NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IN-NETWORK PROCESSING ON LOW-COST IOT NODES FOR MARITIME SURVEILLANCE

by

Andrew R. Belding

March 2017

Thesis Advisor:	Gurminder Singh
Co-Advisor:	John H. Gibson

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE IN-NETWORK PROCESSING ON LOW-COST IOT NODES FOR MARITIME SURVEILLANCE	**5. FUNDING NUMBERS**
6. AUTHOR(S) Andrew R. Belding	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The effective distribution of offensive weapon capabilities to naval units at the tactical edge is a critical focus for Navy leaders. A direct byproduct of this priority is the need to employ sensor and data collection systems that can effectively guide the targeting of that offensive capability. In the recent past, wireless sensor networks have received limited use in the maritime domain due to the exploratory nature of technology, high system complexity and the high cost of system deployment. With the Internet-of-Things revolution, commercially available hardware and software components can be used to build low-cost, reliable, disposable wireless sensor networks that can leverage in-network processing schemes to greatly expand the intelligence collection footprint.

In this research, a technology demonstrator composed of low-cost wireless sensor nodes leveraging in-network processing for the gathering of wireless transmitter data was investigated. The sensor nodes were created using consumer electronic components, open-source software libraries, and networking protocols used commercially to support distributed sensors organized in a network. The network demonstrates that, for a fraction of the cost associated with conventional persistent surveillance systems, a complete sensor network can be implemented at the tactical edge and provide valuable intelligence from a variety of sources.

14. SUBJECT TERMS wireless sensor networks; internet-of-things; intelligence, surveillance and reconnaissance; in-network processing;	**15. NUMBER OF PAGES** 85
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

IN-NETWORK PROCESSING ON LOW-COST IOT NODES FOR MARITIME SURVEILLANCE

Andrew R. Belding
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2017

Approved by: Gurminder Singh
Thesis Advisor

John H. Gibson
Co-Advisor

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The effective distribution of offensive weapon capabilities to naval units at the tactical edge is a critical focus for Navy leaders. A direct byproduct of this priority is the need to employ sensor and data collection systems that can effectively guide the targeting of that offensive capability. In the recent past, wireless sensor networks have received limited use in the maritime domain due to the exploratory nature of technology, high system complexity and the high cost of system deployment. With the Internet-of-Things revolution, commercially available hardware and software components can be used to build low-cost, reliable, disposable wireless sensor networks that can leverage in-network processing schemes to greatly expand the intelligence collection footprint.

In this research, a technology demonstrator composed of low-cost wireless sensor nodes leveraging in-network processing for the gathering of wireless transmitter data was investigated. The sensor nodes were created using consumer electronic components, open-source software libraries, and networking protocols used commercially to support distributed sensors organized in a network. The network demonstrates that, for a fraction of the cost associated with conventional persistent surveillance systems, a complete sensor network can be implemented at the tactical edge and provide valuable intelligence from a variety of sources.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AP	access point
COMINT	communications intelligence
COTS	commercial-off-the-shelf
DCO	Defensive Cyber Operations
DHCP	dynamic host control protocol
EW	electronic warfare
ICS	industrial control system
IoT	Internet-of-Things
ISR	intelligence, surveillance and reconnaissance
MANET	mobile ad-hoc network
MQTT	message queuing telemetry transport
OCO	offensive cyber operations
OPE	operational preparation of the environment
QoS	quality of service
R&D	research and development
S&T	science and technology
SCADA	supervisory control and data acquisition
SIGINT	signals intelligence
STUAS	small, tactical unmanned aircraft system
TLCSN	tactical low cost sensor network
UAV	unmanned aerial vehicle
USV	unmanned surface vehicle
UUV	unmanned underwater vehicle
VPN	virtual private network
WAP	wireless access point

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

To my advisors, thank you for the countless copious hours spent reviewing and critiquing this paper.

To my amazing wife, thank you for your patience and understanding. I never would have been able to finish this thesis without you. I owe you my undying gratitude (and my sanity).

To my precocious son, thank you for many priceless moments of levity, much needed distractions, and the constant reminder of what is truly important in life.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

In today’s world, the sheer volume of data generated and transmitted across both wired and wireless networks is astounding. Mobile device usage is increasing at an exponential rate, and the number of smartphones in use worldwide is expected to exceed 6.1 billion in 2020 [1]. The estimated number of connected Internet-of-Things (IoT) devices is expected to reach 30 billion by year 2020 [2]. The ability to be constantly connected, both transmitting and receiving data, is no longer exclusive to humans—billions of wireless devices, machines, sensors, etc. are connected and rely upon information to function. The communication of information is the new lifeblood of society with everyone and every “thing” becoming networked and connected.

With this global growth of information flow, challenges facing the Navy are twofold. First, it is confronted with the challenge of collecting data from almost countless sources across many different operating environments. Second, it must be able to analyze the data collected, and then it must effectively counter or target any threats that may be discovered therein. In order to match the explosive growth of data collection and transmission, new means for first analyzing the operational environment, and then tracking and monitoring various emitters becomes critically important to the situational awareness of battlefield commanders. Wireless sensor networks provide a wide range of options to enable collection of data from areas outside the reach of conventional shipboard sensors. Through their persistent location and proximity to the target, these network nodes can provide greater granularity and better continuity of information than other remote systems, such as satellites, may be able to consistently provide.

The littoral environment, from the coastline to approximately 60 km inland, is home to more than 40% of the world’s population [3]. From the movement of bulk cargo in and out of mega-ports aboard container ships, to the movement of personnel onboard cruise ships and private vessels, the littoral environment is a key area of focus for intelligence collection. These coastal regions provide an environment well suited for the demonstrated effectiveness of maritime wireless sensor networks.

At the nexus of land and water, the amount of data being exchanged is simply mind-boggling. In these environments, the Navy is well positioned to capitalize on this data by leveraging its existent maritime presence in the littoral environment and expanding the operational picture available to commanders through sensor collection networks.

B. OBJECTIVE

This thesis researches the application of emerging commercial-off-the-shelf (COTS) IoT hardware, software and network technologies to the tactical environment, providing increased versatility and cost savings for maritime intelligence, surveillance and reconnaissance (ISR) networks. The use of in-network processing in this context is of key interest as it can provide a more efficient way of processing data in a computational, power and bandwidth constrained environment. The end result is a low-cost technology demonstrator highlighting data collection, processing and reporting techniques for use in unattended maritime ISR sensor networks.

This study examines existing capabilities by reviewing various employed maritime sensor networks and mobile ad-hoc network (MANET) solutions. Gaps are identified and focus areas for development and testing are determined. Additional steps are outlined as follows.

1. Identify and explore current maritime unattended sensor networks, both fielded and under development, to provide a frame of reference for existing capabilities.
2. Research solutions for sensor node components: processor, sensor/actuator, communication device, power solution, and memory.

3. Research open source IoT software packages and libraries of relevance in the collection and analysis of data.

4. Identify and test various in-network processing technology architectures for data collection and aggregation.

5. Develop and test a technology demonstrator and provide documentation addressing successes and shortfalls of the system.

At its completion, this study provides a technology demonstrator able to collect and process wireless and GPS signals collected by the sensor network; leverage in-network processing capabilities to enable the processing of network data without the use of a central database; and finally isolate and report relevant events to the end user.

C. THESIS STRUCTURE

The remainder of the thesis explores the feasibility of leveraging commercially available, commodity hardware components and emerging IoT software capabilities to create and test low-cost sensor networks. Chapter II provides insight into the state of the Navy policy and the ISR requirements that directly result from that policy. It also explores various fielded systems and commercially available products that can be used to address these requirements. Chapter III outlines the general components required to create a low-cost sensor network, as well as technologies and open source software libraries that are used. Chapter IV discusses the specific implementation of these components in a test network and provides documentation of three test sessions in which first individual sensor nodes are tested, and then various network configurations are deployed and evaluated. Chapter V concludes the thesis with remarks on the network usability and future work to further optimize specific capabilities as well as expand on the usefulness of the capability.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW AND BACKGROUND

> What we need to do is *distribute* the lethality of our Navy and make all of our Navy more lethal, not just surface ships but surface ships, submarines and aircraft across the broad spectrum that we operate.
>
> —Vice Admiral Thomas Rowden
> Commander, Naval Surface Forces, 2016

A. PROBLEM DOMAIN

Across the Navy, the move to decentralize forces while developing a wider distribution of critical warfighting capabilities across the fleet has become a key strategic position. The idea of “distributed lethality,” or more specifically, “the condition gained by increasing the offensive power of individual components of the surface fleet” [3], is a cornerstone of advancing the Anti Access/Area Denial (A2/AD) capabilities of the surface fleet. While defending high-value and mission-essential units will always be a key component of doctrine, increasing offensive capabilities and enabling those units at the tactical edge—in many situations those most removed from the resources of the Strike group or Amphibious ready group—has become of paramount concern [4].

1. Maritime Domain Awareness

While the concept of distributed lethality may conjure up images almost exclusively of offensive weapon systems, it is an idea that can easily be expanded to include the decentralization and distribution of intelligence collection systems. In turn, it highlights the necessity for detect-to-engage sensors and sensor networks. A commander’s intelligence picture, or in broader terms, Maritime Domain Awareness, is critical for the execution of offensive and defensive actions. At the most fundamental level, increasing the intelligence, surveillance and reconnaissance (ISR) capabilities of a platform increases its ability to effectively target, thereby increasing its offensive capabilities. As with increasing the raw offensive power of an individual ship, increasing that unit's ability to deploy and manage expansive organic sensor networks should greatly increase the overall value that unit provides to its own tactical operating environment, as well as to the theater and other operational commanders. Vice Admiral Thomas Rowden clearly highlights the criticality of persistent organic airborne ISR capabilities, conducting "dispersed operations apart from centralized command and control networks" [5], and the essential ability of a unit to achieve "localized battlespace awareness." The ability to deploy extensive unattended sensor networks, whether via airborne platforms, undersea, or on the surface, is integral to furthering over the horizon capabilities and battlespace situational awareness.

2. Historical Precedent

First in recognized value, and then in actual operational employment, use of unmanned systems in support of ISR missions has exploded over the last decade. Just as a small indicator, in 2005, 95% of the Department of Defense (DOD) aircraft inventory was manned aircraft. By 2012, that number had shrunk to 69% [6]. Tested and proven time after time in Iraq and Afghanistan, the use of unmanned systems has demonstrated how such capabilities can change the battlefield. Gradually, we are seeing these capabilities transition from expeditionary land-based forces to fleet forces, and the Navy at large. Even though unmanned aerial vehicles (UAVs) are just now gaining momentum in the surface fleet, this technology is not new. The core UAV technologies that have only recently been adopted as Small Tactical Unmanned Aerial System (STUAS) requirements have been in battlefield use in Iraq and Afghanistan since the early 2000s. Over years of expeditionary and ad-hoc maritime use, the Scan Eagle UAV paved the way for the larger RQ-21 Blackjack to finally break onto the operational scene as a formalized Navy and Marine Corps program of record [7]. The timeline, sometimes as long as 10 to 12 years, required by many current DOD acquisition mechanisms to formally acquire an approved solution to an operational requirement is evident [8]. Even though the technology may exist in various other channels, whether commercial industry, other military branches, or other government agencies, formal programmatic adoption time significantly lags that of production within many commercial industries. A similar trend can be seen in the explosion of multi-rotor drone use in the industry and commercial sectors, with only limited adoption by the military.

3. Commercial Industry

Emerging technologies in the consumer electronics sector follow very similar trends in how they are conceptualized, developed, marketed and produced. The Gartner Inc.'s Hype Cycle is a very interesting way of tracking and labeling industry trends and the level of hype associated with them. The Cycle suggests that it takes roughly five to 10 years for a new technology to develop from an innovation trigger to the "plateau of productivity," when the technology has transitioned from an over-hyped trinket that everyone discusses, to a ubiquitous part of society and the marketplace—it has gone from gimmick to commercialized technology [9].

The Internet-of-Things is a buzzword whose use spans industries from manufacturing to transportation to retail to information technology. Its meaning can be summed up in a few simple sentences. IoT is the networking together of "things"—sensors, actuators, devices, tools, machines, etc.—vice just people. The key focus is on the collection and processing of information, with the intent to bring about some result. The scope of adoption and the scale of growth of IoT is expected to dwarf that of smartphones and associated mobile devices with an estimated 30 billion IoT devices in use by 2020 [2].

Mobile device usage has exploded. Estimates put the number of smartphones and cell phones in use globally at approximately 4.6 billion in 2016 [10]. However, experts anticipate a typical household could contain hundreds of IoT devices—orders of magnitude greater than the two to five smartphones per household. It would logically follow that technologies associated with the development of these networks and sensors could translate to the military field where data and intelligence collection is of paramount importance. The ability to decentralize system control and distribute sensors and collection nodes at a cloud scale while keeping cost and power requirements low would revolutionize ISR networks.

Given previously discussed trends, a rough estimate could put the timeline at least 10 years beyond the commercialization of these technologies before we see any sort of formal adoption by the Navy. If, according to the Gartner group, IoT is just reaching the peak of hype in the commercial world, a five- to 10-year timeline until the plateau of productivity is reached is not unreasonable. Furthermore, given the latency seen by the military's conservative mentality before the adoption of new commercial technologies in the past, one could easily add an additional three to five years onto the timeline [8]. Based on these numbers, 10 years could even be an aggressive estimate.

Given the current sensor network requirements by science and technology (S&T) organizations across the Navy [11], the strategic doctrine provided by senior leadership, and the operational challenges the Navy is facing today, there are far more benefits to be reaped from the IoT revolution than are currently being leveraged. Further research into the development of unattended maritime sensor networks, based on IoT technologies, stands to reap some significant rewards, as described below.

B. TACTICAL USES FOR UNATTENDED MARITIME SENSOR NETWORKS

Unattended sensor networks have the ability to provide a wide range of benefits to maritime forces. Deployment of these networks can provide sustained, low visibility collection in both permissive and non-permissive environments. Whether fixed and in place for an extended period of time, or dynamic and only intended for single mission use, the ability to collect and aggregate data from multiple vantage points can provide a significant tactical advantage. Their use supports a wide range of functions; a few critical ones are covered in the following subsections, but there are applications in many other domains.

1. Intelligence, Surveillance and Reconnaissance

The use of both persistent and expendable unattended sensors for intelligence, surveillance and reconnaissance is not a new idea. By distributing sensor networks across the battle space, overall intelligence collection can be significantly increased. Since the Vietnam War, wireless sensor networks have played important roles in both combat and non-combat arenas [12]. Access and placement in various locations can provide visibility of previously unseen or un-exploitable signals. The same sensors could also be used to provide deceptive capabilities, transmitting signals simulating traffic, combat systems, or other capabilities to confuse or otherwise disrupt adversary operations. This concept applies to virtually every intelligence discipline from targeting communication networks collecting Communications Intelligence (COMINT) to deploying acoustic sensors to target submarines and other maritime vessels. The value in having a remote sensor with access to a denied environment can be crucial for providing battlefield commanders with critical intelligence facilitating everything from more timely and effective tactical decisions to more educated and comprehensive strategic plans. In the maritime environment, there are many situations where overhead assets are simply too expensive or unavailable, and having a full-sized maritime platform with the appropriate sensor suite at that location is simply impossible. The use of unattended sensors helps mitigate these collection gaps.

2. Cyber Effects

Sensor networks can also be used to provide cyber effects, whether in support of Offensive Cyber Operations (OCO), Defensive Cyber Operations (DCO) or Cyber Operational Preparation of the Environment (OPE). In an offensive operation, sensor nodes can be used as mobile access points providing semi-persistent launch points for implants or root kits that might otherwise not have access to an off-net target. In support of Cyber OPE, they can be used to collect, characterize or trigger certain effects based on the environment or detection of specific actions. From a DCO standpoint, a multitude of deceptive operations can be used, from creating attractive tar-pits to lure, trap, and analyze an adversary's offensive capability, to simply monitoring the environment to assist in identifying blue force vulnerabilities, or detecting various attack signatures.

3. Electronic Warfare

Sensors networks can also provide an extension to fleet electronic warfare capabilities. Whether conducting electronic attack and providing a remote platform for jamming an adversary's signal, collecting electronic intelligence transmissions in order to identify or characterize a specific unit for electronic support, or acting as remote perimeter defense to conduct electronic protection, sensor networks can extend the range of maritime domain awareness.

4. Meteorological Study

One of the most common uses for unattended maritime sensors has been the monitoring of meteorological phenomena. Buoy-mounted sensors are used for tracking water temperature, salinity, swell height, and many other attributes. These buoys play roles in tsunami early warning systems, hurricane tracking networks, and other environmental studies.

C. CURRENT SYSTEMS AND CAPABILITIES

Sensor and sensor networks have been a focus of military R&D for many years. In the tactical ground environment, there are multiple options for fielding and managing tactical networks. For years, the primary focus of these networks has been on voice communications between ships, vehicles, and personnel. Over the last 10 years, development priorities have gradually evolved to include data, as industry and government have begun to realize that not just people need to be connected in a communication network, but so do various sensors and devices. As this trend has progressed, the available form factor of many of these radios has evolved from handheld tactical radios to include much smaller, more power-efficient embedded devices better suited for sensor node integration.

When reviewing current systems for wireless sensor networks, the systems can be split into two main categories from the standpoint of (1) systems or technologies that enable the wireless network creation and (2) actual sensor platforms with full network and sensor integration. The following subsections address three of the more widely used tactical MANET systems, and five specific sensor platforms in use or development today. These lists are in no way all-inclusive but provide a good snapshot of today's tactical and maritime wireless sensor network capabilities.

1. Tactical Networking Technologies

These systems all focus on establishing a seamless wireless network. However, from different vendors, each provides a similar capability with various proprietary implementations and signal processing schemes. Key in the tactical environment is the ability to build MANETs: the ability to communicate without any required central infrastructure, as well as adapt to nodes coming in and going from the network. Each one of these radios provides a MANET capability.

a. Persistent Systems

The Wave Relay line of radios is used across multiple Navy and Marine Corps acquisition programs of record. The current line of systems includes the MPU3, MPU4 and MPU5 (5100-series) and Gen 4 integration board. All are targeted at connecting personnel in an operational environment over both voice and data communications. Tactical use has ranged from basic intra-squad communications to providing meshed command and control communication networks for UAVs.

The integration board, shown in Figure 1, is targeted at embedded systems from UAVs to sensor networks. It has a much smaller form-factor than their tactical radios and is optimized for a variety of power sources and housings. It weighs roughly 3-4 ounces, provides full IP connectivity, can support a bandwidth of up to 31 Mbps, and is fielded using frequencies from 900 MHz to 5 GHz at up to two watts of power [13].

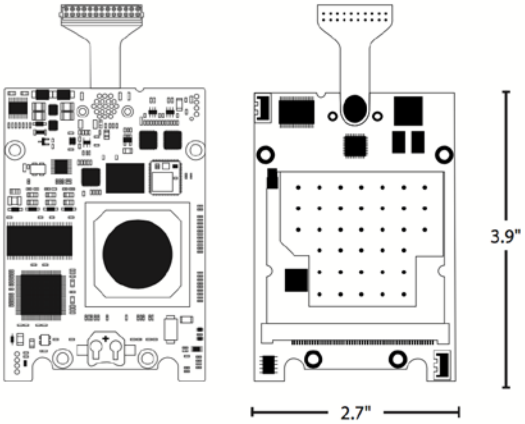


Figure 1. Persistent Systems' Gen 4 Integration Unit. Source: [13].

b. Trellisware

Trellisware, another staple in government inventory, also provides its own line of networking products for creating MANETs in the tactical environment. The TW-600 Ocelot is designed for embedded device use. Like the Gen 4, the TW-600 is available across various radio frequency bands. It advertises a 26-mile, line-of-sight single hop range, with up to eight Mbps throughput. It also weighs approximately three oz. and has a maximum transmit power of approximately two watts [14].

Figure 2. Trellisware TW-600 Ocelot Embedded Tactical MANET Module. Source: [14].

c. Silvus Technologies

Silvus Technologies markets their Streamcaster series of radios for tactical MANET creation. The SC 3822 is the company's embedded module marketed for use in unmanned and unattended systems (Figure 3). The SC 3822 leverages a dual-band MIMO technology and provides the widest range of frequency bands of the three discussed here, ranging from 400 MHz to 5 GHz. Advertising a maximum throughput of up to 70 Mbps, the SC3822 is delivered in a slightly more robust package weighing approximately one pound [15].

Figure 3. Silvus Technologies Streamcaster 3822 MIMO Module. Source: [15].

2. Integrated Sensor Platforms

The following systems cover a wide range of sensor platforms. From highly mobile to single use and expendable, these sensors illustrate the breadth of devices currently in use for maritime purposes.

a. Liquid Robotics

Liquid Robotics produces the SV3 Wave Glider unmanned surface vehicle (USV). At approximately 9.5 feet long and roughly 230 pounds, the SV3 is a long endurance, mobile monitoring platform (Figure 4). It is able to spend over a year at sea and leverages both solar power and wave motion for propulsion and payload power. More than a simple radio, the Wave Glider has been used as a fully networked data collection node able to be completely self-powered and self-sustaining for months at a time. Typically incorporating satellite (iridium) connectivity, it can be used as a standalone sensor or as a gateway for other sensors. Utilizing its ability to provide data connectivity for local sensor networks, the Wave Glider has been used in testing scenarios in the SOUTHCOM AO [16]. With a speed of approximately two knots, it is well suited for loitering and data collection within a specified maritime region.

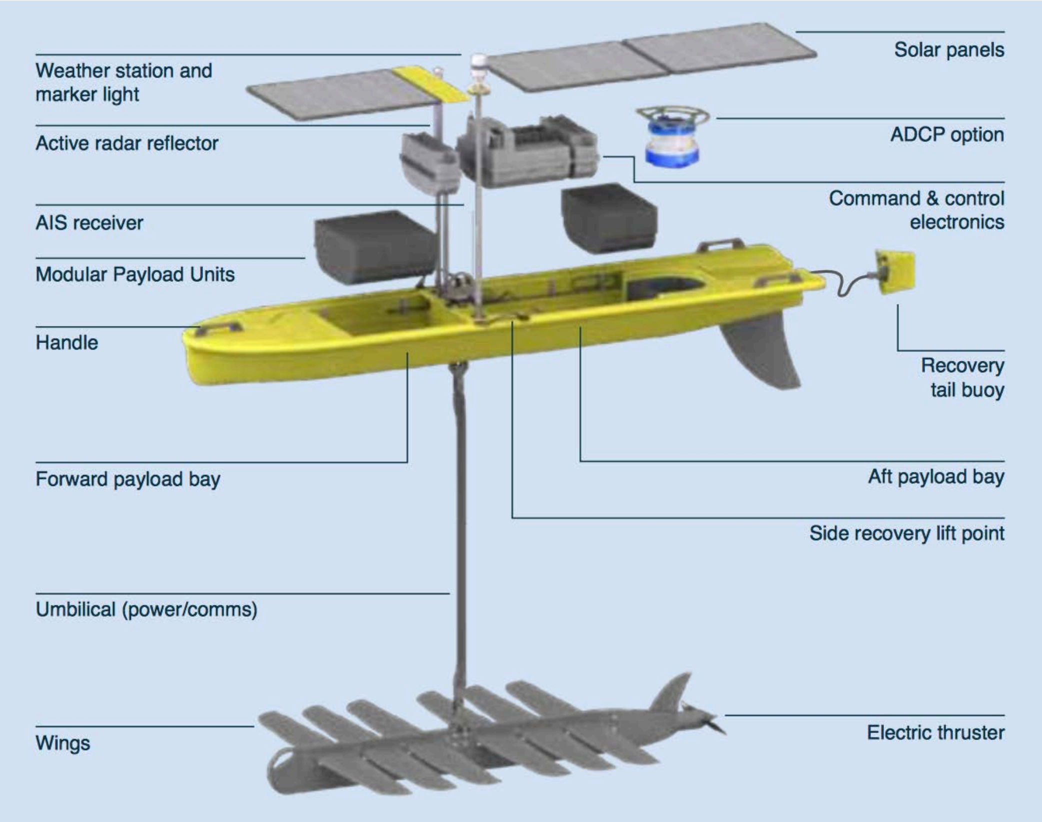


Figure 4. Liquid Robotics SV3 Wave Glider. Source: [17].

b. SEAWEB

SEAWEB is a static collection network that has been developed and tested by NPS for the U.S. Navy. It is composed of persistent nodes, anchored to the ocean floor and floating at various depths for the purpose of collecting and identifying underwater acoustic signals [18]. Once a node is triggered, it uses through-water communications to a floating gateway that allows for exfiltration of collected data [18] (Figure 5). SEAWEB sensors comprised the data collection portion of the SOUTHCOM experiment, referred to above, with the SV3 acting as the gateway.

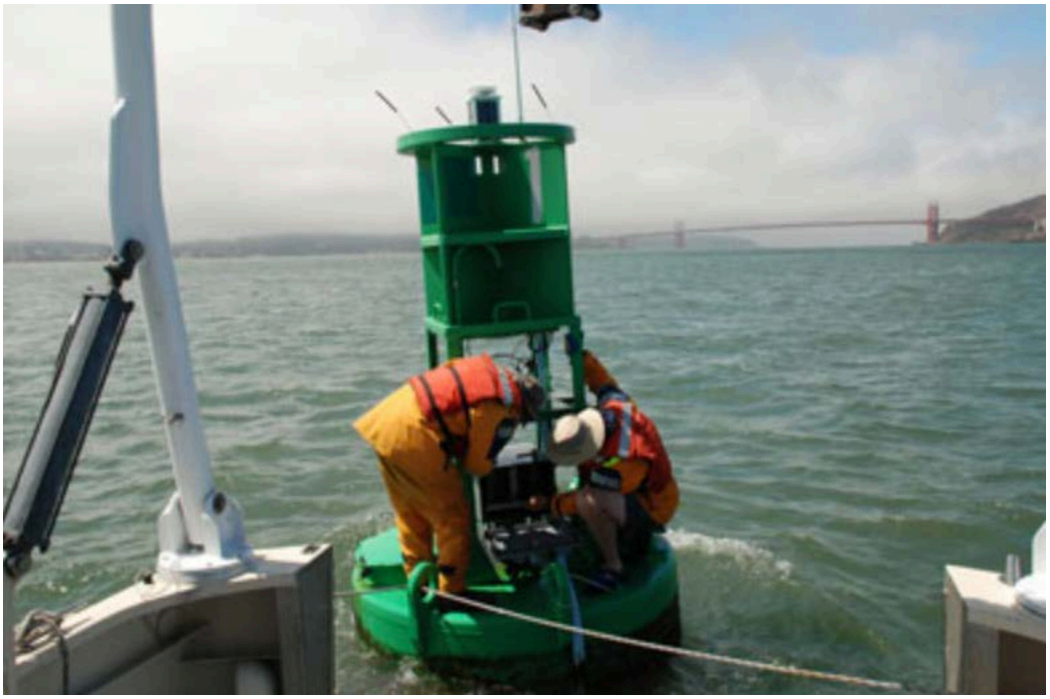


Figure 5. SEAWEB Gateway Installation on Coast Guard Buoy in San Francisco Bay. Source: [18].

c. *Sonobuoys*

The most commonly used expendable sensor in the Navy inventory is the sonobuoy. The Navy employs sonobuoys primarily in the conduct of airborne anti-submarine warfare (ASW). Most commonly used are the AN/SSQ-53G DIFAR (Passive) and the AN/SSQ-62F DICASS (Active) models (Figure 6). Using both passive and active acoustic measures, sonobuoys are effective ASW devices, with hydrophones deploying to reprogrammable depths for optimal readings. These sensors weigh roughly 20–40 pounds and have a deployment life of up to 8 hours. Transmissions are via VHF with an advertised range of up 35 miles to an aircraft flying at roughly 5000ft AGL. While effective for short term tracking and collection of signals, the fact that sonobuoys are static with no ability for repositioning complicates both the logistics and cost of their deployment. Once they reach end of life, they sink in the ocean [19].

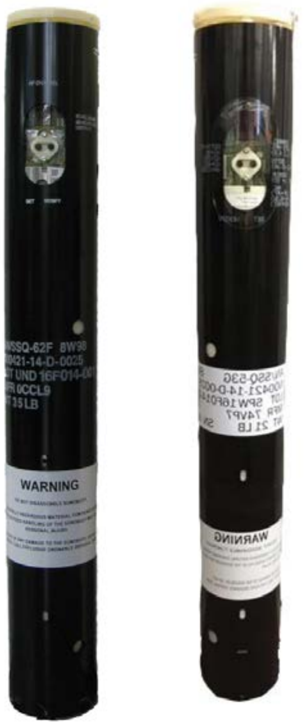
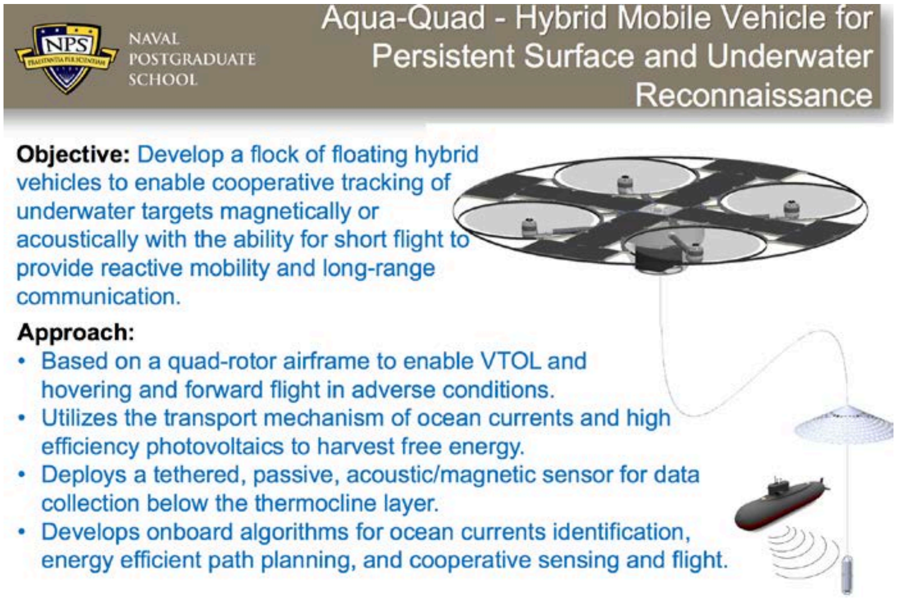


Figure 6. AN/SSQ-62F (Left) Active Sonobuoy and the AN/SSQ-53G (Right). Source: [19].

d. Multi-Rotor Sonobuoys

The explosive growth in fielding of multirotor systems over the last three to four years has provided a unique opportunity for sensor delivery. With lift capacities ranging from ounces to hundreds of pounds, and costs from tens of dollars to hundreds of thousands of dollars, there are a multitude of potential use cases. Recent work by NPS faculty and students has resulted in the award of a patent for a mobile multi-rotor sonobuoy used for undersea and surface contact detection [20] (Figure 7). The ability to reposition an unattended sensor repeatedly in support of either a moving target or changes in target priorities can be invaluable to a commander. Additionally, the ability to recover the sensor, or at a minimum the delivery vehicle, provides considerable cost saving potential. Dubbed the “Aqua Quad,” it has been designed with underwater acoustic sensors as the focus, however the payload versatility and the use of solar to maintain power could easily lend itself to various other sensor nodes. In addition to sensor mobility, the collaborative nature of these platforms provides significant benefit, increasing the effectiveness of sensor networks.



NPS
NAVAL
POSTGRADUATE
SCHOOL

Aqua-Quad - Hybrid Mobile Vehicle for Persistent Surface and Underwater Reconnaissance

Objective: Develop a flock of floating hybrid vehicles to enable cooperative tracking of underwater targets magnetically or acoustically with the ability for short flight to provide reactive mobility and long-range communication.

Approach:

- Based on a quad-rotor airframe to enable VTOL and hovering and forward flight in adverse conditions.
- Utilizes the transport mechanism of ocean currents and high efficiency photovoltaics to harvest free energy.
- Deploys a tethered, passive, acoustic/magnetic sensor for data collection below the thermocline layer.
- Develops onboard algorithms for ocean currents identification, energy efficient path planning, and cooperative sensing and flight.

Figure 7. Aqua-Quad Overview. Source: [20].

D. SENSOR NETWORK USE CASES

Understanding what wireless devices and networks exist in an operating environment can be a critical intelligence requirement. While this intelligence gain is obvious in a dense urban environment, it can be equally important in understanding maritime littoral environments. With the growth of wireless devices and sensor in port facilities, harbors and marinas, and onboard all types of vessels, wireless networks are no longer strictly land based technologies used in urban environments. Wireless networks exist on commercial and military vessels and are used from everything to recreational internet access to monitoring and controlling ships systems. Being able to identify clients, infrastructure devices, and actual networks can provide critical intelligence and insights to a commander.

Immediate proximity can be a key component to successful collection and characterization of many signals. The littoral environment can be particularly difficult within which to work due to the restrictions place on operating forces with the requirement to stay in international waters. At this range, onboard systems can be seriously limited in the fidelity of information that can be collected. The ability to remotely position, and reposition a cloud of unattended sensors in the immediate vicinity of a target would greatly enable the type of ISR collection that could be conducted.

1. Tracking of Movement

A Navy frigate is tasked with a counter-narcotics mission in the Caribbean. A very porous coastline has been contributing to a significant increase in drug running operations to and from various islands as well as the mainland. Maritime traffic is incredibly high, and identifying drug boats is nearly impossible. Forces must begin monitoring the region, but are unsure how to correlate various boats to known drug routes and known individuals. By strategically placing sensor networks around harbors and known maritime routes, data correlation will be able to provide a list of emitters seen across multiple locations.

2. Characterization for Follow-on Operations

After building a baseline of the wireless environment, establishing what the key components of the local infrastructure are, and what networks and clients exist, the groundwork is laid for a variety of operations, from leveraging cyber effects to further intelligence gathering.

E. SUMMARY

This chapter has discussed the strategic direction the surface Navy is taking with distributed lethality, and how this movement directly affects the need for greater ability to decentralize and distribute mechanisms for collecting intelligence. Wireless sensor networks have long had resounding impact on the battlefield, but only in recent years, with the explosion of IoT capabilities and devices, has the commercial off-the-shelf technology been so inexpensive with such a wide range of capabilities. There is necessity across the Navy for inexpensive, rapidly-deployable wireless sensor networks and industry innovation with IoT devices has provided a wealth of capability able to be leveraged for expanded maritime collection. In Chapter III, the concepts for developing and employing a tactical low cost sensor network are discussed. Key focus is placed on cost, network efficiencies, and in-network processing.

III. SYSTEM DEVELOPMENT AND EXPLORATION

This chapter addresses the specifics surrounding the hardware components, software development and overall network structure of the tactical low-cost sensor network, or TLCSN. The intent of TLCSN is to provide a low cost ISR capability that leverages in-network processing techniques to conduct the identification, categorization and tracking of wireless clients and access points of interest. The chapter begins with a high-level illustration of the system architecture, followed by a discussion of the general system data flow. Next is a deep dive into the specific hardware selected for each system component, and it concludes with the software, data structures, libraries and network architectures that are used to complete an end-to-end sensor network.

The selection of components for this technology demonstrator is based on commercial availability and cost. One objective of this project is to minimize cost; explore whether capable sensor networks can be easily created at a fraction of the expense of many current systems, and field in volumes that could provide increased fidelity of wireless emitter across the operating environment. In total, less than $1,500 was spent on the hardware used to create 11 main components—10 collection nodes with power, GPS and 802.11 networking capability, and one central server.

It is understood that most commercial products do not have the required military specification ratings required by acquisition programs, nor are they likely to be completely waterproofed for long-term operation in a maritime environment. This problem is no small feat to overcome given the current requirements levied on these programs. However, the focus for this demonstration is not a production ready system, but rather the utilization of widely available computing and networking capabilities to collect, process, and disseminate key pieces of data in support of littoral collection operations.

A. SYSTEM ARCHITECTURE

The demonstrated application of the TLCSN is the collection, processing, and dissemination of 802.11 messages—beacons, probe requests and data packets, to name a few. This data provides enhanced situational awareness identifying and displaying what clients and APs may exist in a specific operating area, as well as add context by analyzing and characterizing any trends and patterns of movement that may appear. The overall system assists in characterizing the wireless environment by aggregating the data into a streamlined format for display. In colloquial terms, the data set is comparable to that provided by "war-driving"; however, this data serves as an initial demonstration of the sensor node and network capabilities. It provides enough volume over time to serve as an interesting data set for analysis while also providing useful data to a commander. Various other sensors can be added for different intelligence gain and used either interchangeably or in concert with each other. The key here is the collaborative nature of the sensor nodes in providing a more effective operational picture.

There are four main components of the TLCSN system: the sensor nodes, the server/broker platform, the system controller, and the overall network structure. Each is described below.

1. Sensor Nodes

The sensor nodes conduct the actual data collection and reporting. They would be spread across a specified target area as dictated by the operational or testing requirements. When deployed, they would be automatically activated and establish connectivity with the network at large. Through this network connection, each node will be able to transmit and receive collected data across the network, reach back to a designated repository, and receive additional or modified tasking by a system controller. Nodes are assumed to be more or less static during their collection mode in this environment. Whether mounted to an existing buoy or part of a separate delivery mechanism, persistent collection location will provide better data characterization. Minimal movement, accounting for drift and currents, is acceptable.

2. Server / Broker

The broker consolidates, processes and distributes data across the network as required. It has a complex role and provides various capabilities from acting as a VPN server allowing connectivity from outside the network, to serving as the broker for collected data. In some situations, it may also act as a WAP and DHCP server for the local network.

3. System Controller

The system controller serves as the user interface for the system. It provides mechanisms for accessing, viewing and analyzing the data, while providing additional tasking and control to various sensors. For example, if the controller wishes to use the sensor node for additional activities beyond the passive collection of data, that capability is available. To provide the maximum flexibility for this system, it is assumed that the controller resides at some external location. In all cases, it connects to the sensor network via a VPN connection and is never physically present on the same network. This assumption is in keeping with many operational situations where the controller will not be collocated with the sensor net and a remote connection to access data and control sensors will be required.

4. Network

There are three general network architectural constructs that are explored. For the sake of simplicity, each one is illustrated using only five nodes. Actual implementation can be scaled to a much larger extent—both in the number of nodes working with a server (covering more area within a general region), as well as the total number of servers (covering more regions). This would serve to provide both greater breadth as well as depth of coverage. Hybrid constructs of these two architectures are further addressed in the network development section.

a. Centralized Architecture

The first architecture is a centralized network with all collection nodes and servers operating on the same local area network (Figure 8). This is best illustrated by envisioning a typical home wireless network. In this case, each client on the network is connected to the same wireless AP. The AP then typically serves as a router and gateway to the Internet, providing bidirectional access to the home network through a single public IP. There are various pros and cons with this approach. To begin with, the network is incredibly simple to create. Once the AP is set up, nodes can immediately connect to it, gaining access to that network and any additional ones to where routing may exist. Nodes can dynamically connect and disconnect from the network and connectivity for other nodes is not affected. That being said, having a central AP creates a single point of failure, if the AP is disabled all network connectivity is lost until it can be replaced. This makes for a network with very poor robustness unless that AP is either extremely reliable or part of a highly redundant system. As with a home network, this type of network can have significant range limitations contingent upon which physical access technology is used. For example, with commercially fielded 802.11 APs, the network connection is typically only effective to a few hundred feet. The point-to-point nature of this network is thereby limited to that maximum range. The use of higher gain antennae and power amplifiers can greatly extend the overall range of each node, but it will always be a two-hop trip between nodes (node1-AP then AP-node2).

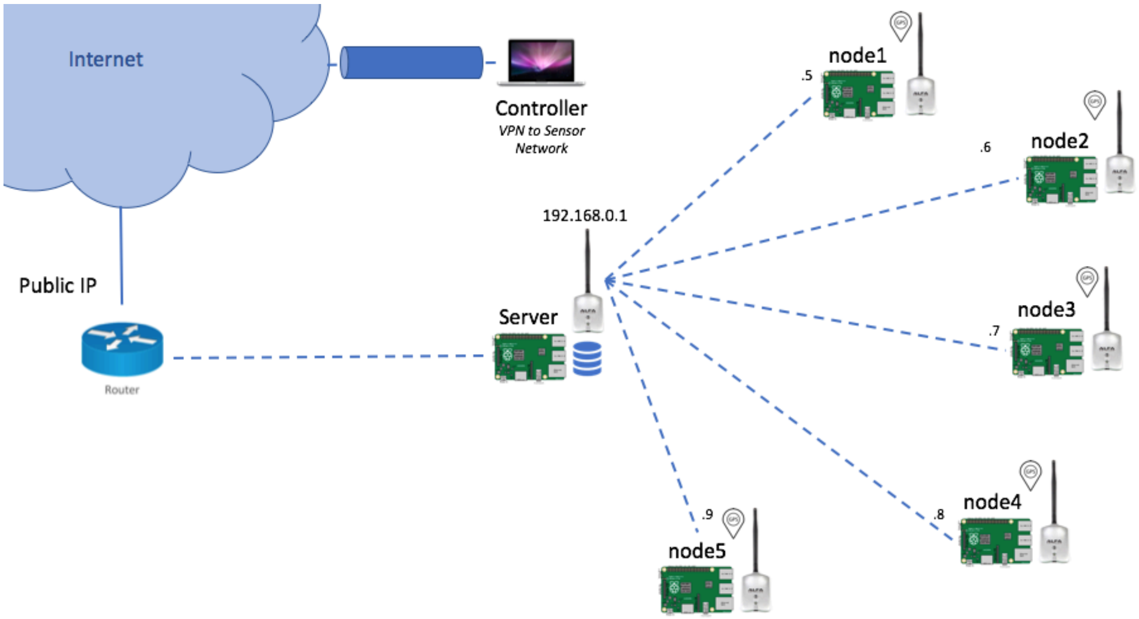


Figure 8. Centralized Sensor Network

b. Ad-Hoc Architecture

The second type is an ad-hoc architecture (Figure 9). Like the centralized approach, all the nodes are directly connected to the same local area network. However, with an ad-hoc network there is no central AP providing connectivity to the network. The central server still exists and is providing access to outside networks but no longer provides the central AP, it is just another member of the network. Each node connects to other nodes within range of its transceiver and an infrastructure-less network is formed. As long as at least one node has connectivity to the main server, sensors can be daisy chained one after the other to greatly extend the overall coverage area. In this way, sensors connected by a lower power and shorter range technology can still cover large geographic areas. This also provides much greater network robustness and resilience. If the gateway server was to crash in this illustration, all the other sensor nodes would still have connectivity across the entire network and would be able to exchange data.

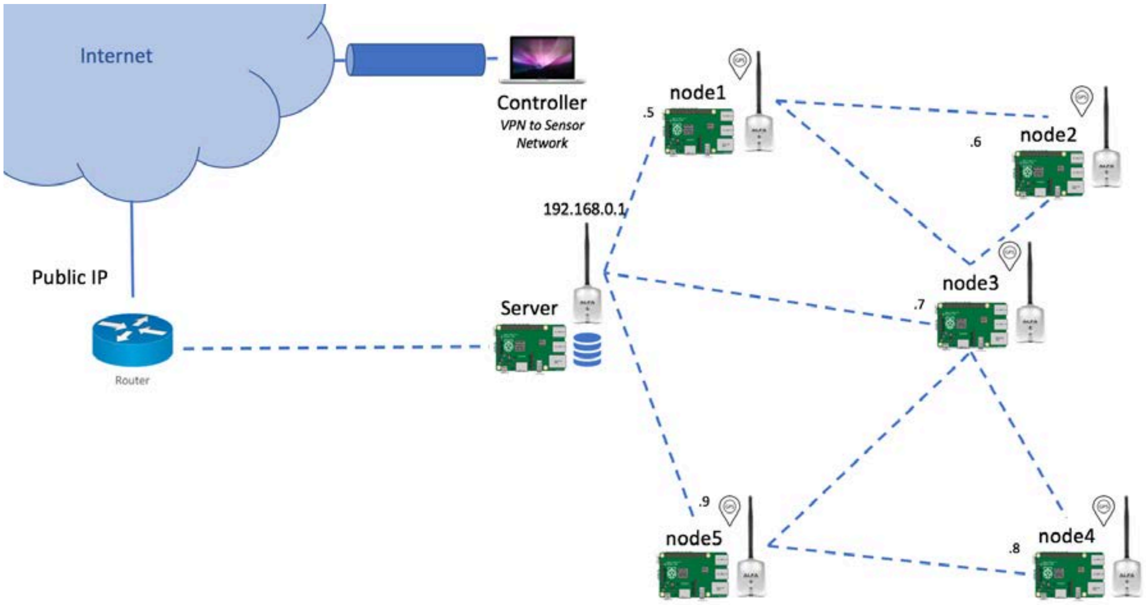


Figure 9. Ad-Hoc Sensor Network

c. *Distributed Architecture*

The third architecture is a distributed one (Figure 10). In this case, each node establishes a VPN tunnel back to the home network for connectivity. As long as a node is able to connect to the Internet, it can securely reach back to the home network and establish a connection with the central broker and controller. This type of connectivity provides far greater flexibility and range, with the ability to have nodes in different regions of not just a city or town, but of the globe. This not only provides a greater physical extension of the network, but also establishes an encrypted tunnel for the data as it passes back to the home network. However, with greater range and flexibility comes far greater latency and potential for delays across the network. No longer is the data travelling over the air via a single point-to-point wireless link, it could be traversing significant portions of the globe, tens or even hundreds of other routers and autonomous systems. Network overhead is also increased with VPN connectivity.

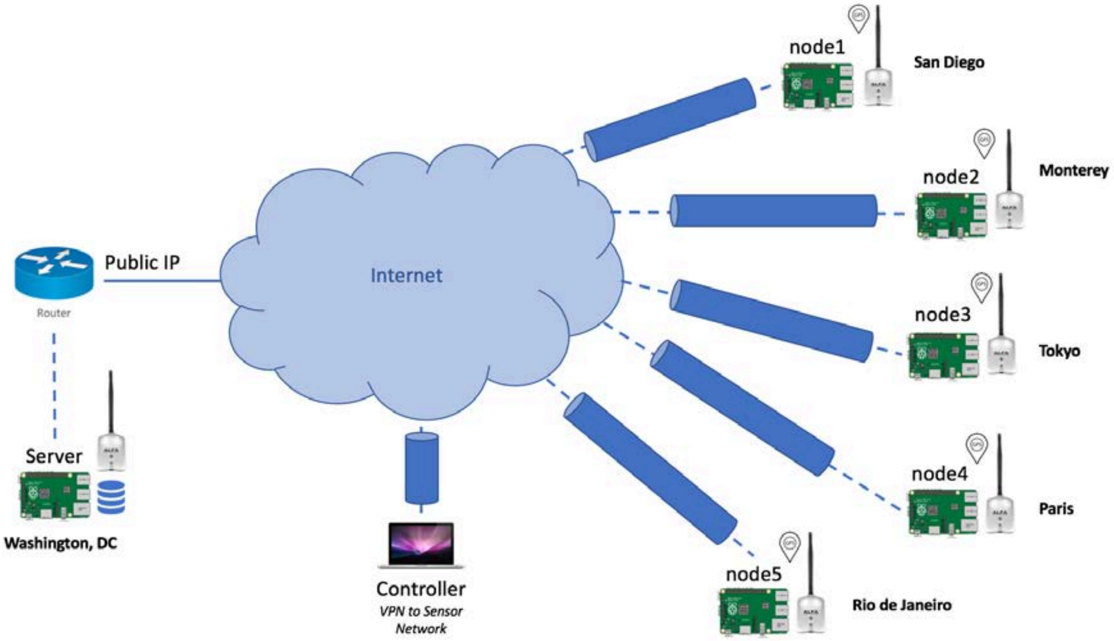


Figure 10. Distributed Sensor Network

B. DATA FLOW AND MQTT

Operating an unattended wireless sensor network in a littoral environment faces some potential limitations with regard to data flow. In general, both bandwidth and power are significantly constrained. As a result, the flow of data across the system is kept as simple as possible while still achieving the desired end-state. With this goal in mind, MQ Telemetry Transport (MQTT) is used for all machine-to-machine communication across the network. MQTT is simply described as “a M2M/IoT connectivity protocol. [MQTT] was designed as an extremely lightweight publish/subscribe message transport. It is useful for connection with remote locations where a small code footprint is required and/or network bandwidth is at a premium” [21]. Built on top of the TCP/IP stack, MQTT is used for lightweight messaging and data transfer services all over the world to include highly commercialized applications like Facebook Messenger [22]. First developed in 1999, the growth of the IoT in recent years has brought the protocol to the forefront. Its design to run on embedded, power and bandwidth-constrained systems, makes it an ideal candidate for wireless sensor networks.

MQTT is based on a publish/subscribe messaging paradigm built around “topics.” For basic functionality, a client first connects to the MQTT broker. Once connected to the broker, a client can publish messages to a topic or simply subscribe to a topic. Any client connected to that broker will receive any message published to a topic to which it is subscribed. For example, Client 1 is publishing a list of its favorite concerts to the topic “events.” Client 2 is a concert fanatic and has already subscribed to the topic “events.” Client 2 will receive all messages that Client 1 publishes to the topic “events” (Figure 11).

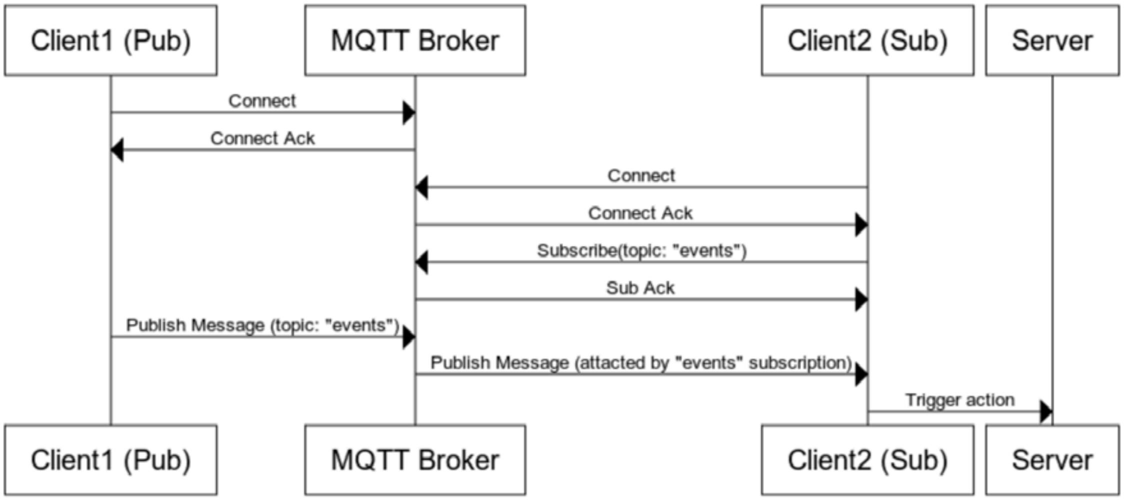


Figure 11. MQTT Publish/Subscribe Sequence Diagram. Source: [23].

The general flow for the system is as follows. First, the node powers up and establishes network connectivity, either directly to a predetermined AP or via VPN through an alternate source. The node then connects to the MQTT broker, authenticating itself via user name and password. It begins polling GPS data as well as passively collecting beacons and probe requests through a separate wireless interface. Based on the periodicity provided by the user, reports containing GPS position and collected 802.11 data are published through an MQTT topic that pertains to that specific node. These topics are received by the MQTT Broker and any other entities that are subscribed. At the same time, each node will also be subscribing to various other topics published by surrounding nodes. The simplicity of MQTT is used to facilitate in-network processing of data providing greater awareness at the sensor node level and more rapid analysis within the network.

The server hosting the MQTT Broker also serves as backup database server for all collected data, subscribing to all topics and receiving the published messages. Upon receiving a published message, it parses all the data, populating the database with requisite information—times, locations, MAC addresses, RSSI values, SSIDs, etc. In order to consolidate systems, the platform also serves as the VPN server for the sensor network.

The system controller remains relatively separate from the sensor network - its purpose being mainly to simply display the data that the network has processed and reported. It also connects to the database in order to conduct additional queries and have access to all collected data. Connectivity is provided via client software associated with that database. For purposes of specific sensor control, it will publish messages to topics associated with each specific sensor.

C. SENSOR NODE DEVELOPMENT

Each sensor node passively collects and stores 802.11 AP beacons, 802.11 client probe requests, 802.11 data packets and all associated data (MAC addresses, SSIDs, etc.) that are transmitted by wireless systems in the vicinity of the node. It also collects and records GPS data to provide locational information in conjunction with the 802.11 messages. This information is periodically compiled in a report and published over the network to the MQTT broker. At a minimum, the node has a username and password protected connection with the MQTT broker and is on a secure network with commercial grade encryption. The node also subscribes to topics published by other nodes in the zone. Through this collaborative, in-network processing (explained in Section F.1), a greater picture of the operational area emerges, identifying static and dynamic nodes, tracking movement, and other characteristics.

1. Components

The main sensor used for the collection of wireless data is the Alfa AWUSO36NH 802.11b/g/n USB network interface card. Costing roughly $30 on the commercial market, the Alfa card is capable of being run in monitor mode for the passive collection of wireless traffic while not associating with any network. It is also capable of packet injection to support various active network operations. This card was chosen because of both cost and its use of interchangeable antennas. For general purposes, a 5 dBi 2.4 GHz antenna is used for all experimentation operations. The wireless data collected is decoded using the *scapy* Python library [24].

An external USB GPS puck (US GlobalSat BU-353S4) is used for the collection of location data. Also available for $30 on the commercial market, it has a magnetic mount, is waterproof, and has a high tracking sensitivity of -163 dBm [25]. The Python libraries *gpsd* and *gpsd-clients* are used to parse the data to include time, latitude, longitude, and elevation [26]. This data is used to tag each report sent from the node, as well as for logging data internally.

All processing is done using commercially available Raspberry Pi 3 Model B single board computers. With a 1.2 GHz 64-bit quad-core ARMv8 processor, 1GB of RAM, a 16GB SD card, built-in 802.11n wireless interface, and Bluetooth 4.1, the Raspberry Pi 3+ provides immense capability in a very small package for approximately $40 [27]. The Raspberry Pi 3 also has four USB ports, an HDMI port, 40 GPIO pins, and one Ethernet port for straightforward and simple expansion of capabilities and sensors. The strength in the Raspberry Pi as a processing platform is truly the ease by which different sensors can be integrated or removed. While in this study the two main sensors are GPS and 802.11 collection, these are not the limits of either the Raspberry Pi or the sensors themselves; it is merely because these are sufficient for the demonstrations of concept we are developing. Different sensors and data sets can be used with the same overall sensor node structure and system architecture.

All programming for the Raspberry Pi is done in Python using commercially available libraries and other open source software. As noted earlier, communication between the nodes and other servers is over the transport/application level protocol MQTT. Developed by Eclipse, Mosquitto and its associated client software provide an open source method for developing and implementing network communication over MQTT [28]. For our demonstration, each node runs the Mosquitto Client enabling it to publish and subscribe to topics across the network. *Paho-MQTT* is the Python library used to interface with the Mosquitto client software [29].

To enable VPN connectivity for each node, the OpenVPN client is also installed [30]. Individual configuration files are pre-loaded on each node to ensure unique secure connections. The VPN server manages certificates, which are revoked and issued as needed.

When operating in a standalone mode, power is provided by the Anker Powercore 10000. This 10,000 mAh battery provides up to a 2.4 Amp current, significantly more than the combined total required by the Raspberry Pi, the Alfa card and the GPS [31]. At $30 and roughly the size and weight of a deck of cards, the Anker is a viable, inexpensive solution for powering the individual nodes. While operational use will most likely require a greater battery life than five to six hours, this is more than long enough duration for a technology demonstrator. The entire node is shown in Figure 12.

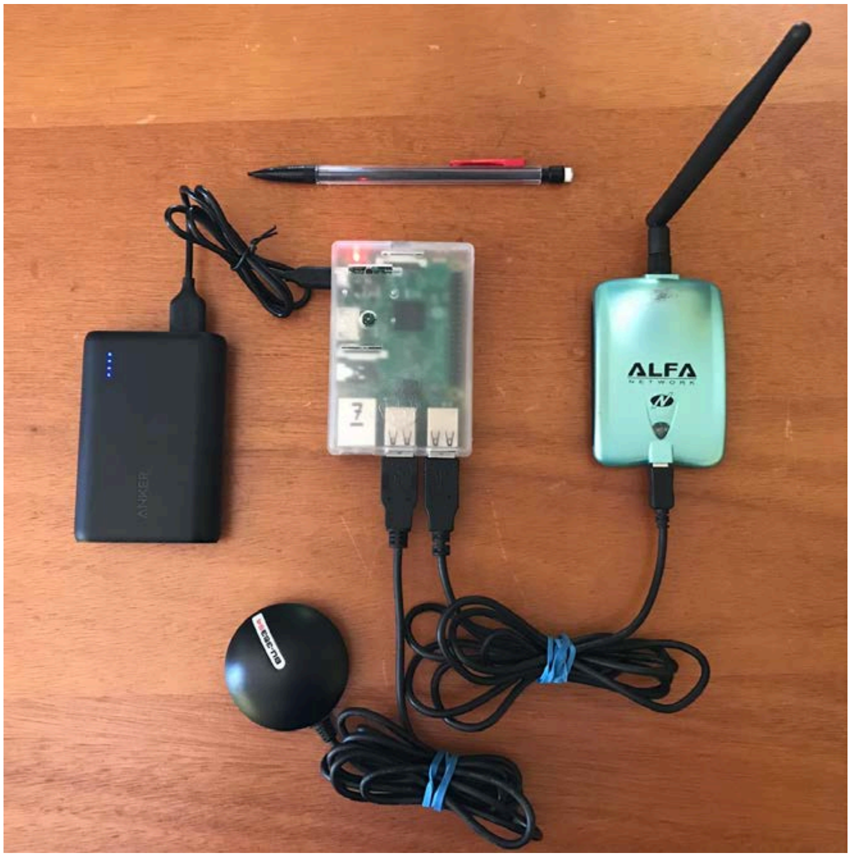


Figure 12. Sensor Node Hardware

2. Data Overview

The means of data formatting, management, and transmission are key for sensor networks. Wireless sensor networks and IoT devices are almost always operating in a power and bandwidth constrained environment. These limitations directly affect how data is structured and transmitted. Optimizing this becomes an ongoing challenge and an often-researched problem. In this situation, there is the potential for a node to receive hundreds, even thousands of AP beacons in a second; how we collect, analyze, summarize, format and transmit this data will have not only a significant impact on the performance of the network but also on the power requirements and overall lifetime of the nodes.

In order to keep processing to a minimum and formatting as simple as possible, data is sent as a simple byte string. Collected data is compiled using a simple Python dictionary and some nested lists then converted to a byte string for transmission across the network via MQTT message. Two main dictionaries are used – one containing all the requisite GPS data for each report and another nested within that dictionary which contains all the specific 802.11 data (Figure 13). Upon receipt by the broker, the byte string will be reformatted and processed as necessary—ingested into the database and forwarded to all subscribed nodes. By keeping data formatting as simple as possible, the node will save time and power that would otherwise be wasted with more complex data types in unnecessary places.

It also logically follows that no more data should be transmitted than required to keep records current. For example, if a node has seen no new APs since the last report, it should not retransmit the same data that has already been reported. The same is true for GPS data reporting. If the node has not moved more than ten meters since the last report, no new location data is reported. Data retransmission only serves to deplete valuable bandwidth and as such, should be minimized. Additional optimization of traffic and reporting is discussed in the Network section.

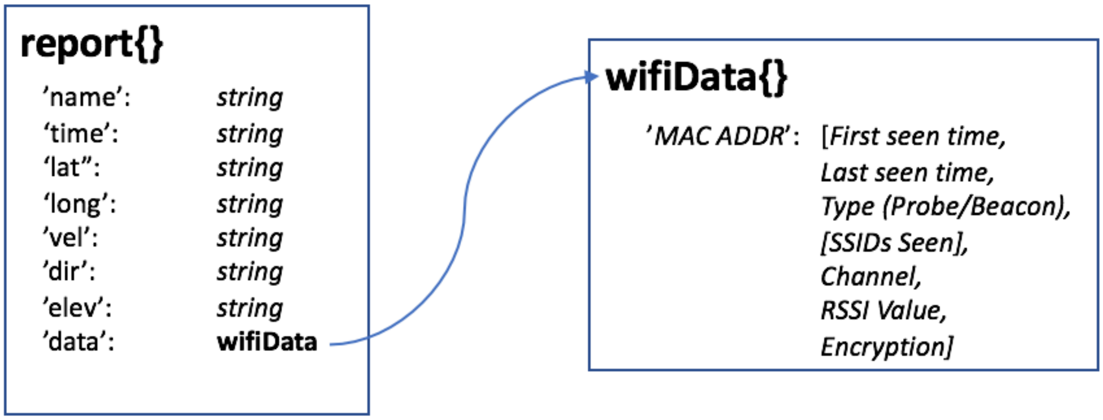


Figure 13. General Reporting Format for Sensor Nodes

D. SERVER / BROKER PLATFORM DEVELOPMENT

The central server provides much of the administrative support for the sensor network. Its primary responsibilities are acting as the VPN server, a redundant database server, and as the MQTT broker. In the centralized network structure, it also serves as the wireless AP and DHCP server for the network. It is important to note that this server/broker construct is not a limiting factor. The server platform and the following capabilities can more accurately be seen as functional requirements for the network as a whole, but the actual structure by which those functions are implemented is completely up to the developer. In other words, the functionality of the MQTT Broker, VPN server, and Database server must exist but having them all on one server is just one of many ways to implement it; there is no system requirement for them to be collocated. In fact, should operational requirements dictate separation, the system could just as easily be implemented with three physically separate devices–one for each server and one for the broker.

As with the nodes, the hardware for the server is a Raspberry Pi 3 Model B. For more complex network architectures, there can be multiple server platforms. The additional platforms will strictly serve as MQTT brokers for additional “zones” of collection, each with its own set of MQTT clients.

1. MQTT Broker

As the MQTT Broker, the server is primarily responsible for (1) managing connections to clients and (2) controlling which clients receive messages associated with topics to which they have subscribed. The broker also establishes the accounts used for node authentication. In this case, each sensor node has a separate username and password to prevent rogue nodes from joining the network. The broker also stores the “last will” of a client. This “last will” is published to the network in the event that a client is suddenly disconnected from the broker. In our case, this is very important as it highlights collection gaps for the network and allows other nodes to adjust accordingly. The broker will also store and forward messages that have been compiled under a specific, subscribed topic, once a client is back online.

2. VPN Server

The VPN server is created using PiVPN, an adaptation of OpenVPN developed for ease of implementation on a Raspberry Pi [30]. The main purpose for the VPN server is to provide access to the sensor network for the system controller and any decentralized nodes. Each sensor node, as well as the system controller, is loaded with unique OpenVPN configuration files allowing connections to the server. This enables a wider range of network architectures as discussed earlier. These certificates are all managed by the VPN server and can be revoked at any time should a node become compromised.

3. Wireless Access Point

An additional Alfa AWUSO36NH 802.11b/g/n USB network interface adapter is used to provide a longer-range access point than the built-in wireless interface. To configure the WAP portion of the server, the *hostapd* and *dhcpd* Linux packages are used. The WAP is configured with standard WPA2 encryption and does not broadcast its SSID. DHCP is configured for a basic /24 wireless network, with static IPs set for current nodes, and the remainder of the IP space for additional nodes to be used at a future time. For the purpose of this thesis, the WAP configuration only supports a centralized, infrastructure-based network, not an ad-hoc network.

4. MySQL Server

To provide a redundant database for the network, MySQL was configured on the server [32]. A single Python program is used to subscribe to all MQTT topics on the network and, based on the message format, the program parses the data, populating the database using the MySQL connector library [33]. The database is comprised of all reports that have been submitted across the network and can be used to conduct additional analysis post collection.

E. CONTROLLER DEVELOPMENT

The controller serves as the user interface for personnel accessing the sensor network. It provides the option to graphically display the nodes and their respective locations, allows the user to run a variety of queries on collected data, and provides an interface through which nodes can be individually tasked and controlled. The hardware used for the controller is simply a laptop with a functioning Internet connection. It connects to the sensor network over a VPN and interacts almost exclusively with the top-level server of the network.

1. Database Connectivity

Connectivity from the laptop to the database server is over a VPN through the MySQL client software [32]. Once a connection to the MySQL server is established the user is able to filter and display all raw data collected using a variety of queries. The user also has access to network processed data providing additional information on actual status of various emitters.

2. Data Visualization

Once the controller has established the desired filter parameters, it is exported to a KML file viewable with Google Earth. The export itself is accomplished using the *simplekml* Python library [34]. This enables straightforward formatting of data in KML files.

F. NETWORK

The network itself provides the overall processing power behind the data collection. Traditional sensor networks rely on the network edge almost exclusively for data collection. In other words, the sensors feed data back to a central repository for consolidation and analysis. A sensor is not aware of what other sensors around it see, where they are located, and status of the environment as a whole. Due to this one-dimensional use, a sensor node does not add any value to the network beyond data collection, and in many cases, there can be significant duplication of data across the network. This results in wasted bandwidth and wasted power. TLCSN takes a different approach, focusing on in-network processing of data vice the traditional utilization of backend analytics. This leverages each node’s awareness of what the network as a whole is seeing to better tailor collection and analysis.

1. In-network Processing

Due to resource limitations, a single sensor node alone is limited in the amount of analysis that it can perform. In the case of TLCSN, a node can report whether it is collecting information on a particular AP/client or not; essentially, whether it sees a signal or not. The result is very binary and while data from many sensors can be compiled into a central database, there is still a significant amount of follow-on analysis that must be done to distill the raw data down to important events.

Consider a simple example. One analytic goal may be to differentiate between static and dynamic APs in a specific littoral region. Essentially, to identify what APs are on-board specific maritime vessels and what APs are static and part of harbor, or shore-based, infrastructure. This can be done by reviewing data from a server after collection is complete. However, to do that each data point must traverse the entire network back to the central database, using up valuable backhaul bandwidth. Another way is for each node to subscribe to the unique collect of all other nodes within the zone. If each sensor node maintains a local database for all the data-points seen in a zone, each node is able to quickly screen for duplicate data (overlap of node collection footprints), identify movement of an AP or client between nodes within the zone, and isolate static APs. These specific events can then be reported to the broker and back to the analyst. So instead of pushing a firehose of raw data back to the server, thereby depleting limited bandwidth, only a few key data points ever leave the immediate zone.

To further illustrate, consider a small collection network of five nodes spread across a geographic area. Each node runs its initial collection and publishes it to the broker. The broker aggregates the data, filters duplicates, and publishes it to a master topic to which each node is subscribed. That data has now been received by each node in the zone. At this point, after one full collection cycle, each node has a baseline of the zone's environment. All further collections can now be compared to that baseline at the node level. No collection has been sent back to a central server at this point, and each node can begin to infer trends. Suppose an emitter is no longer seen by Node 1 but is seen by Node 4 That emitter is immediately characterized as mobile and the MAC address is published in a topic for attention at the zone level. If an emitter is no longer seen by any sensor node in that zone the broker for that zone immediately publishes an alert to neighboring brokers as a potential mobile target. If the suspect emitter then arrives in a neighboring zone it is then elevated to the network level as a target traversing regions of interest. For the first time, data is pushed all the way back to the home server. All of the processing to this point has taken place within the network, data filtering happening at each level until a discrete event, vice a mass of raw data, emerges at the top.

2. Hybrid Architectures

A hybrid of the centralized, ad-hoc, and distributed architectures can be used to create a tiered-network with multiple sensor zones. In this model, each zone has a broker for data distribution within that zone. That broker also serves as a bridge to brokers in neighboring zones. Each broker can then be connected back to the sensor network via a direct network connection or through a VPN tunnel. As such, the MQTT topics now form a more significant hierarchy—network/zoneX/nodeX—allowing for data to be filtered by whatever means necessary. As discussed earlier, there can be considerable benefit to leveraging in-network processing both within a zone, as well as across all zones (Figure 14).

Each tier above the node level formulates a more and more refined picture of the data being tracked. At the node-tier, all the raw data is viewable, duplicate collections can be filtered out, collection areas can be optimized, and local trends can be identified. At the zone-level, the clarity as to which emitters are static within the zone, which are dynamic within the zone, which have just entered the zone, and which have left the zone is apparent. Each zone broker can immediately identify when a previously seen emitter enters its zone, indicative of transit between the two locations. This becomes an event at the network-level, which can be used to identify emitters of interest and their associated patterns.

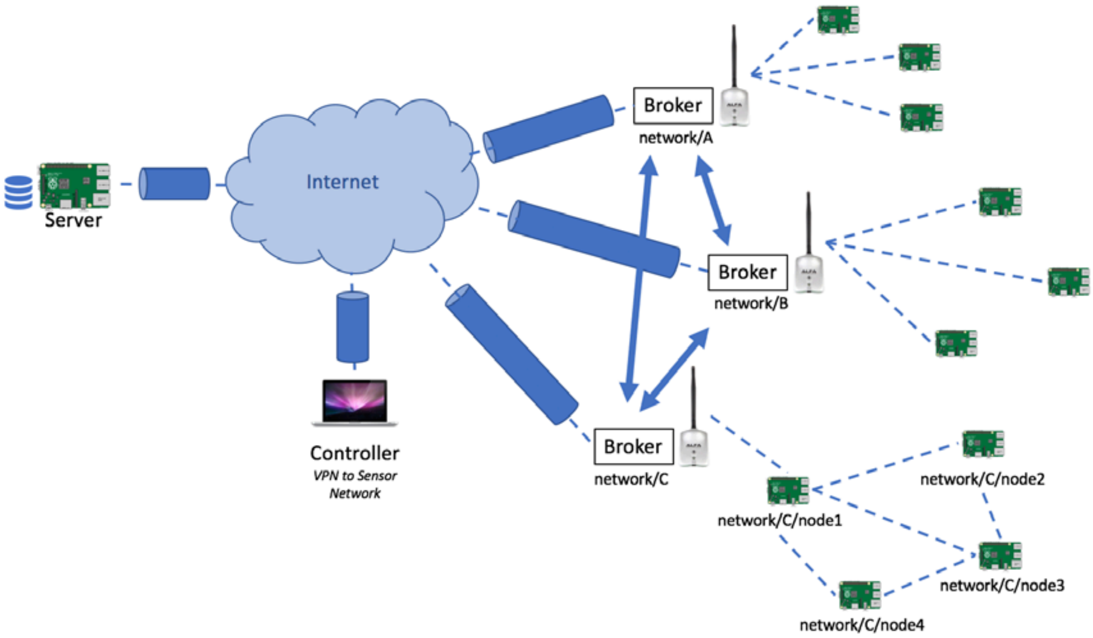


Figure 14. Model of a Tiered, Hybrid Network Architecture for Leveraging In-network Processing

G. SUMMARY

This chapter addressed the structure and components of the wireless sensor network as a whole. Three different types of possible network architectures were discussed, critical system hardware components were noted, and software libraries and packages potentially used in the implementation were identified. In addition, the means of data management, from formatting to general flow, were outlined, along with some initial explanations of in-network processing and ways of optimizing the use of nodes for data characterization and analysis.

In Chapter IV, the specific implementation of in-network processing as well as testing and basic demonstration of the TLCSN are discussed.

IV. IMPLEMENTATION AND TESTING

This chapter discusses the implementation of in-network processing as well as the levels of system testing and technology demonstration that were conducted for TLCSN. The testing is broken down into three separate phases focusing on the overall performance of the sensor nodes and network. The first phase is based on testing a single node and evaluating its ability to collect and ingest the targeted data. The second phase tests the operation of the network with a single zone of three nodes and its associated broker. It examines the effectiveness of the network not just collecting but, more importantly, characterizing the data collected. The third phase tests a scenario based around three separate zones of nodes. It evaluates the effectiveness by which the network as a whole can characterize the movement of targets in and out of the various zones and report that data to the central controller.

A. IMPLEMENTATION

In the previous chapter the general system components and potential network architectures were discussed at a high level. The software libraries and specific hardware components used were identified. The focus is now on the specific software make-up for each node, the actual network architecture that is used, and how the data is collected and analyzed both within a collection node, and across the network.

1. Network Structure

When discussing in-network processing a common lexicon of terms is used to prevent confusion. The network structure referred to throughout this chapter is shown in Figure 15. The term *node* refers to a single sensor node. In the illustrated case, each *zone* is made up of three nodes. Each of these *zones* is connected to the other two resulting in a three-zone network. The red dotted lines represent inter-zone connectivity, while the black dashed lines represent intra-zone connectivity. The circles represent the theoretical collect area for that specific node. The number of nodes used is for demonstration purposes only and can be scaled to much greater levels to cover more area both across a zone and across the entire network.

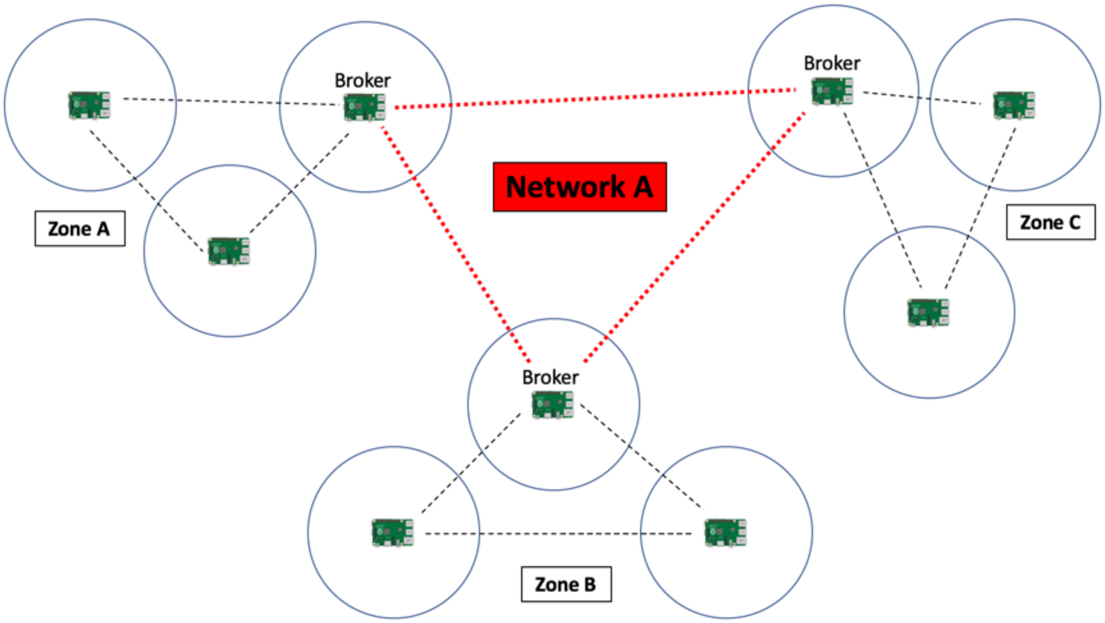


Figure 15. Sample Network Construction

2. Modules

The sensor node software consists of four main modules each providing a specific function. They are identified as the collect, connect, process, and database modules. Each is responsible for managing roles specific to that function. Figure 16 displays each module and key functions associated with it. The data discussed in this section is referred to in two different ways: as *local* and *network data*. Relative to a specific node, local data is that collected by the sensors of that node. Network data refers to any data being reported across the network from another collection node.

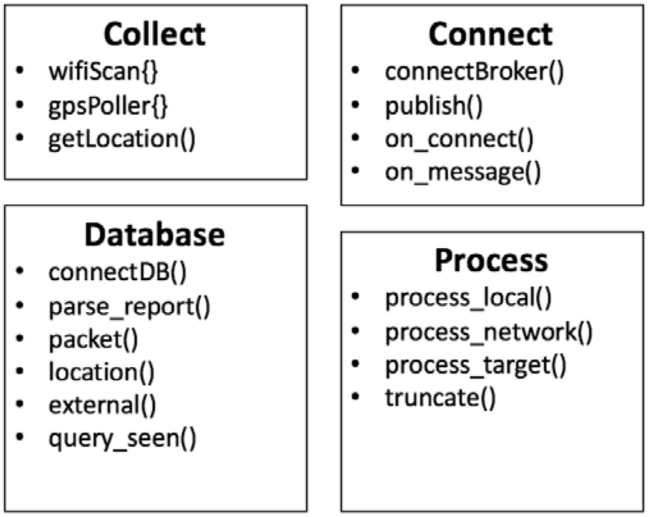


Figure 16. Sensor Node's Functional Modules and Major Functions

a. Collect

The *collect* module provides all the data collection functionality for the various sensors in a node. It controls the threads responsible for polling and reporting GPS data, as well as for conducting the actual collection and decoding of the 802.11 packets. The GPS polling thread constantly receives GPS coordinates from the device, but only reports them when queried. The 802.11 collection works in cycles, with each cycle having a defined run time and wait time. The run time specifies how long the collection cycle should run. The wait time defines the time the thread should sleep between actual collection cycles. For example, if a user wished to collect data for only 15 seconds every minute, they would set a 15 second run time and a 45 second wait time. These values can be dynamically defined and changed based on various other factors, such as the general RF environment, time of day, etc. The 802.11 collection runs continuously based on the collection parameters specified.

b. Connect

The *connect* module manages the network connections of the sensor node. It authenticates and establishes the connection with the MQTT broker, subscribes the node to the appropriate MQTT topics, receives data published by other nodes, and publishes the appropriate data collected by its own sensors. All external connectivity for the node is managed through the connect module. When data is received from the broker, the connection node filters it based on what topic is received and forwards the data to the collect function in the processing module for initial analysis.

c. Database

Each sensor node runs its own MySQL database to store collected and received data. The database contains four separate tables for storing specifics of the packets collected. The tables store data for raw local data, received network data, the location data of nodes across the network, and the specifics of targeted selectors of interest, as shown in Figure 17. As the name suggests, the *database* module controls all database related functionality for the sensor node. It is responsible for correctly ingesting data into the various tables and providing query results to the other modules. This includes properly parsing various reports, both local and network. All data is passed to the *database* module only after it has been screened and analyzed by the *process* module.

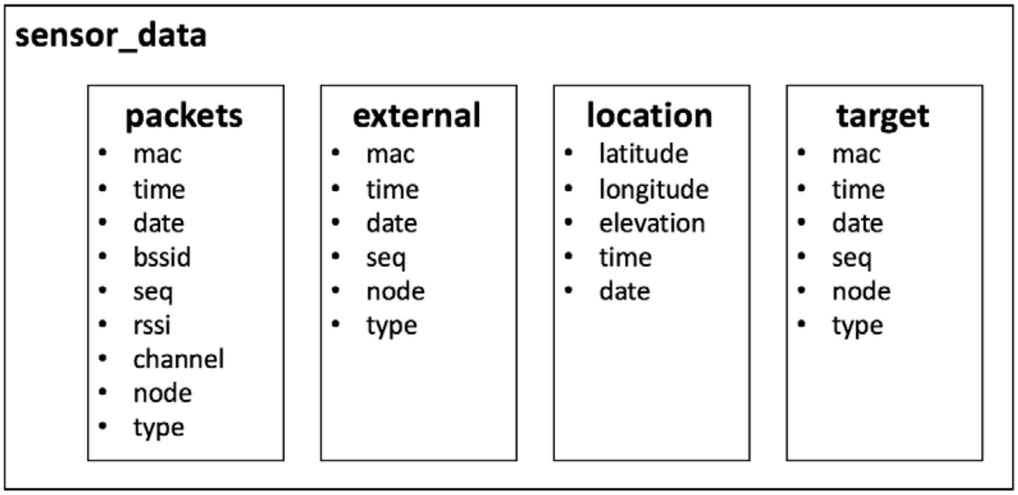
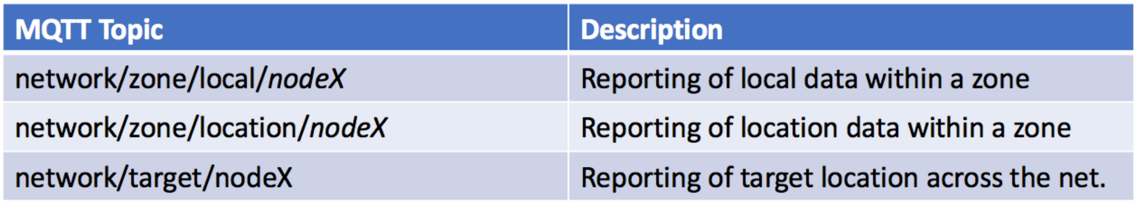


Figure 17. Database Schema Used for Each Sensor Node

d. Process

The *process* module conducts the bulk of the analysis for the node. It has two main functions—processing local data (collected by the home node) and the network data (data received by the home node that was collected by other nodes) and ensuring that each type is properly ingested and reported across the network. The MQTT topics used are listed in Table 1. The specifics of the processing methodology are addressed in detail in the next section.

Table 1. Main MQTT Topics Used in TLCSN



MQTT Topic	Description
network/zone/local/*nodeX*	Reporting of local data within a zone
network/zone/location/*nodeX*	Reporting of location data within a zone
network/target/nodeX	Reporting of target location across the net.

3. In-network Processing Methods

As previously stated, *local data* is data derived by the sensor node itself and *network data* is that received from the surrounding nodes of the network. At the most fundamental level, there are two questions that can be posed in initial analysis of the 802.11 data collected: “Have I seen this MAC address before?” and “Has anyone else on the network seen this MAC address before?” This simple comparison is the basis for the identification of movement and fundamental characterization of targets.

a. Local Data Processing

Data collected locally is processed in accordance with Figure 19. Decisions are made as illustrated, with final actions being listed in each black box. The actions taken following analysis of data are divided into two categories: the ingest of data into a specific database table and the reporting of data across the network under a specific topic.

There are two types of data ingestion with local data: ingestion into the *packet table*, or into the *target table*. After initial filtering by the collection module, all locally collected packets will be ingested into the packet table. This is to maintain a detailed warehouse of all raw data collected. Data will only be ingested into the target table if it is associated with a MAC address deemed to be a mobile target. Mobile characterization is made by noting the collection of the specific selector by more than one node. In other words, as long as the collection areas of two sensors do not overlap, if the same MAC is seen in both collection areas at different times, it can be assumed to be mobile.

There are also two types of reporting that happen: reporting to the *local* MQTT topic and reporting to the *target* MQTT topic. The local topic is subscribed to by all the nodes in the local zone. The target topic is subscribed to by all the nodes in the entire network. By referencing Figure 18, it is apparent that data is only reported to the local topic when a MAC is seen for the first time by a node. If this MAC has already been seen locally by another node in the network, it will be reported in the target topic so that its movement can be tracked. One general rule of thumb follows throughout the network—only the collecting node itself has the ability to elevate a MAC to the target table. No node will elevate based strictly on network provided data. This will be highlighted in how network data is processed by a node.

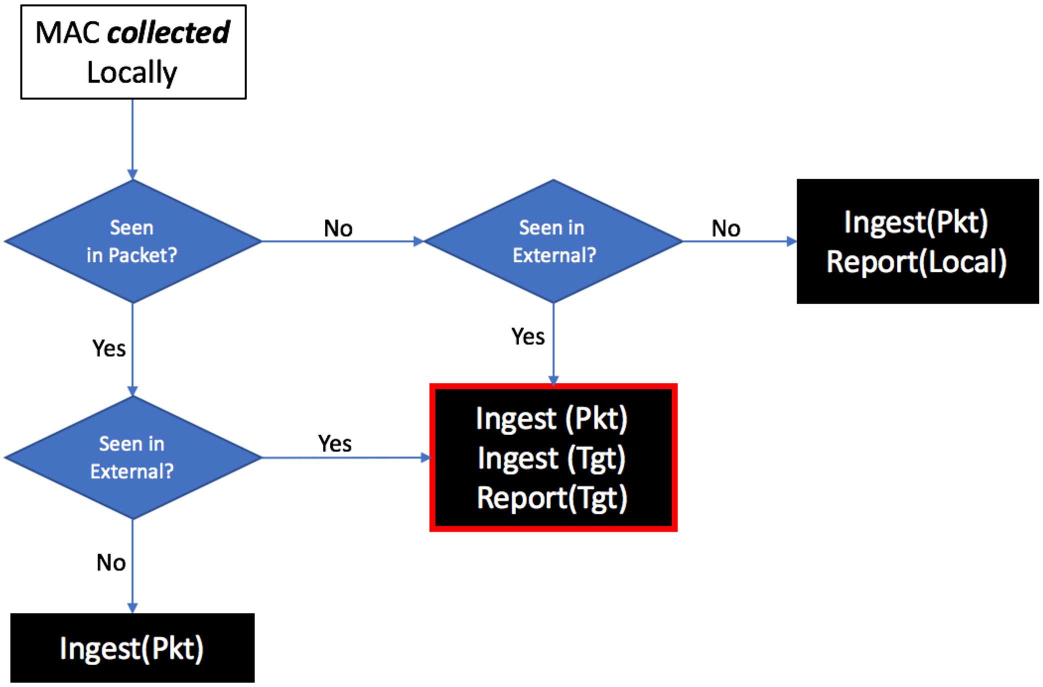


Figure 18. Processing Algorithm for Locally Collected Data

b. Network Data Processing

Data that is received from the network is processed in accordance with Figure 19. This data provides the necessary information to give each node a complete picture of the zone's collection environment. Data transmitted across the network is a limited and truncated version of the packet data that was collected locally. For example, if the same beacon has been seen 20 times by the same node and no other, it will only be reported across the network the first time it is seen. All the rest of the nodes need to know is that this specific MAC was seen by node X at a given point in time.

As with the processing of local data, network data also has two types of ingestion: into the target table or the external table. However, only one type of report will be triggered: a report under the *target* topic. As shown in Figure 19, if a MAC has never been seen in the packet table, it will simply be immediately ingested into the *external* table. The same is true if it has been seen both locally and over the network before. The key case here is when a MAC has been seen locally, but there has been no network reporting of it. This case indicates the first discovery of a mobile target and the node reports and ingests it as such.

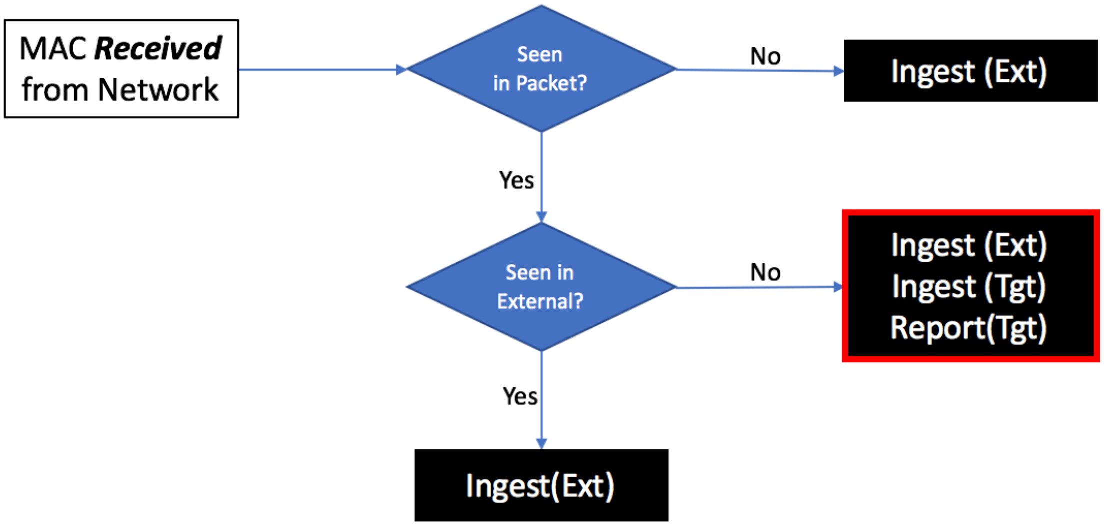


Figure 19. Processing Algorithm for Data Received from the Network

B. SINGLE NODE TESTING

The testing of a single node provides a baseline for whether or not the core data and GPS collection capabilities are happening correctly. It also confirms the ingestion of the appropriate data into the appropriate tables in the correct format. The basic sensor node functionality should not be taken for granted as benchmarking performance before moving on to more advanced network structures can help in isolating issues later. If the collection software is not working with a single node system, issues need to be identified now before scaling the network up to greater levels.

1. Test Plan and Criteria

Single node testing addresses four main functional areas—data collection, database ingestion, reporting, and basic characterization.

a. Collection and Ingestion

From a collection standpoint, the key test is to ensure the node is collecting the correct GPS data for its location and an accurate representation of the 802.11 packets transmitted in its vicinity. GPS data is correlated with known points to verify accuracy. The 802.11 data collected is compared against that seen by the packet sniffer *airodump*. This indicates if the node is receiving similar representative collection from APs and clients in the vicinity. The 802.11 tests are run on the same wireless interface card with the same antenna to ensure accurate comparison.

The ingestion tests ensure that once data is collected, it is correctly ingested into the packet database. It confirms that GPS data is correctly ingesting into the Location table and all other packet data into the Packets table. The results of the raw collection compared with those in the location and packet tables to ensure complete ingestion.

b. Reporting

This test ensures that the node is able to properly connect to the MQTT Broker, and that data is being correctly reported across the correct MQTT topics based on the policies dictated. The topics of focus are the network/zone/node topic and the network/location topic.

c. Basic Characterization

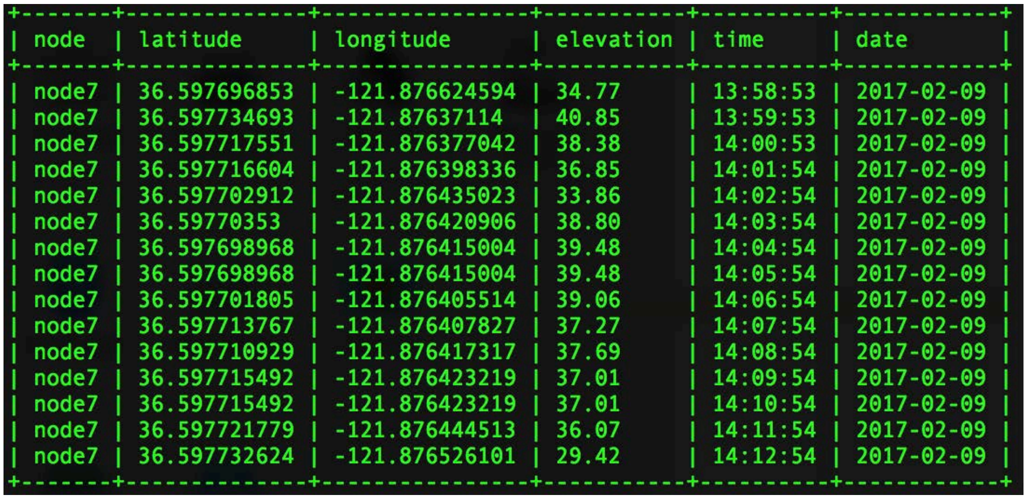
The main characterization that happens with a single node system is the identification of a new transmitter in its area. If an AP or client appears for the first time, this is noted and reported across the network/zone/nodeX MQTT topic. This system will not be reported across the network again in a single node set-up.

2. Test Results

All testing was successful with the single node configuration. The specific results for each functional area are outlined below.

a. Collection and Ingestion

Testing of the collection module was successful. The expected GPS coordinates were collected and parsed by the module to provide latitude, longitude, elevation, time, and date. GPS data from Google Earth was compared with the results to ensure correctness. A sample of the collect is shown in Figure 20. Results were comparable when an 802.11 collection was run sequentially using first the sensor module and then using *airodump*. In three different urban and suburban testing environments, the number of access points seen was the same for both methods of collection. The number of clients seen cannot be as reliably compared due to client movement and frequency of probe request transmission, however in comparing the results there was still significant overlap. The combination of these two tests indicates that the software is correctly collecting, decoding, and filtering the packets.



```
+-------+--------------+----------------+-----------+----------+------------+
| node  | latitude     | longitude      | elevation | time     | date       |
+-------+--------------+----------------+-----------+----------+------------+
| node7 | 36.597696853 | -121.876624594 | 34.77     | 13:58:53 | 2017-02-09 |
| node7 | 36.597734693 | -121.87637114  | 40.85     | 13:59:53 | 2017-02-09 |
| node7 | 36.597717551 | -121.876377042 | 38.38     | 14:00:53 | 2017-02-09 |
| node7 | 36.597716604 | -121.876398336 | 36.85     | 14:01:54 | 2017-02-09 |
| node7 | 36.597702912 | -121.876435023 | 33.86     | 14:02:54 | 2017-02-09 |
| node7 | 36.59770353  | -121.876420906 | 38.80     | 14:03:54 | 2017-02-09 |
| node7 | 36.597698968 | -121.876415004 | 39.48     | 14:04:54 | 2017-02-09 |
| node7 | 36.597698968 | -121.876415004 | 39.48     | 14:05:54 | 2017-02-09 |
| node7 | 36.597701805 | -121.876405514 | 39.06     | 14:06:54 | 2017-02-09 |
| node7 | 36.597713767 | -121.876407827 | 37.27     | 14:07:54 | 2017-02-09 |
| node7 | 36.597710929 | -121.876417317 | 37.69     | 14:08:54 | 2017-02-09 |
| node7 | 36.597715492 | -121.876423219 | 37.01     | 14:09:54 | 2017-02-09 |
| node7 | 36.597715492 | -121.876423219 | 37.01     | 14:10:54 | 2017-02-09 |
| node7 | 36.597721779 | -121.876444513 | 36.07     | 14:11:54 | 2017-02-09 |
| node7 | 36.597732624 | -121.876526101 | 29.42     | 14:12:54 | 2017-02-09 |
+-------+--------------+----------------+-----------+----------+------------+
```

Figure 20. Snapshot of GPS Data Collected

b. Reporting

To test reporting, the node must be able to successfully connect to an MQTT broker. To test this basic functionality the node connected to a broker on the local network. Within each main MQTT topic there is a subtopic that includes the name of the specific node. The node is subscribed to all the MQTT topics that it publishes but filters out the messages it sent to avoid duplication. In this situation, a simple script is used to ensure that each message the node publishes is subsequently sent out by the broker. The node receives the messages, but discards them before data ingestion. All messages are published to the correct MQTT topics as follows: network/zone/node1 for packet data to the local zone, network/location for the location data of the node, and network/target for the identification of specific target selectors that were detected by multiple nodes.

c. Basic Characterization

The sensor node correctly screened the 802.11 data and only reported across the network when a new selector was detected. To simulate a target, a preformatted message is published to the broker from a separate terminal. This simulates what the node would see if there were another sensor on the network. The test was successful and the MAC detected was elevated to the target table. This verified the logic behind comparing the local and network data to establish movement.

C. MULTI-NODE, SINGLE BROKER TESTING

The second phase of testing addresses a collection network with three separate nodes and a single broker. The overall testing goal was to determine whether an external controller can subscribe to the target topic and receive an accurate representation of what transmitters, both APs and clients, are moving between the various collection areas of each node. The logical setup of the network is shown in Figure 21. Node 2 operated as both the MQTT Broker as well as the VPN server on a local area network, while nodes 1 and 2 are connected to the network via VPN.

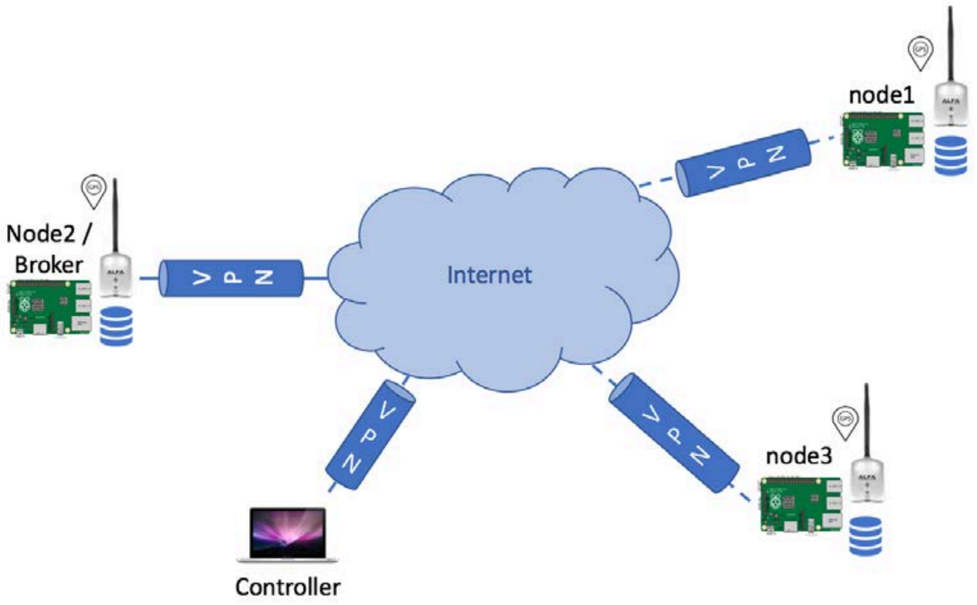


Figure 21. Multi-node Testing Network Diagram

The physical positioning of the sensor nodes for the test is shown in Figure 22. Each ring around the nodes has a 100-meter radius indicating the rough collection area for each location. They are positioned in a neighborhood, along a major thoroughfare, and at the Naval Postgraduate School. This initial setup demonstrates the identification, isolation, and then tracking of a phone being moved to and from the various locations. Later testing will involve the actual placement of the nodes in a maritime environment, whereas the intent here is to showcase the in-network processing and analytics provided by the network.

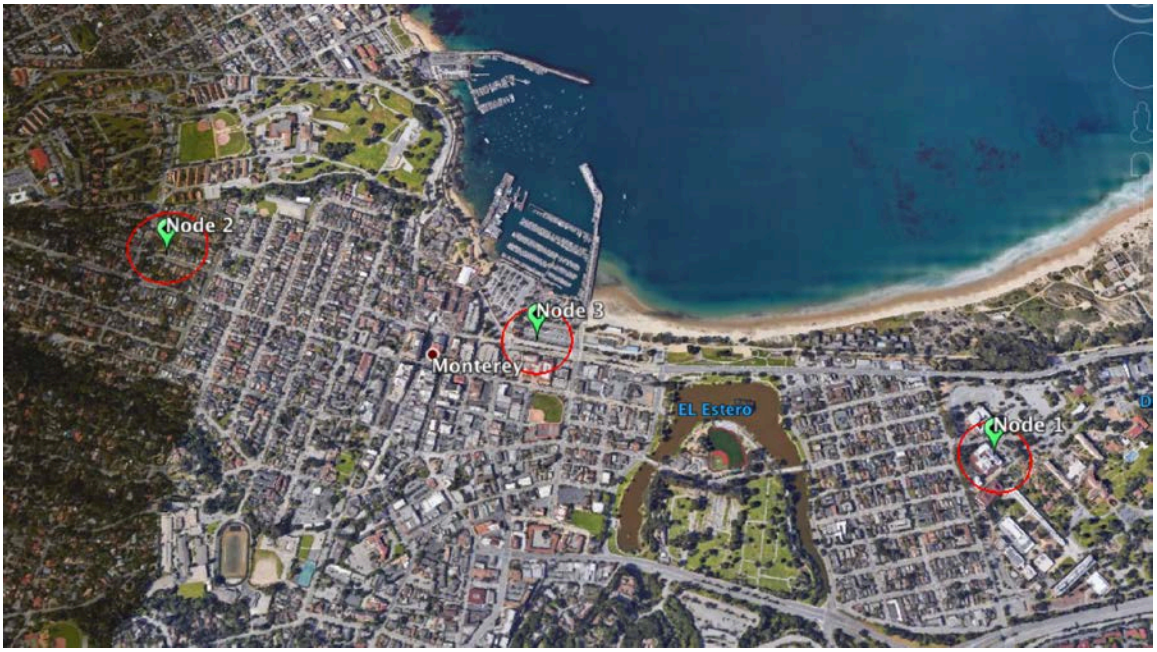


Figure 22. Physical Location of Sensor Nodes Using Google Earth Pro

1. Test Plan and Criteria

In this test, two separate 20-minute iterations are run. The first involves the movement of an 802.11 access point between the various nodes. An apple iPhone 7+, running in "hot spot" mode, will serve as the target device. The second iteration will involve the same iPhone, however it will only be acting as a client. It will move between the collection area of nodes 1 and 2, and will connect to a local wireless network at each location. It will remain in each area for two to three minutes, and then move back to the first location. The controller will be connected to the broker and subscribed to only the target MQTT topic. The controller has no previous knowledge of the target device. A successful test will end with the controller receiving real time identification, followed by more granular notification of the target's movement between the sensor node collection areas.

2. Test Results

The multi-node testing was ultimately successful and provided valuable data for future modifications and optimizations of the network. Testing results for each functional area are discussed in detail below.

a. Reporting

As demonstrated in the first testing phase, all three nodes were able to correctly collect GPS and 802.11 data from their environments. The key focus in this testing phase is a demonstration that the nodes can collaboratively populate the packet and external tables across the network. Each node has unique packet and external tables, however, when compared, each should still have a common list of all the distinct MAC addresses seen by the network. This illustrates that the data is indeed being received by the network as a whole and correctly distributed. After running all three nodes simultaneously for ten minutes, comparison of the data held in the packet and external tables revealed the same 226 distinct MAC address when combined for each node. This test was run on three separate occasions with the same results each time.

Additionally, collection cycles of different lengths were run to ensure that all data was being processed and reported. MQTT ensures delivery across the network if a quality of service (QoS) number of one or two is mandated by a publisher. However, it is important to make sure that data is not lost between the receipt of a message by a client, and its ingestion into the database, especially with multiple nodes reporting at the same time. The test set up involved collection cycles of five seconds, with no break between cycles except for the time needed to process and report the data. The nodes all started their collection at the same time and ran for ten minutes each. At the completion, the data was compared. Threading is used for each ingest process and each process has a separate database connection to prevent any collisions. Between the three nodes, no data was lost and tables contained the same distinct MAC addresses.

The target and location tables should be identical across the zone. This is verified and indicates that each node is correctly sharing and processing these two topics.

b. Characterization

Characterization happens in two steps. First, the device is identified as mobile by appearing in the collection area of multiple sensor nodes. Second, the actual movement is collected as it transits between zones. Once the device has been identified as a target, each time a transmission is collected, it is immediately added to the target table and published across the network. This provides much more accurate granularity as an observer can distinguish when a device arrives in a certain area, how long the device is in a node's area of collection, and when it leaves. Over time, the table will continue to populate, showing trends across days and weeks.

D. MULTI-NODE, MULTI-BROKER TESTING

The last phase of testing focuses on the ability to mesh multiple zones together and identify movement between them. This translates to the ability to connect tracking across multiple distinct geographic locations throughout an area of operations.

Utilizing multiple brokers, shown again in Figure 23, provides additional means to scale the overall network. In this abbreviated test, the collection nodes themselves are abstracted out to maintain a certain level of simplicity. Three brokers are set up to forward the one significant topic across zone borders—*network/target/nodeX*. This provides the absolute minimal bandwidth requirement needed for each zone to maintain awareness of the mobile targets within the overall network.

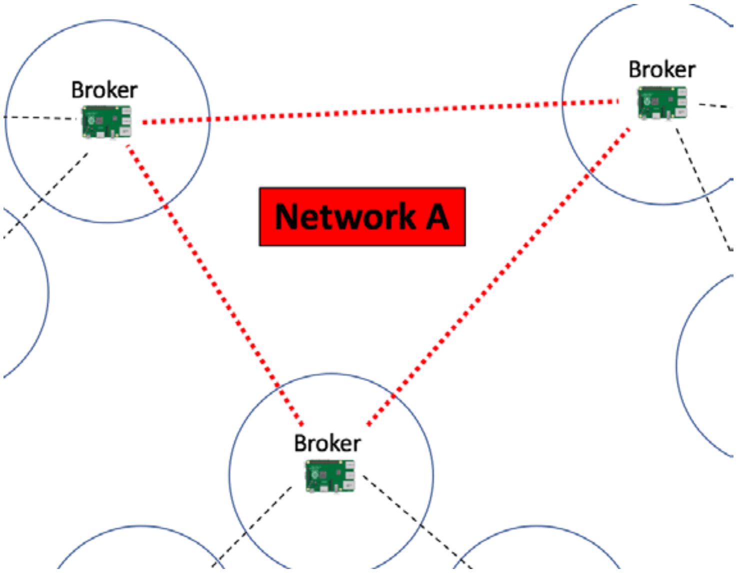


Figure 23. Illustration of Inter-zone Connectivity between Brokers

To accomplish this, forwarding of topics between brokers is enabled in the broker configuration files—also known as bridging. The specific topics that should be forwarded are identified and added to the configuration file. These topics synchronize between the brokers and are published within the respective zones as are any other topic. Hence, the target topic a node in Zone A receives will be the same target topic a node in Zone B receives. Testing of this concept was only with one simulated node per broker to demonstrate the capability. In the future, more work could be done to further optimize and explore other implementations of this concept.

E. SUMMARY

This chapter addressed the specific implementation used for TLCSN in the testing environment. The methods used for in-network processing and characterization were explained and the general testing plans were outlined. Formal testing was then conducted in three separate scenarios. First, the single node functionality was tested. After successfully demonstrating the ability of a node to collect, process, ingest and report GPS and 802.11 data, the test environment was modified to include multiple nodes in a single zone. Once the zone was able to successfully identify a mobile access point and track it within the zone, a third test was conducted. The last test addressed the ability to scale the network to multiple zones across a larger geographic area by using multiple brokers and forwarding topics. The next chapter assesses the key opportunities and methods highlighted by the TLCSN testing, as well as identifying key areas for future research and development.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

The goal of this study was fourfold. First, explore Navy policy and ISR requirements and determine the general breadth of maritime wireless sensor network use. Second, research and compile the necessary hardware and software components needed to create a low-cost wireless sensor node able to collect and process 802.11 wireless signals and GPS data. Third, provide a technology demonstrator able to collect and process wireless and GPS signals collected by a scalable sensor network. Last and most importantly, leverage in-network processing capabilities to enable the processing of network data without the use of a central database while isolating and reporting relevant events to the end user. Through exploring current policy, it is clear that the capabilities a maritime ISR sensor network could provide are in high demand. The explosion of IoT technology growth provides a wealth of technologies and hardware, which can support these capability requirements.

As a technology demonstrator, we built TLCSN—a network of nine collection nodes comprising three geographically separate zones. For the cost of less than $120, we were able to build and program an individual wireless sensor node. Each node consisted of power, collection, networking and processing capabilities. These capabilities leveraged in-network processing to remove any reliance on formal infrastructure or data back-haul for analysis.

The first construct of TLCSN consisted of a single zone with a network of three nodes. This configuration was tested and able to correctly identify, track, and categorize transmitters within the collection area of the zone. The architecture was subsequently expanded to include three separate zones, accomplishing the same fundamental tasks in a scalable fashion across a larger geographic area. These three zonal networks communicated and collaborated with each other to help identify and track objects of interest.

The hardware components used to build each node were widely available pieces of commodity electronics. The software libraries used were all open source and available at no cost to a developer. MQTT was used at the application layer to create the network, and the MOSQUITTO broker and client software were used to control the connections. Simple MySQL databases were used on each of the nodes with the overall control software being written using Python 2.7.

During the implementation and testing phases, the functionality of the individual sensor node, followed by its performance in a networked environment, were documented. The sensor node performed well in both the simple collection of data as well as the critical analysis and transmission of data across the network. The actual range of each collection node was not explicitly tested in every environment; however, access points were seen up to 100 meters away in suburban settings. The ability of the sensor nodes to collaborate, each maintaining detailed records of its own collection, enables the in-network processing capabilities desired. As a result, the sensor network no longer relies upon a central repository for the consolidation of data, but rather distributes the responsibility across the network itself and leverages each node for the reporting of relevant events. TLCSN’s use of in-network processing provides an example of how a sensor network can directly support the ISR collection requirements of a commander, without relying on an expensive, bandwidth-intensive link back to a central database. This provides the potential to increase collection footprints in both the tactical maritime and ground operating environments where these high bandwidth links may be unavailable.

TLCSN does not provide an immediate solution to existing ISR gaps at the tactical edge; nor could it immediately replace existing ISR collection systems. For either of these to happen, significantly more work would need to be done from testing, development and productization standpoints. What it does provide is how emerging IoT technologies can drive the development of future sensor nodes and sensor networks.

The construction of the TLCSN widens the scope of visibility for future sensor network development based on the IoT technology. It also provides a platform for the integration of additional sensors and physical network technologies on a mission-dependent basis. The choice of components is an example of how to mitigate many of the past, cost-prohibitive limitations, of wireless sensor network employment. TLCSN provides a framework upon which future sensor networks can be developed and deployed, catering to a low cost of acquisition, and a broad scope of tactical usage. With the overall architecture in place, each component of the system can be modified or refined to provide increased versatility and operational effectiveness. For the maritime environment, various buoy systems exist and the existing sensor node hardware is easily adapted to fit various forms. For traditional ground based operations, the existing network and node constructs provide simple and expendable capability that can be used to increase ISR collection by tactical forces.

B. FOLLOW-ON WORK

Additional proposals for follow on work are divided into four main categories addressing the network itself, the storage and manipulation of collected data, the sensors and technologies targeted, and the actual physical employment of the TLCSN.

1. Physical Network

The physical and transport layers of TLCSN were abstracted out of most of the tests conducted in this study, as the emphasis was placed on the sharing and analysis of data within the network. Future testing should address implementing the TLCSN across various ad-hoc network technologies demonstrating true infrastructure-less capabilities. There are various “mesh” networking technologies that provide support for TCP/IP connectivity at a low cost. In this study sensor nodes were also assumed to be static. The additional issue of having mobile nodes could also be explored and expanded upon to provide a wider range of network uses.

2. Data Storage

In this study, the storage and recovery of data was conducted using simple MySQL databases and basic queries. More research should be conducted into optimizing how data is managed both within each sensor and across the network. Finding ways to generate faster efficient queries to extend battery life and minimize processing power could provide valuable improvements to the system. Additional optimization of data transmission to take into account the possibility of overlapping collection areas will also increase the scope of use for the system as well as its efficiency.

3. Target Medium

The 802.11 wireless technology was used because it provides a large amount of data with significant potential for intelligence gain. However, depending on the area of operation, there may be many other wireless technologies that are more relevant for collection. The Raspberry Pi possesses the processing power to support many more sensors as well as conduct additional tasks such as image processing and facial recognition. Expanding the TLCSN to provide a wider range of data could increase the operation impact of the system. Thus, it may be useful to expand the collection scope to look at various supervisory control and data acquisition (SCADA) protocols that might be more relevant to mission taskings.

4. Testing Environment

The TLCSN has significant potential in both the maritime and shore-based environments. Additional testing and deployments should be conducted in both environments, as well as hybrid combinations of the two. A system of multi-intelligence sensors that is able to provide data across the littorals, both from the sea and the shore in concert, will greatly enable commanders of maritime and amphibious forces.

LIST OF REFERENCES

[1] I. Lunden. (2015, Jun. 2). 6.1B smartphone users globally by 2020, overtaking basic fixed phone subscriptions. Tech Crunch. [Online]. Available: https://techcrunch.com/2015/06/02/6-1b-smartphone-users-globally-by-2020-overtaking-basic-fixed-phone-subscriptions/

[2] M. Asadullah and A. R. Celik, “An effective approach to build smart buildings based on Internet of Things (IoT),” *J. Basic Appl. Sci. Res.*, vol. 6, no. 5, pp. 56–62, Apr. 2016.

[3] M. Maupin, “Fighting the network : MANET management in support of littoral operations,” M.S. thesis, Dept. Info. Sci., Naval Postgraduate School, Monterey, CA, 2016.

[4] D. Majumdar. (2016). The U.S. Navy just Gave Us the Inside Scoop on the “Distributed Lethality” Concept. [Online]. Available: http://nationalinterest.org/blog/the-buzz/the-us-navy-just-gave-us-the-inside-scoop-the-distributed-18185

[5] T. Rowden, P. Gumataotao, and P. Fanta. (2015, Jan.). Distributed lethality. *Proceedings Magazine* [Online]. Available: http://www.usni.org/magazines/proceedings/2015-01/distributed-lethality

[6] J. Gertler, “U.S. unmanned aerial systems,” Congressional Research Service, Washington, DC, Rep. R42136, Jan. 2012.

[7] NAVAIR – STUAS. (2016). Naval Air Systems Command. [Online]. Available: http://www.navair.navy.mil/index.cfm?fuseaction=home.display&key=4043B5FA-7056-4A3A-B038-C60B21641288

[8] R. L. Arellano, R. G. Pringle, and K. L. Sowell, “Analysis of rapid acquistion processes to fulfill future urgent needs,” Joint Applied Project, Dept. Business and Public Policy, Naval Postgraduate School, Monterey, CA, 2015.

[9] Hype cycle for 2016. (2016, Aug. 16). Gartner Inc. [Online]. Available: http://www.gartner.com/newsroom/id/3412017

[10] Number of mobile phone users worldwide 2013-2019. (2015). Statista. [Online]. Available: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

[11] C. D. Becker, “FY16 PEO C4I science and technology focus areas and capability gaps,” Program Executive Office C4I, San Diego, CA, Rep. C4I/057, May 2016.

[12] E. D. Haider, “Unattended ground sensors and precision engagement,” M.S. thesis, Dept. Def. Anal., Naval Postgraduate School, Monterey, CA, 1998.

[13] Gen 4 integration unit datasheet. (n.d.). Persistent Systems. [Online]. Available: http://www.persistentsystems.com/integration-board-gen4/. Accessed: Jan. 14, 2017.

[14] TW-600 Ocelot datasheet. (n.d.). Trellisware. [Online]. Available: https://www.trellisware.com/wp-contentDir/uploads/2016/05/TW-600-Ocelot-Datasheet.pdf. Accessed: Jan 14, 2017.

[15] StreamCaster 3822 datasheet. (n.d.). Silvus Technolgies. [Online]. Available: http://silvustechnologies.com/products/streamcaster-3822. Accessed: Jan 14, 2017.

[16] K. Stewart. NPS special report : supporting the COCOMs – U.S. Southern Command. (2014, Apr. 24). Naval Postgraduate School. [Online]. Available: http://www.nps.edu/About/News/NPS-Special-Report-Supporting-the-COCOMs-U.S.-Southern-Command.html

[17] Liquid Robotics product specifiations: Wave Glider SV3. (2013). Liquid Robotics. [Online]. Available: https://www.liquid-robotics.com/platform/overview/

[18] B. Honegger. NPS pioneers “Seaweb” underwater sensor networks. (2010, Aug. 2). Naval Postgraduate School. [Online]. Available: http://www.nps.edu/About/News/NPS-Pioneers-Seaweb-Underwater-Sensor-Networks.html

[19] Sonobuoy tech systems. (n.d.). Sonobuoy Tech Systems. [Online]. Available: http://www.sonobuoytechsystems.com/products/. Accessed: Jan. 28, 2017.

[20] K. D. Jones and V. N. Dobrokhodov, “Multirotor Mobile Buoy for Persistent Surface and Underwater Exploration,” U.S. Patent 9 457 900 B1, Oct. 4 2016.

[21] MQTT. (n.d.). Eclipse. [Online]. Available: www.mqtt.org. Accessed: Jan. 28, 2017.

[22] L. Zhang. Building Facebook messenger. (2011, Aug. 12). Facebook. [Online]. Available: https://www.facebook.com/notes/facebook-

engineering/building-facebook-messenger/10150259350998920/

[23] Node and MQTT do something on message. (2015, Sep. 12). Stack Overflow. [Online]. Available: http://stackoverflow.com/questions/32538535/node-and-mqtt-do-something-on-message-

[24] Scapy Python library. (n.d.). SecDev. [Online]. Available: http://www.secdev.org/projects/scapy/. Accessed: Jan. 28, 2017.

[25] GPS BU-353-24 datasheet. (n.d.). U.S. Global Sat. [Online]. Available: http://usglobalsat.com/store/download/688/bu353s4_ds.pdf. Accessed: Jan. 28, 2017.

[26] E. S. Raymond. GPSD library how-to. (2015). Catb.org. [Online]. Available: http://catb.org/gpsd/client-howto.html

[27] Raspberry Pi 3 model B datasheet. (n.d.). Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. Accessed: Jan. 2, 2017.

[28] Mosquitto Documentation. (n.d.). Eclipse. [Online]. Available: https://mosquitto.org/documentation/. Accessed: Jan. 29, 2017.

[29] Paho MQTT Python Library. (n.d.). Python. [Online]. Available: https://pypi.python.org/pypi/paho-mqtt/1.1. Accessed: Jan. 28, 2017.

[30] Pi VPN documentation. (n.d.). The Pi VPN Project. [Online]. Available: http://www.pivpn.io. Accessed: Jan. 28, 2017.

[31] Anker Powercore 10000 datasheet. (n.d.). Anker. [Online]. Available: https://www.anker.com/products/A1263011. Accessed: Jan. 28, 2017.

[32] MySQL Server community downloads. (n.d.). Oracle. [Online]. Available: https://dev.mysql.com/downloads/mysql/. Accessed: Jan. 28, 2017.

[33] MySQL Connector/Python developer guide. (n.d.). Oracle. [Online]. Available: https://dev.mysql.com/doc/connector-python/en/. Accessed: Feb. 5, 2017.

[34] K. Lancaster. SIMPLEKML 1.3.0 documentation. (n.d.). SimpleKML. [Online]. Available: http://simplekml.readthedocs.io/en/latest/. Feb. 5, 2017.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California