPC information services with No-SQL datastore for the Internet of Things. In *2014 IEEE International Conference on RFID (IEEE RFID)* (pp. 47–54). https://doi.org/10.1109/RFID.2014.6810711

LeHong, H., and J. Fenn Gartner (2014). Digital business technologies dominate gartner 2014 emerging technologies hype cycle, *Forbes*, available from http://www.forbes.com/sites/gartnergroup/2014/09/17/digital-business-technologies-dominate-gartner-2014-emerging-technologies-hype-cycle/ (last accessed June 2015).

Mainetti, L., Patrono, L., & Vilei, A. (2011). Evolution of wireless sensor networks towards the Internet of Things: A survey. In *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks* (pp. 1–6).

Martinez, K., Padhy, P., Elsaify, A., Zou, G., Riddoch, A., Hart, J. K., & Ong, H. L. R. (2006). Deploying a sensor network in an extreme environment. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)* (Vol. 1, p. 8 pp.-). https://doi.org/10.1109/SUTC.2006.1636175

McEwen, A., and H. Cassimally (2014). *Designing the Internet of Things*, Wiley, Chichester, United Kingdom.

McNamara, D. E. (2004). Ambient noise levels in the continental United States, *Bull. Seismol. Soc. Am.* **94,** no. 4, 1517–1527, doi:10.1785/012003001.

Merchant, H., & Ahire, D. D. (2017). Industrial Automation using IoT with Raspberry Pi.

Miller, M. (2015). *The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities are Changing the World*, QUE, Indianapolis, Indiana.

Morgan, T. P. (2018, February 19). Pushed To The Edge. Retrieved December 28, 2019, from https://www.nextplatform.com/2018/02/19/pushed-to-the-edge/

Nanometrics (n.d.). Trillium Compact Broadband Seismometer | Nanometrics, available from http://www.nanometrics.ca/seismology/products/trillium-compact (last accessed January 2016).

Neves, P., & Rodrigues, J. (2010). Internet Protocol over Wireless Sensor Networks, from Myth to Reality. *Journal of Communications*, *5*. https://doi.org/10.4304/jcm.5.3.189-196

Nordrum, A. (2016, August). Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated - IEEE Spectrum. Retrieved January 7, 2020, from https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

Obspy Development Team (n.d.a). Visualizing Probabilistic Power Spectral Densities—ObsPy Documentation (0.10.2), available from http://docs.obspy.org/tutorial/code_snippets/probabilistic_power_spectral_density.html (last accessed January 2016).

———. (n.d.b.). Obspy.signal-Signal Processing Routines for ObsPy—ObsPy Documentation (0.10.2), available from http://docs.obspy.org/packages/obspy.signal.html#module-obspy.signal (last accessed January 2016b).

Ooi, G. L., Tan, P. S., Lin, M.-L., Wang, K.-L., Zhang, Q., & Wang, Y.-H. (2016). Near real-time landslide monitoring with the smart soil particles. *Japanese Geotechnical Society Special Publication*, *2*, 1031–1034. https://doi.org/10.3208/jgssp.HKG-05

Orsini, L. (2014). How to host a website with Raspberry Pi, *Readwrite*, available from http://readwrite.com/2014/06/27/raspberry-pi-web-server-website-hosting (last accessed July 2015).

Özsu, M. T., & Valduriez, P. (2011). *Principles of distributed database systems* (3rd ed). New York: Springer Science+Business Media.

Ploetz, A., Neeraj, N., & Malepati, T. (2018). *Mastering Apache Cassandra 3.x: an expert guide to improving database scalability and availability without compromising performance*. Retrieved from http://proquest.safaribooksonline.com/9781789131499

Premsankar, G., Di Francesco, M., & Taleb, T. (2018). Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal*, *5*(2), 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263

Press, G. (2014). It’s official: The internet of things takes over big data as the most hyped technology, *Forbes*, available from http://www.forbes.com/sites/gilpress/2014/08/18/its-official-the-internet-of-things-takes-over-big-data-as-the-most-hyped-technology/ (last accessed July 2015).

Raspberry Pi Foundation (n.d.). *Raspberry Pi NOOBS Setup, Get. Started NOOBS*, available from https://www.raspberrypi.org/help/noobs-setup/ (last accessed January 2016).

Richardson, D. P. (2017). *Cloud Benchmark Testing of Cassandra on Raspberry Pi for Internet of Things Capability* (No. AFIT-ENG-MS-17-M-065). AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH WRIGHT-PATTERSON AFB United States. Retrieved from https://apps.dtic.mil/docs/citations/AD1054746

Ringler, A. T., Hagerty, M. T., Holland, J., Gonzales, A., Gee, L. S., Edwards, J. D., et al. (2015). The data quality analyzer: A quality control program for seismic data. *Computers & Geosciences*, *76*, 96–111. https://doi.org/10.1016/j.cageo.2014.12.006

Romero López, E. (2017). Creación de un cluster formado por tres Raspberry PI-3 que alojarán una base de datos NoSQL (Apache Cassandra) y motor de cálculo distribuido (Apache Spark), desplegado en contenedores (Docker). Retrieved from http://openaccess.uoc.edu/webapps/o2/handle/10609/67505

Ryza, S., Laserson, U., Owen, S., & Wills, J. (2015). *Advanced analytics with Spark* (First edition). Beijing ; Sebastopol, CA: O'Reilly.

Sanchez-Gomez, J., Gallego-Madrid, J., Sanchez-Iborra, R., Santa, J., & Skarmeta, A. F. (2020). Impact of SCHC Compression and Fragmentation in LPWAN: A Case Study with LoRaWAN. *Sensors*, *20*(1), 280. https://doi.org/10.3390/s20010280

Satyanarayanan, M. (2017). The Emergence of Edge Computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/MC.2017.9

Savazzi, S., Goratti, L., Fontanella, D., Nicoli, M., & Spagnolini, U. (2011). Pervasive UWB sensor networks for oil exploration. In *2011 IEEE International Conference on Ultra-Wideband (ICUWB)* (pp. 225–229). https://doi.org/10.1109/ICUWB.2011.6058833

Savazzi, Stefano, Rampa, V., & Spagnolini, U. (2009). High-Density Wireless Geophone Networks for Oil and Gas Monitoring and Exploration. *ERCIM News*.

Savazzi, Stefano, Goratti, L., Spagnolini, U., & Latva-aho, M. (2009). Short-range wireless sensor networks for high density seismic monitoring. Retrieved August 5, 2020, from /paper/Short-range-wireless-sensor-networks-for-high-Savazzi-Goratti/00ca7e20a8f216b9078950d66d6fa4137dd528fe

Schutt, R., and C. O'Neil (2013). *Doing Data Science*, First Ed., O'Reilly Media, Beijing, China.

Sepulveda, F., & Pulliam, J. (2016). The Internet of Geophysical Things: Raspberry Pi Enhanced REF TEK (RaPiER) System Integration and Evaluation. *Seismological Research Letters*, *87*(2A), 345–357. https://doi.org/10.1785/0220150234

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. *Future Generation Computer Systems*, *99*, 278–294. https://doi.org/10.1016/j.future.2019.04.016

Talzi, I., Hasler, A., Gruber, S., & Tschudin, C. (2007). PermaSense: Investigating permafrost with a WSN in the Swiss Alps (pp. 8–12). Presented at the Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007. https://doi.org/10.1145/1278972.1278974

Technopedia (n.d.). What is System Integration (SI)? - Definition from Techopedia, available from https://www.techopedia.com/definition/9614/system-integration-si (last accessed January 2016).

TNET (n.d.). TNET Raspberry Pi Site: SMTP Mail Setup, available from https://rpi.tnet.com/project/faqs/smtp (last accessed January 2016).

Toutain, L., Minaburo, A., Barthel, D., Zúñiga, J.-C., & Gomez, C. (2018). Static Context Header Compression (SCHC) and fragmentation for LPWAN, application to UDP/IPv6. Retrieved August 5, 2020, from https://tools.ietf.org/html/draft-ietf-lpwan-ipv6-static-context-hc-10

Traffic between the clusters | DSE 6.0 Admin guide. (n.d.). Retrieved August 3, 2020, from https://docs.datastax.com/en/dse/6.0/dse-admin/datastax_enterprise/advReplication/advRepClusterTraffic.html

Upton, E. (2020, May 28). 8GB Raspberry Pi 4 on sale now at $75. Retrieved August 26, 2020, from https://www.raspberrypi.org/blog/8gb-raspberry-pi-4-on-sale-now-at-75/

———. (2019, June). Raspberry Pi 4 on sale now from $35 - Raspberry Pi. Retrieved August 26, 2019, from https://www.raspberrypi.org/blog/raspberry-pi-4-on-sale-now-from-35/

USArray - Adopt a Station - Lower 48. (n.d.). Retrieved January 10, 2020, from http://www.usarray.org/researchers/adopt/lower48

Valero, M., Li, F., & Song, W. (2019). Smart Seismic Network for Shallow Subsurface Imaging and Infrastructure Security, 13.

Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016). Challenges and Opportunities in Edge Computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 20–26). https://doi.org/10.1109/SmartCloud.2016.18

Vasseur, J.-P., & Dunkels, A. (2010). *Interconnecting smart objects with IP: the next Internet*. Burlington, MA: Morgan Kaufmann Publishers/Elsevier.

van der Veen, J. S., van der Waaij, B., & Meijer, R. J. (2012). Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 431–438). Honolulu, HI, USA: IEEE. https://doi.org/10.1109/CLOUD.2012.18

Vincent, J. (2019, March 8). Older GPS devices are facing their own mini Y2K bug next month. Retrieved August 12, 2019, from https://www.theverge.com/2019/3/8/18255847/gps-week-rollover-issue-2019-garmin-tomtom-devices-affected

Voight, J. (2014). Which big brands are courting the maker movement, and why, *AdWeek*, available from http://www.adweek.com/news/advertising-branding/which-big-brands-are-courting-maker-movement-and-why-156315 (last accessed July 2015).

Werner-Allen, G., Johnson, J., Ruiz, M., Lees, J., & Welsh, M. (2005). Monitoring volcanic eruptions with a wireless sensor network. In *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005* (pp. 108–120). https://doi.org/10.1109/EWSN.2005.1462003

Whitmore, A., A. Agarwal, and L. D. Xu (2014). The internet of things: A survey of topics and trends, *Inf. Syst. Front.*, **17**, no. 2, 261–274, doi:10.1007/s10796-014-9489-2.

Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., et al. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture*, *98*, 289–330. https://doi.org/10.1016/j.sysarc.2019.02.009

Zorzi, M., Gluhak, A., Lange, S., & Bassi, A. (2010). From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view. *IEEE Wireless Communications*, *17*(6), 44–51. https://doi.org/10.1109/MWC.2010.5675777