INTEROPERABILITY ENHANCEMENT AT REMOTE LOCATIONS USING THREAD PROTOCOL WITH UAVS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Sivateja Reddy Vangimalla

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

December 2018

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Mohamed El-Sharkawy, Chair
Department of Engineering and Technology
Dr. Brian King
Department of Engineering and Technology
Dr. Maher Rizkalla
Department of Engineering and Technology

Approved by:

Dr. Brian King

Head of the Graduate Program

To my parents,

Dr. Mallikarjuna Reddy Vangimalla

&

Dr. Prasanthi Lakkireddy

ACKNOWLEDGMENTS

I would like to thank my thesis advisor Dr. Mohamed El-Sharkawy for his enduring motivation, support and suggestions.

To Dr. Brian King and Dr. Maher Rizkalla for being a part of thesis committee and their valuable comments.

To NXP Semiconductor Drone Program and its program lead, Iain Galloway, for providing most advanced technology for this research.

To Sherrie Tucker for her timely assistance in catching up academic deadlines and thesis documentation.

Lastly, I would like to extend my hearty thanks to Parvati R Menon and entire IoT Collaboratory for believing in me and their invaluable support through out my journey.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Vangimalla, Sivateja Reddy. M.S.E.C.E., Purdue University, December 2018. Interoperability Enhancement at Remote Locations using Thread Protocol with UAVs. Major Professor: Mohamed El-Sharkawy.

In 21st century, interoperability in remote locations has always been a matter of contention. Interoperability is very closely related to internet and an efficient process saves a lot of time and money. With the advent of Wireless Sensor Networks (WSN), Native Internet Protocol (NIP) is considered as one of the most pragmatic solutions in market to address interoperability challenges and is gaining more attention in research. However, challenges like reliability, security of data, power consumption, range and maintenance, and accessibility of such internet in remote locations still remain a matter of concern, creating further barriers for interoperability.

This research aims at proposing a viable solution to interoperability issues at remote locations, irrespective of its network or payload size, by integrating more advanced Wireless Sensor Protocols like Thread Protocol with a proposed Over The Air (OTA) file transfer functionality, into UAVs. Furthermore, this study analyzes power consumption, reliability, latency and scope of the proposed system and their applications in health care and industries.

1. INTRODUCTION

Interoperability, the ability and extent to which systems and devices can exchange data and interpret that shared data [1], has become more of a need rather than an option [2]. Efficient interoperability in various fields can help increase accuracy, save time, reduce human interventions and save money [3]. In this era of technological development, the idea of interoperability is very closely related to internet, elements of cloud computing, networking and furthermore, to consolidate heterogeneous communication innovations, both wired and remote.

Among various classic ways to achieve interoperability, Wireless Sensor Networks (WSNs) generated an increasing interest from industrial and research perspectives owing to its low cost. A WSN is a self-configuring small network of spatially distributed autonomous nodes that communicate with other connected nodes in the network using radio signals. A sensor node has as the main components a sensing circuit, a micro-controller with internal memory, a wireless transceiver, and power source. Also, WSNs are characterized by diversity because there are many different proprietary and non-proprietary solutions. This extensive range of technologies had held up new deployments and incorporation with existing sensor networks [4] [5].

However, the present trend is to use the Internet Protocol (IP), a communication protocol to achieve native connectivity between WSNs and the Internet. It basically offers a worldwide standard to reach any sensor from anywhere in the world at any time. This can be a potential solution that address the challenges of interoperability during system integration of different devices from various suppliers. IP applies Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6) between its Data Link Layer (DLL) and Network Layer (NL), hence supports other upper layer protocols that are IP-based. Furthermore, many WSN standards like Thread and Internet Engineering Task Force (IETF) IoT stacks started to adopt IP connectivity [6].

Despite the advancements, interoperability being closely related to internet, the ability to collect data from remote locations and its secure transfer to database or repository is still a challenge. Therefore, the question on how best interoperability be achieved in remote locations with poor internet still remains. Taking into consideration the exchange of data, it has three models: Federated, centralized and Hybrid. Federate model enables controlled sharing among autonomous databases. Centralized model uses a centralized repository which can be accessed by a requester on authentication. Hybrid model utilizes combined approach of both the models [7]. Hence it is clear that for any kind of model either a repository or a database is needed which can be created only when data is collected and uploaded to cloud. In developed and developing regions, due to the availability of Internet, creating a database or repository has never been a concern. However, due to lack of internet in remote locations, data is not being collected from such locations autonomously. This leads to lack of data at regular frequencies, human interventions and time consumption. Moreover, there are certain areas which are not easily accessible by humans but there exists an urgent need to collect data. Hence there is a need to develop a system which could help collect large files or sensor data autonomously in absence of Internet from remote locations without human interventions.

Through this research, a possible solution is being proposed to address this problem by analyzing the various existing WSN, integration of the most advanced WSN protocol (Thread protocol) into UAVs and its implementation to access the best it can be used to address interoperability issues in remote locations. Currently, UAVs are being extensively used to deploy various operations at remote locations and hence even in this research, UAV is considered as the most practical and time saving approach that could be conjugated with advanced protocols to reach remote locations irrespective of any barriers.

2. BACKGROUND STUDY OF EXISTING WIRELESS SENSOR NETWORKS

2.1 Overview of the IEEE 802.15.4 Standard

The Institute of Electrical and Electronic Engineers (IEEE) 802.15.4 is a wireless standard, defining the physical layer (PHY) and digital link layer which consists of the Media Access Control (MAC) and Logical Link Control (LLC) sub-layers. It can operate on most commonly used 2.4 GHz band that consists of 16 available channels and it uses the direct sequence spread spectrum technique which uses offset quadrature phase-shift keying modulation with a bit rate of 250 Kbps. It uses binary phase-shift keying for the modulation, with a bit rate of 20 Kbps and the chip rate of 300 Kbps. There are several possible ways of implementing this standard based on application, which includes beacon-enabled mode and non beacon-enabled mode [3]. The MAC layer is responsible for generation of beacons (if the device is an access point), device security, employing channel access methods and enabling a reliable link between peers and other things. Carrier Sense Multiple Access-Collision Avoidance (CSMA-CA) method of channel access is used. The unsecured mode contains no security measures whatsoever and is therefore not highly recommended. The secure mode has the functions: access control, data confidentiality or encryption, data authenticity or integrity and replay protection or sequential freshness [6].

2.2 6LoWPAN

IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) is an adaption layer that allows sending IPv6 packets within small link layer frames [8]. It has been used to send and receive IPv6 packets over 802.15.4 links. The protocol data unit size of IEEE 802.15.4 is 127 octets while the Maximum Transmission Unit (MTU) size of IPv6 packets is 1280 octets. However, 6LoWPAN can act as an adaption layer between IPv6 network layer and 802.15.4 link layer by fragmenting the IPv6 packets at the sender and reassembling them at the receiver. 6LoWPAN uses a compression mechanism [9] to reduce the IPv6 headers size thus reducing transmission overhead. An efficient and low-overhead mechanism is provided to forward multi-hop packets in mesh networks. Thread Protocol uses IP layer routing and link layer packet forwarding instead of parsing the packets to network layer. The MAC layer short address is used to further reduce the information bits needed to be sent over the air. In this manner, the processing cycles and power consumption are reduced while using the IP based routing protocols [10].

2.3 Constrained Application Protocol

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol, used with constrained nodes and constrained (e.g., low-power, lossy) networks in the IoT. CoAP has a similar but lighter version of Hyper Text Transfer Protocol (HTTP) and has same RESTful principle to run on resources constraint embedded devices [11]. CoAP has a much lower header overhead and parsing complexity than HTTP [6]. CoAP has adopted similar patterns, such as URIs and resource abstraction while featuring a flexible communication with World Wide Web (WWW). However, CoAP uses User Datagram Protocol (UDP) protocol rather than TCP which is considered to have a poor performance in resource constrained devices with respect to energy consumptions. CoAP has an advantage of retransmission mechanism to increase the reliability of the WSN. Confirmable (CON) request messages by parent node gets Acknowledgements (ACK) as response from destination nodes. Retransmission of the CON message is processed multiple times based on the user defined counter, if the ACK is missing and the waiting time expires [11].

3. THREAD PROTOCOL

3.1 Protocol Overview

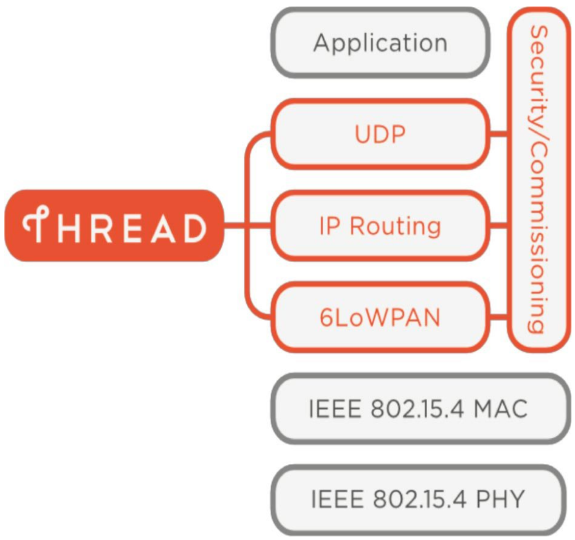

Fig. 3.1. Thread Network Stack [12]

The Thread protocol is an open standard for wireless end-to-end communication launched by the Thread group. The Thread standard is based on IEEE 802.15.4 Media Access Control (MAC) and Physical layer operating at 250 Kbps in the 2.4 GHz band. It is based on low power IPv6 mesh networking technology which can handle up to 250 end nodes with high level of encryption and authentication [13]. It is based on several standards, combining their advantages, featuring reliability, cost-effectiveness, low-power consumption, etc. It is initially designed for building automation and smart home applications where IP-based networking is desired. Thread has no standard application layer, which enables developers to use one of their own, according to the need. It uses User Datagram Protocol (UDP) for the network layer (NWK) on top of IP routing and 6LoWPAN. Despite having a common layer, there are several aspects which are not defined by 6LoWPAN such as routing protocol and its parameters resulting incompatibilities. Thread protocol overcomes this inability and defines all the layers and technologies required to create a common communication network [14].

3.2 Network Architecture

The support of full mesh connectivity among all the routers in the network makes Thread a reliable connectivity. The topology of network depends on type of devices in it. A single router or border router,ca adopt a basic star topology, while multiple router network automatically calls for a mesh topology.Include figure.

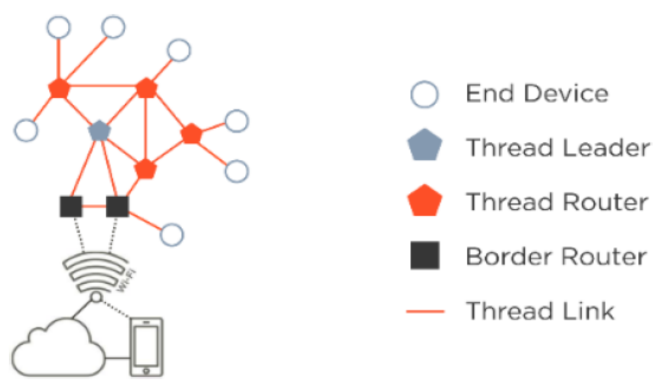

Fig. 3.2. Thread Network Architecture [15]

3.3 Device Types

Border Router

This device forms the link between the enterprise Wi-Fi or Ethernet network, to the energy efficient end devices. Thread supports having multiple Border Routers on the same network that dynamically take over the function, so there's no single point of failure [10].

Router

These devices are used to provide routing services to the network, handle joining and security services for devices trying to join the network. They are also allowed to act as sleepy end devices by downgrading to Router Eligible End Devices. REEDs do not forward messages or provide joining or security services for other devices in the Thread Network. But, the network modifies them as Routers without user interaction, according to the requirement [12].

End Devices

These are the devices on the network that generally only operate ”on-demand”. They don’t re-route data and communicate only through their parent node. They remain in sleep until they are activated and immediately become part of the Thread network upon use [10].

Leader

This is an elected device among available routers to make decisions within the network and manage router ID assignments.

3.4 General Characteristics

Commissioning

Commissioning is a process of adding a new device into the network. This process has two phases, ’Petitioning’ and ’Joining’. A CoAP based mesh commissioning protocol performs maintenance, management, petitioning and relay functions [16]. This process does not transfer any network credentials such as master key. Commissioning requires one device with the Commissioner role, and one device with the Joiner role. The Commissioner is either a Thread device (Direct Commissioning) in an existing Thread network that configures commissioning information directly onto a device using out-of-band method while this information allows the joining device to be a part of proper Thread Network as soon as introduced to the network, or a device external (External Commissioning) to the Thread network (such as a mobile phone) that performs the Commissioner role. Joiner is the device required to join the Thread network, follows the commissioning process [17]. Joining device is assigned a 16-bit short address by the Thread Network. Routers are assigned an address using the high bits in the address field with the lower bits set to 0. Child nodes are assigned a 16-bit short address using their Parents high bits and respective lower bits for their address, allowing other devices in the Thread Network to understand the Childs routing location simply by using the high bits of its address field [18].

Table 3.1.
Thread Short Address

router_id	child_id
u16	

Security

Thread Network uses an elliptic curve variant of J-PAKE (EC-JPAKE), using the NIST P-256 elliptic curve, as the fundamental security, which is a password-authenticated key exchange with juggling. This technique establishes a secure communication between two remote nodes based on the shared key alone, without any Public Key Infrastructure (PKI). A network-wide key is used as elementary form of security at the MAC layer to protect the 802.15.4 MAC data frames, to prevent casual disruption of the Thread Network from outsiders. Network-wide key is not solely used form of protection within the Thread Network as the compromise of any

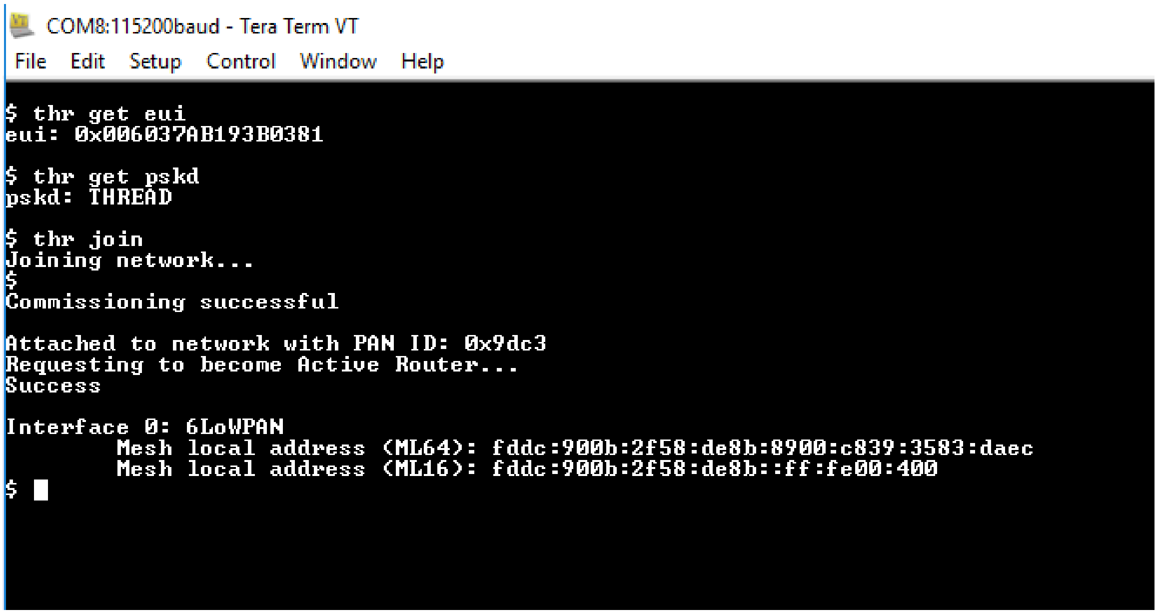

Fig. 3.3. Direct Thread Commissioning

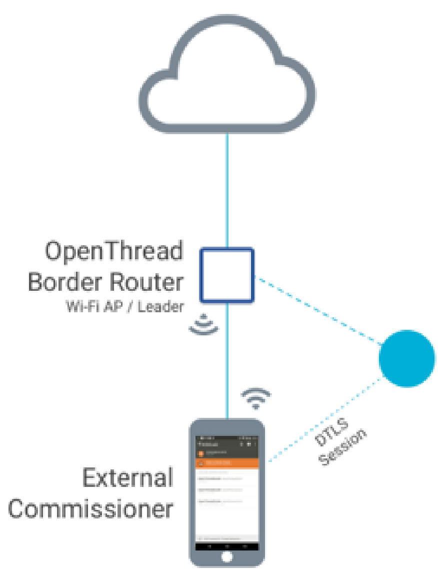

Fig. 3.4. External Thread Commissioning

Thread Device could potentially reveal the key and thus leads to an unsecured network. Therefore, to secure the key, joiner receives other network parameters besides network-wide key using Key Encryption Key (KEK) [16].

No Single Point of Failure

The design of the Thread stack enable devices in the network to perform various special functions and allows them to be replaced without affecting the ongoing communication within the Thread Network. A sleepy end-node communicates with the parent device which is a router or a border router, represents a single point of failure in this scenario. However, the sleepy end-node can find another parent when its parent is unavailable. Thread supports self-healing and does not use coordinator like other networks [19].

Battery Friendly Design

As discussed in the previous sections, Thread is based on the power efficient IEEE 802.15.4 MAC/PHY, where short messaging conserves bandwidth and power consumption. Also, it has reduced network overhead and latency due to streamlined routing protocol [20]. However, Thread uses the asynchronous mode of operation within the 802.15.4 specification, which conserves lot of energy. These devices turn their radios active when required and generate data only during an event. Figure 3.5 depicts the typical wake-sleep cycle with current consumption on Y-axis in milli amperes and time on X-axis in milliseconds [21].

3.5 Over The Air

Over-The-Air (OTA) programming enables a server to distribute new configuration settings, firmwares, etc., to the clients. The asynchronous mode of IEEE 802.15.4 allows parent device to buffer data for the end-nodes to extract while polling. This

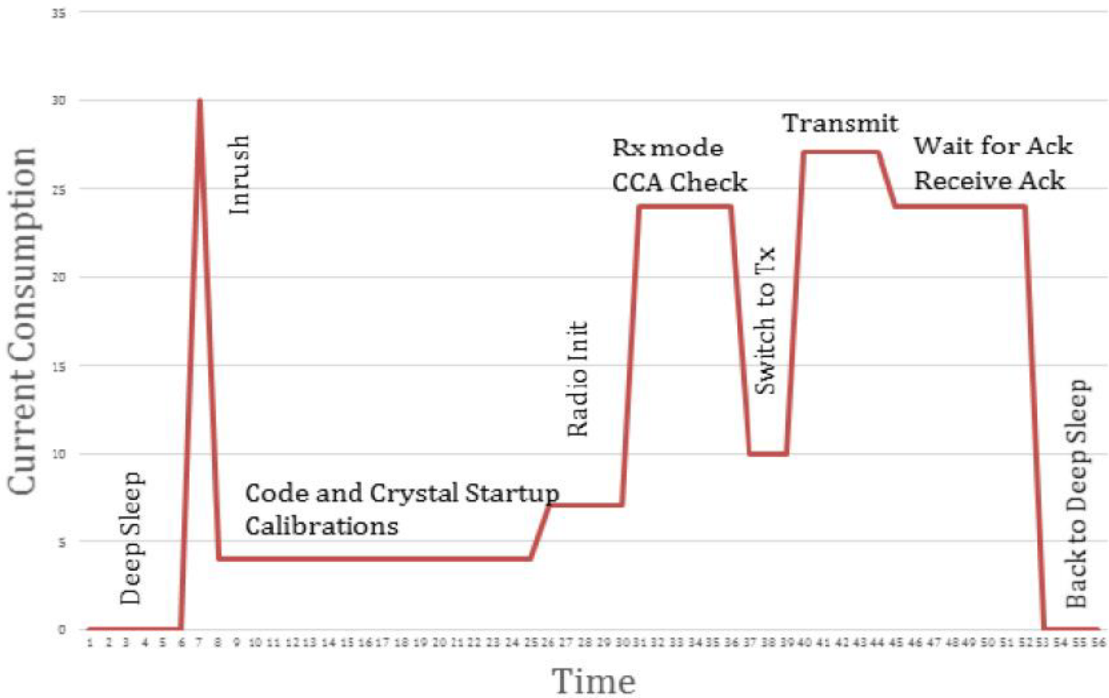

Fig. 3.5. Time Vs Current Consumption of Sleepy End Device [20]

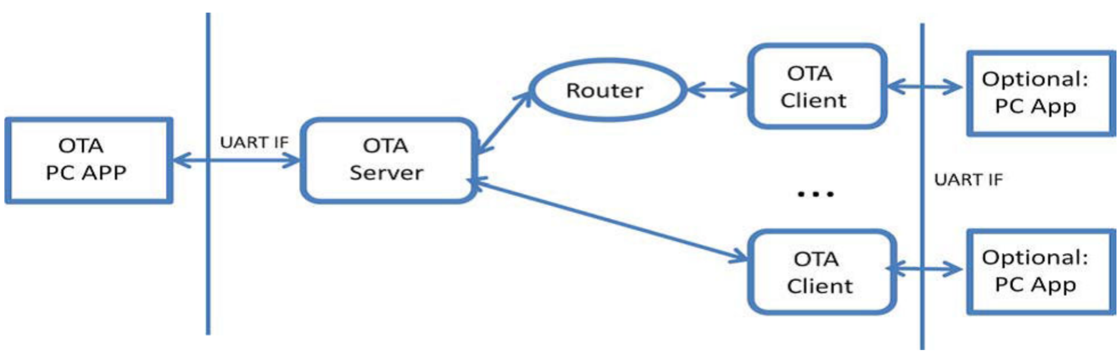

Fig. 3.6. OTA Block Diagram [22]

keeps the OTA message exchange session short, resulting in minimal consumption of power. An asynchronous architecture has the flexibility of adjusting poll frequency, trading-off battery life and down-link latency depending on the applications. A traditional system setup consists of one OTA server and one or more clients, with potential intermediary multi-hop routers, as shown in Figure 3.6. The serial connections with PCs are for displaying status messages and for sending messages to the devices in the network. The OTA client serial connections are optional as the network nodes can also function in autonomous mode [22].

Table include The OTA file format has a header that contains general information

Table 3.2.
OTA File Format

Header	**ImageTag**	**Image**	**Bitmap Tag**	**Bitmap**	**CRC tag**	**CRC**
60 bytes	6 bytes	Variable	6 bytes	32 bytes	6 bytes	4 bytes

about the file such as version, the manufacturer that created it, and the device it is intended for as shown in Table 3.2. Bitmap sub-elements contains information regarding the sectors that need to be erased in the internal flash where setting a bit to 1 indicates the erasure and 0 to protect a particular sector. Other sub-elements in the file may contain upgrade data for the embedded device, certificates, configuration data, log messages, or other manufacturer-specific pieces [22].

4. UNMANNED AERIAL VEHICLES

4.1 Overview

An unmanned aerial vehicle (UAV) is a type of aircraft that operates without a human pilot onboard. Recent technologies have allowed for the development of many different kinds of advanced unmanned aerial vehicles used for various purposes. An unmanned aerial vehicle is also known as a drone.

UAVs had been in use ever since 1840's mostly restricted to military with the earliest being recorded when the Austrians attacked the Italian city of Venice using unmanned balloons that were loaded with explosives. Although balloons would not be considered a UAV today, this was a technology the Austrians had been developing for months before, which led to further advancements. British military, in 1915, used aerial photography in the Battle of Neuve Chapelle. They were able to capture more than 1,500 sky view maps of the German trench fortifications in the region [23].

Today, this technology underwent many advancements and are proving to be an extremely flexible platform for a wide variety of applications.(Picture of advanced UAV) With advances in computation, sensor, communication, and networking technologies, UAV's are finding a place in many civilian applications besides military improving the efficiency of communications range and data aggregation capability. These civilian applications are generally fall into three categories: safety control, scientific research, and commercial applications. Figure 4.1 depicts the classification of UAVs available in market [24].

Till date, some of the key barriers for using UAVs in civilian applications include the cost of acquiring UAVs, building the required applications, and operating the system. UAVs are easy to deploy and are flexible in performing difficult tasks, supporting high-resolution imagery and covering remote areas.

Table 4.1.
Classification of UAVs

UAV Categories	Acronym	Range (km)	Flight Altitude (m)	Endurance (hours)	MTOW (kg)	Currently Flying
Tactical						
Nano	η	< 1	100	< 1	< 0.025	YES
Micro	μ	< 10	250	1	< 5	YES
Mini	Mini	< 10	150–300	< 2	< 30	YES
Close Range	CR	10–30	3000	2–4	150	YES
Short Range	SR	30–70	3000	3–6	200	YES
Medium Range	MR	70–200	5000	6–10	1250	YES
Medium Range Endurance	MRE	> 500	8000	10–18	1250	YES
Low Altitude Deep Penetration	LADP	> 250	50–9000	0.5–1	350	YES
Low Altitude Long Endurance	LALE	> 500	3000	> 24	< 30	YES
Medium Altitude Long Endurance	MALE	> 500	14000	24–48	1500	YES
Strategic						
High Altitude Long Endurance	HALE	> 2000	20000	24–48	12000	YES
Special Purpose						
Unmanned Combat Aerial Vehicle	UCAV	Approx 0.1500	10000	Approx. 2	10000	YES
Lethal	LETH	300	4000	3–4	250	YES
Decoy	DEC	0–500	5000	< 4	250	YES
Stratospheric	STRATO	> 2000	>20000 & < 30000	> 48	TBD	NO
Exo-Stratospheric	EXO	TBD	> 30000	TBD	TBD	NO
Space	SPACE	TBD	TBD	TBD	TBD	NO

4.2 Applications of UAV

Table 4.2.
Various Applications of UAV

Category	Military Application	Civilian Application
Security	Security and control, Aerial Reconnaissance, Aerial Traffic and Security Watch, Battlefield Management	Border Patrol, Port Inspection
Search and Rescue	All-Terrain Search and Rescue, Life Raft Deployment, Rescue Point Marking	Epidemic Emergency and Medical Supply
Monitoring	Waterways and Shipping, Pollution Control and Air Sampling, Chemical, Biological, Radiological and Nuclear Deployments	Forest Fire Surveillance, Traffic Monitoring
Impact and Disaster Management	Impact and Disaster Effects Management, Rescue and Clear-Up Effort Supervision, Disaster Damage Estimation	Disaster Management, Damage Assessment, Hurricane Monitoring
Communications	Secure Telecommunications, Telecom Relay and Signal Coverage	Unsecure Communications
Munitions	Air-to-Ground Missiles, Guided Shells, Anti-Tank Missiles, Air-to-Air Missiles, Wide-Area Munition Deployments	

4.3 Flight Controller

The flight controller is considered to be the brain or nerve center of the UAV. The current generation aerial vehicles are using fly-by-wire, replacing conventional mechanical and hydro-mechanical systems. Also, they are likely to migrate to fly-by-light method in future, which has fiber optic infrastructure [25]. There are various controllers varying from radio controlled to GPS enabled Autopilot flown via two way telemetry links to basic stabilization systems [26]. Pixhawk is a most popular independent open-hardware project that aims to provide the standard for readily-available, hiqh-quality and low-cost autopilot hardware designs for the academic, hobby and developer communities. Pixhawk supports multiple flight stacks like PX4 and ArduPilot [27]. PX4 is one of the leading flight control platforms that uses NuttX as the primary RTOS to run. PX4 consists of two main layers. They are, the flight stack which is a flight control system and the middleware which is a general robotics layer that can handle any type of autonomous robot through internal/external communications and hardware integration [28].

4.4 MAVLink

Micro Aerial Vehicle Link (MAVLink) is a very lightweight, header-only message marshalling library for micro air vehicles. The overhead per packet in MAVLink 1 has only 8 bytes where MAVLink 2 has 14 bytes, which allows an inter-system or intra-system level routing. MAVLink 2 can accommodate more than 256 message IDs over 16 million packets. Furthermore, it adds support for packet signing authentication, that allows a MAVLink system to verify messages. Tables 4.3 and 4.4 depicts MAVLink 1 and MAVLink 2 packet format respectively.

Table 4.3.
MAVLink 1 Packet Format

STX	LEN	SYS (Sender)	COMP (Sender)	MSG ID	PAYLOAD (0-255 bytes)	CHECKSUM (2 bytes)

Table 4.4.
MAVLink 2 Packet Format

STX	LEN	INC FLAGS	CMP FLAGS	SYS (Sender)	COMP (Sender)	MSG ID	PAYLOAD (0-255 bytes)	CHECKSUM (2 bytes)	SIGNATURE (13 bytes)

This protocol is well suited for limited communication bandwidth applications featuring efficiency, reliability, support for multiple programming languages, capability of upto 255 parallel systems on the network and off-board, on-board communications. It follows the modern hybrid publish-subscribe and point to point design patterns which implies that data streams are published as topics and sub-protocols like mission protocol which supports guaranteed delivery of messages over a lossy link with re-transmission. The low overhead makes this protocol suitable for UDP and UART/radio model transport layers and allows to execute the protocol on micro controllers with an efficient encoding. These properties built a common communication architecture across the Pixhawk system. MAVLink has already been adopted by a number of other systems like pxIMU autopilot, ArduPilotMega autopilot, etc. The MAVLink sentences are generated based on an XML protocol specification file in the MAVLink format. The code generator ensures well formed messages and generates C89-compatible C-code for the message packing and unpacking. This allows fast and safe extensions and changes to the communication protocol and ensures that no implementation errors will occur for new messages [29] [30].

4.5 QGround Control

QGround Control (QGC) is a powerful ground control station (GCS) for UAVs that features ease of use for both beginners and professionals. QGC provides mission planning, full flight control and vehicle setup for flight stacks like PX4 and AdruPilot via MAVLink Protocol. The setup panel as shown in Figure 4.2 allows to configure vehicle first flight and tune the gains. Also, setup blocks are highlighted with red icon if there is a need to calibrate or configure which is very informative and useful for avoiding failures in pre-flight checks. The set of parameters available to configure depends on the firmware running on the FMU. Besides basic setup, QGC has other useful features like firmware upgrade that allows to choose latest or custom firmware and dump into FMU via Universal Serial Bus (USB), Airframe selection that allows to choose particular frame type while it provides a provisioning to calibration of sensors for the respective frame selected as shown in Figure 4.3, Radio Setup allows to map channels of main transmitter sticks to control PWMs and assign Flight Modes, Power

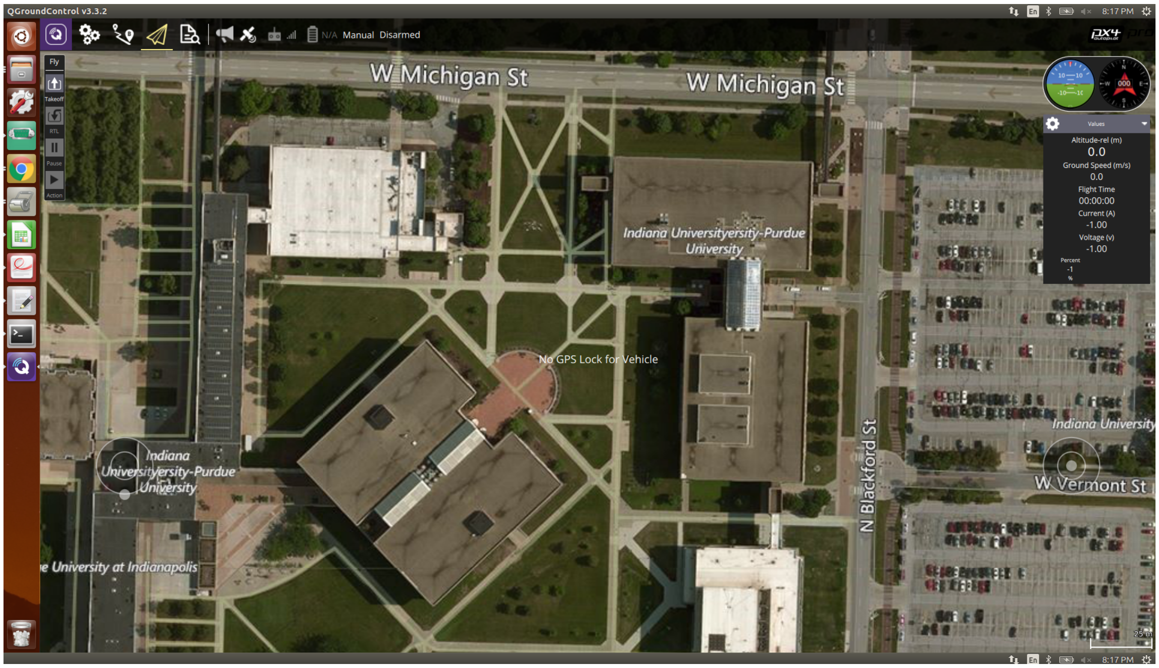

Fig. 4.1. QGround Control - Ground Control Station

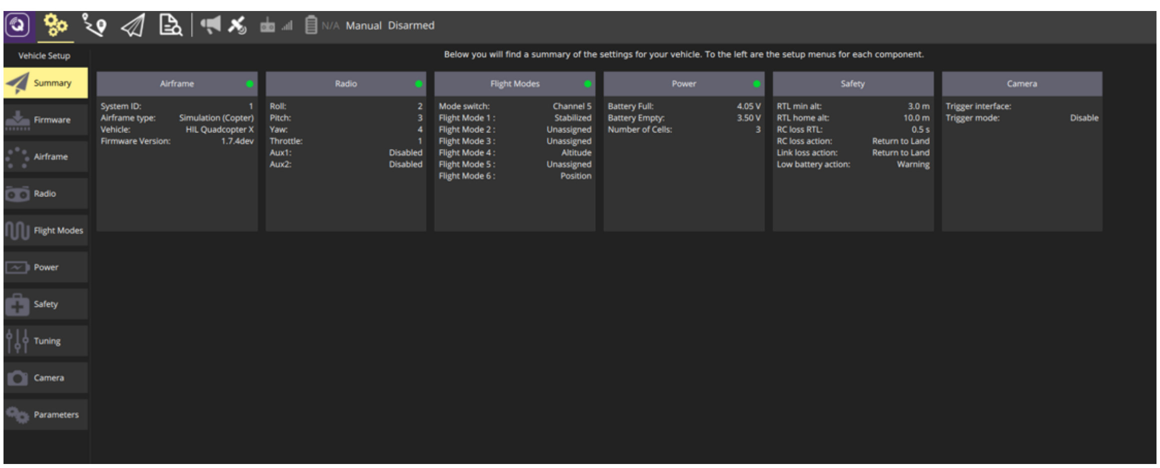

Fig. 4.2. QGround Control - Vehicle Parameters

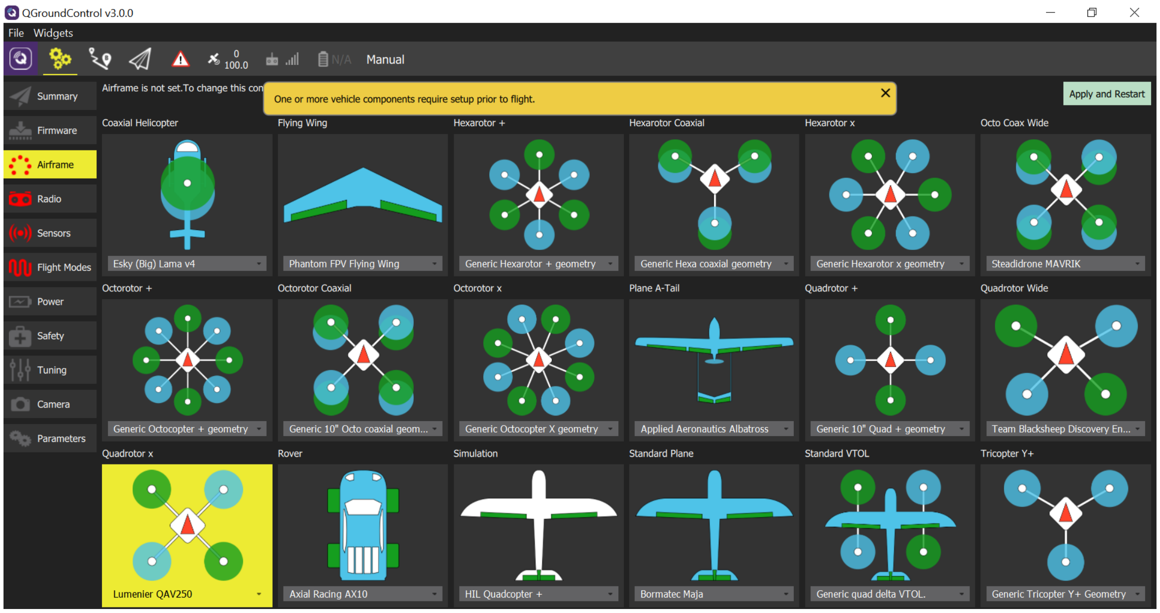

Fig. 4.3. QGround Control - Airframe selection

configurations that provide battery settings and additional power options such as ESC calibration. Furthermore, QGC provides access to system shell via MAVLink Console as it is flexible. Shell provides higher-level access to the system that can be used for basic module running/testing commands, bypass the boot process and perform the specific task and many other development advantages [31].

5. PROPOSED METHOD

5.1 Modified OTA File Transfer

The downloaded OTA image at client, is generally stored in internal or external memory until the bootloader uses it to flash the existing firmware with the new image. Instead of a uploading a firmware from the OTA Server, the proposed modification is to replace it with the required document as a binary format file and the OTA Client stores it in the memory until it is ported to the client computer without passing command to the bootloader for flashing the device, by defining the Bitmap sub-element in the OTA cluster image file.

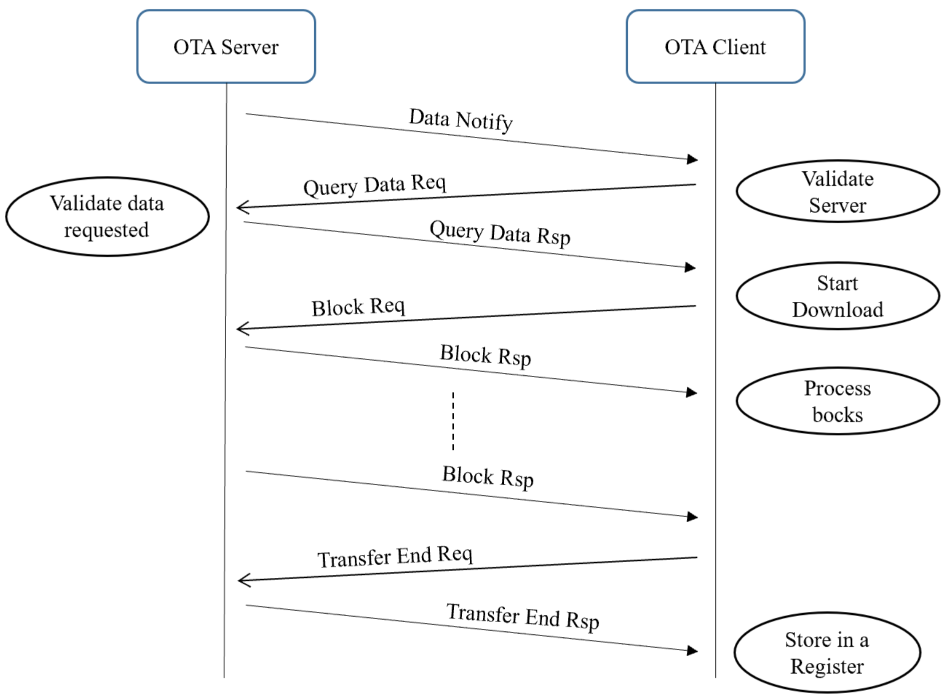

Fig. 5.1. OTA Time Sequence diagram

The Time Sequence Diagram shown in Figure 5.1 explains the proposed OTA transfer for non-firmware binaries. OTA Server looks for clients through multicast discovery process that can look for manufacture specific devices. OTA Client validates the notification and receives data chunks (each chunk can have a maximum size of 1280 bytes which is MTU of IPv6) on request. Post completion of the OTA transfer, OTA Client starts processing the chunks and combine them to retrieve the original file to store in internal or external memory. OTA process can either be in stand-alone

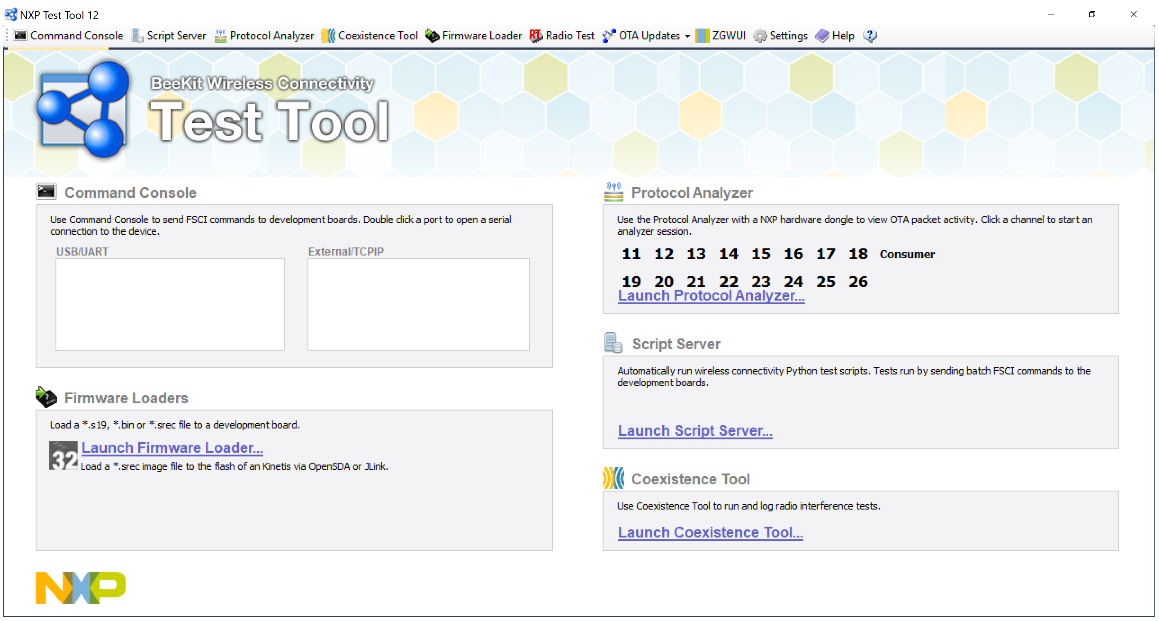

Fig. 5.2. NXP Test Tool

mode which stores the image in server's external flash memory or a Dongle mode which allows Test Tool (NXP Specific) to hold the image and poll each chunk to server. This study uses NXP KW41Z (Cortex M0+) with host control functionality as OTA Server, a Router Eligible End Device as OTA Client and Dongle mode for OTA process that leaves large files on the host PC Test Tool.

5.2 Incorporating Thread Protocol into UAVs

The potential of UAVs in commercial applications has been expanding since half-decade. Developers around the world are exploring this technology for reliable advancements. Though data collection by employing drones is valuable at scale, it has become challenging to many industries to use on-board sensors. UAVs are enabled with wireless networks such as Wi-fi, Bluetooth, etc., that can become a part of sensor network at remote locations to collect data. However, high security and reliability besides low power consumption has been a matter of concern. This study examines the incorporation of Thread Protocol into UAVs for a reliable and secure remote sensor system interoperability and the average power consumption, by performing real-time tests and comparing with other IEEE 802.15.4 based protocol like BLE.

System Design

The proposed test system as shown in Figure 5.3, comprises of Cortex M4 based Flight Controller (Pixhawk) and Thread Radio Modules from NXP (KW41Z). Pixhawk is an autopilot module running PX4 flight stack on NuttX RTOS, which allows to plan autonomous missions to perform sensor data collection at the remote locations or large files using proposed OTA functionality via Thread Radio Module and uses MAVLink for transferring data to base station. Pixhawk is an open-hardware project that is based on 32bit STM32F427 Cortex-M4F core with FPU, 168 MHz, 256 KB RAM, 2 MB Flash and a 32 bit STM32F103 failsafe co-processor. Furthermore, it has inbuilt sensors like ST Micro L3GD20H 16 bit gyroscope, ST Micro LSM303D 14 bit accelerometer / magnetometer, Invensense MPU 6000 3-axis accelerometer/gyroscope, MEAS MS5611 barometer and interfaces like 5x UART (serial ports), 2x CAN, Spektrum DSM / DSM2 / DSM-X, S-BUS, RSSI, I2C, SPI, ADC while one of the UARTs is used to interface with KW41Z in this research.

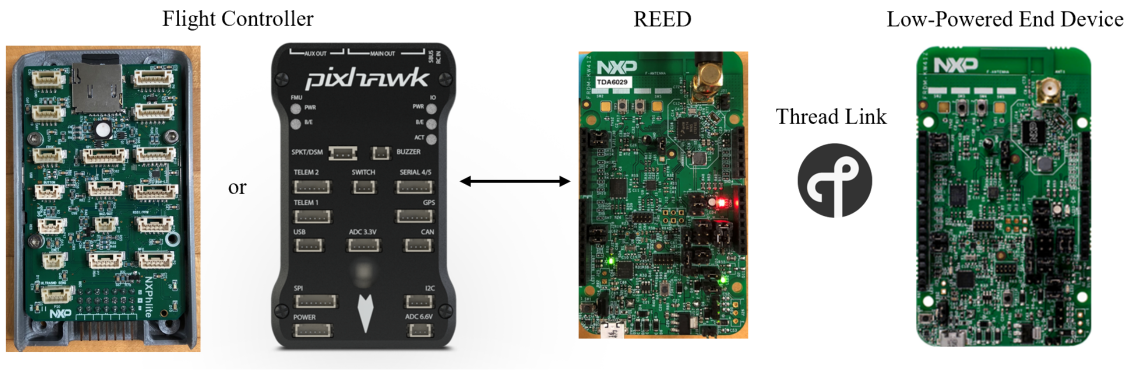

Fig. 5.3. System Design

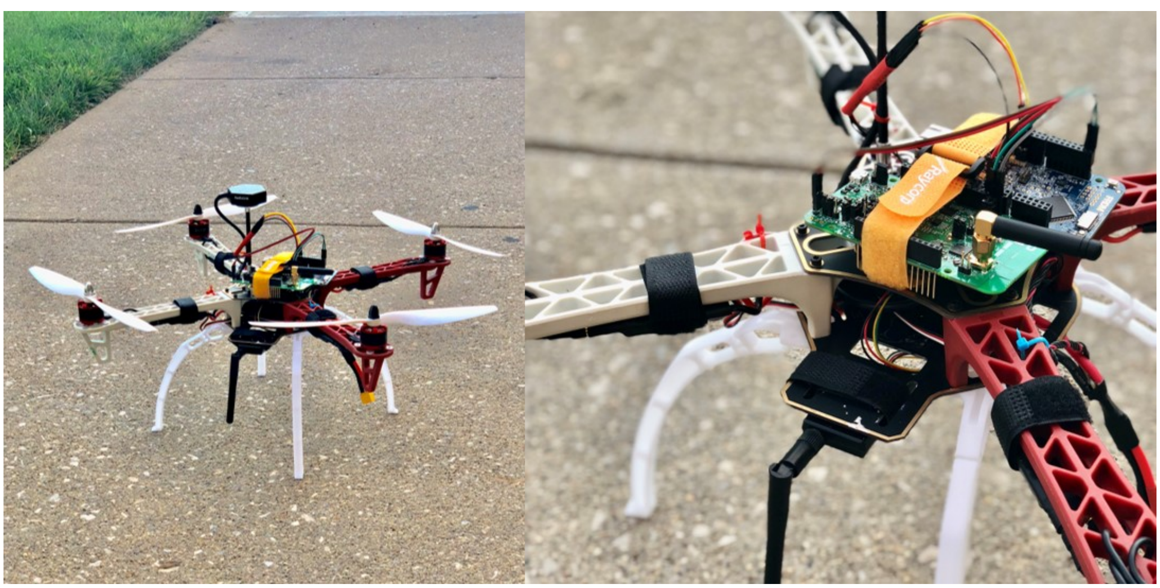

Fig. 5.4. UAV Test System with Pixhawk

Furthermore, this system is tested with NXP's upcoming advanced flight control module, NXPhlite that is based on Kinetis K66 (Cortex M4), @ 2 MB Flash, 180 MHz. NXPhlite runs similar flight software but an advanced version to compensate the harware upgrades. It has automotive grade 2-wire Ethernet 100 Mbps link for optical flow camera or Lidar data, Near Frequency Communication (NFC) besides other interfaces those Pixhawk has, along with Kinetis KV ESCs that can be used to have Field Oriented Control (FOC), interfaced via UAVCAN. Unlike traditional open loop PWM system, this design provides a better control with feedback using sinusoidal waves that can be tuned using NXP Kinetis Motor Suite software. Figures 5.5 and 5.6 shows the industrial drone reference and base platform respectively, by NXP Semiconductors. The industrial reference design depicts the transformation of point-to-point system into a modular based system which is simplified with CAN

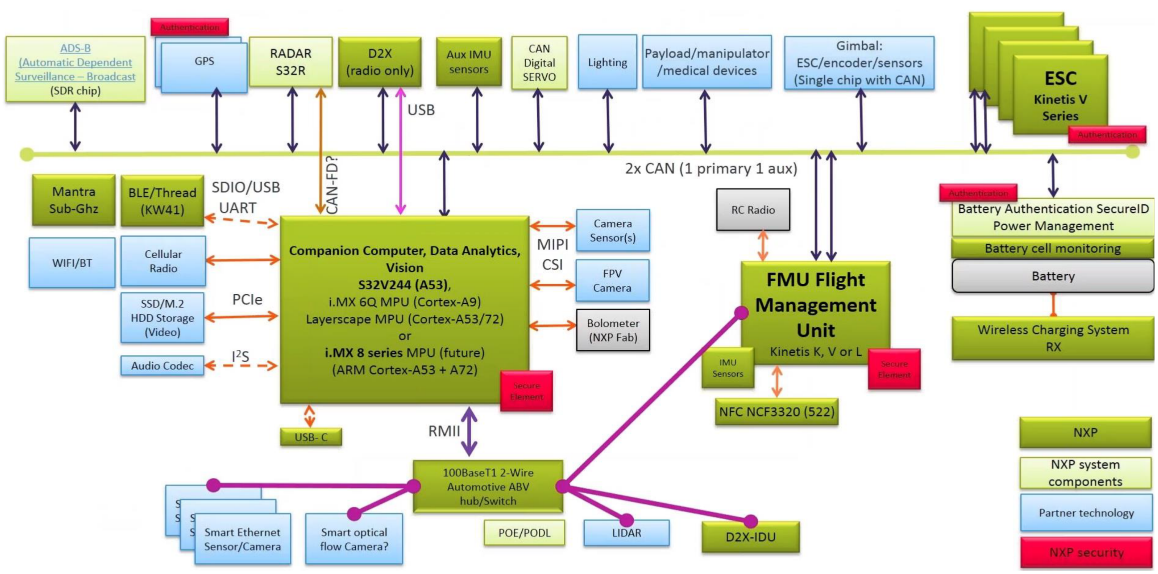

Fig. 5.5. Industrial Drone Reference - NXP UAV Solutions

As explained in previous Chapters, MAVLink is based on the modern hybrid publish-subscribe and point-to-point design pattern, where data streams are published as topics while any parallel task can subscribe to the required topics. This protocol has

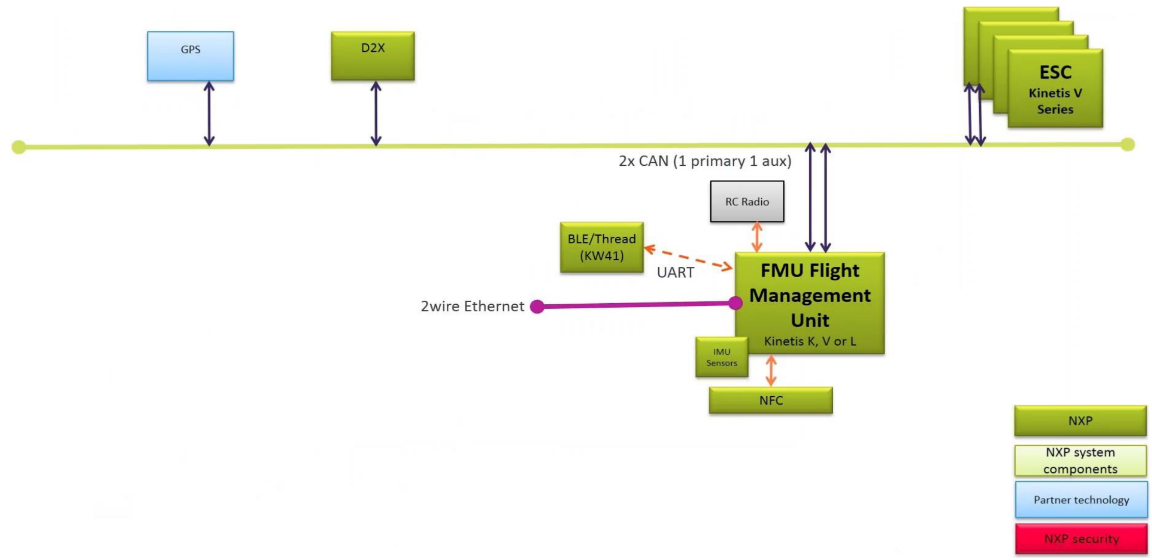

Fig. 5.6. Base Platform - NXP UAV Solution

Fig. 5.7. UAV Test System with NXPhlite

a range of 1000 Meters with regular transceivers while the long range systems can range up to 200 Kilo Meters. A base station comprises of QGC as GCS that supports and receives MAVLink messages besides providing flight control and mission planning.

6. IMPLEMENTATION

6.1 Improvement of Health care at Remote Locations

6.1.1 Overview

In developing countries such as Nigeria and Rwanda, the health care system is of substandard. Most of the areas belong to rural communities that lack facilities like internet, good specialty care hospitals, drinking water, transportation, roadways etc. Rural communities are always at a threat and suffer from many infectious outbreaks year after year. As per 2009 records of Nigerian National Health Conference, health care systems remain weak, with the major reasons being lack of access to care [32]. To change the existing health care scenario, health care transformations are required. In the paper by Dougherty and Conway, it is stated that the health care transformation usually takes place in steps, to create information rich and patient centered health care platform [33]. Such transformations can be achieved by creating an information-rich patient care that includes rural communities.

To obtain data from all domains to help create information rich platform, there must be interoperability which is not taking place due to lack of internet. It is difficult for a patient of rural community to visit a clinic in city due to high expenses and physicians are not willing to come or provide care in rural areas due to lack of transportation and various other amenities. Due to these reasons, neither is the data being collected from such communities, nor is patients' health improving. Evidence has also shown a high mortality rate in such areas too. Hence, tackling this problem will not only help to improve rural health but also contribute to collecting rich data for research purposes, serving for overall the health of the nation.

6.1.2 Evolution of Medical Records

Ever since health care has evolved, it had been a practice to document the patient history, drug doses and prescriptions. The practice of this documentation ranges from paper documentation to digital documentation which is widely practiced now. Over 75% of doctors in the United States use an Electronic Health Record (EHR) to document patient data and for various other purposes. Although the use of EHR is not this widespread in rural areas of various developing countries, evidence has shown that there are many health companies that sprouted up in digitizing health care in developing countries. These companies include One Medical, APMIS, Curacle and Swift Practice EMR which are not costly but efficient for many basic needs and can be found in any small facility [34]. Evidence from a paper by Oyebanjo, shows that the need of IT in rural clinics had already started dating back to 1995 [35].This shows that Information Technology is well embraced in health care. As per the numbers of 2011, 88% of the health care centers were Primary Health Centers distributed across various rural areas in Africa with basic hardware and software adopted in Health IT [36].

Collection of data and its transfer to a tertiary health center, diagnostic/treatment suggestions added from experienced physicians and its transfer back to Primary Health Centre (PHC) can help in providing improved health care to the rural community at low costs compared to a physician coming in or a patient himself visiting a clinic outside his residence area. The collected patient data which can also serve as a rich source of information from areas having increasing number of deaths.

6.1.3 Continuity Care Document

Joint efforts from Health Level 7 (HL7), a standard for exchange, integration, sharing, and retrieval of electronic health information and American Society of Testing and Materials (ASTM), lead to the development of the Continuity Care Document (CCD), an electronic document exchange standard for interoperability of patient information generated from EHR in Extensible Markup Language (XML) format which is human readable and machine interpretable . CCD is built using HL7 Clinical Document Architecture (CDA). This XML file is ready to be exchanged with care providers. This file can contain various kinds of information such as demographics, history, administrative data, medication etc., which on continuous collection can help in prediction analysis of epidemics or most common diseases or risk factors and thereby reducing mortality by being prepared for expected emergencies. Most hospitals in the United states have advanced techniques of machine learning to do this kind of predictive analysis for their patient population using data from their EHR, such that they are prepared to serve the patients instantly. Thus, it is evident that data and predictive analysis play a crucial role in health care. Immunization information of infants in rural areas help governments to monitor the utilization of the provided service and improve it for the betterment of health care. Demographic data can help manage gynecological emergencies. Information of inventory can help manage the inventory better by not having deficient supplies avoiding shortage. Data regarding medications and diagnosis can help identify the efficiency of PHC workers. A comprehensive analysis can relate patterns, quality, costs to timeliness, accuracy and completeness which is highly lacking in health care scenarios in rural areas [37] [38].

6.1.4 Application of Proposed Method

The proposed system is composed of hardware which includes NXP KW41Z, NXPhlite or Pixhawk Flight controller and a PC. There are two nodes, namely Host Controlled Device (HCD) that acts as OTA Server and Router Eligible End Device that acts as OTA Client that communicates serially with Flight Controller.

Host Controlled Device (OTA Server):

The host computer maintains an EHR connected to a KW41Z that is running host controlled interface firmware which enables the host to device control and management API carried over a serial transport over USB and UART and using framing provided by FSCI (Serial Connectivity Interface) transport. The NXP Test tool application is used to communicate with the OTA server via the Thread host control interface (THCI) [19].

Router Eligible End Device (OTA Client):

A Router Eligible End Device is a node which joins the network created by HCD, as an End Device, but can adaptively become a mesh Router, also have capabilities to initialize, create, and bootstrap a new Thread Network for the user or a management entity. The OTA Client receives a new image over the air and writes to the external storage which in our case receives an image in binary file format and dumps it serially, bypassing the firmware update [22].

Flight Controller:

NXPhlite/Pixhawk is an autopilot module that runs PX4 Firmware to drive a quadcopter from base station to remote location which is in turn connected to router eligible end device serially. It uses MAVLink protocol to communicate with QGround Control which is a Ground Control Station (GCS) in this test case [29]. Figure 6.1 summarizes the methodology of the system that is implemented. A base station is considered as any super specialty hospital where the received data can be analyzed. Autonomous unmanned aerial vehicles are deployed from the base station to the desired remote location using GPS and performs OTA file transfer to get the CCD as a binary file with the host computer.

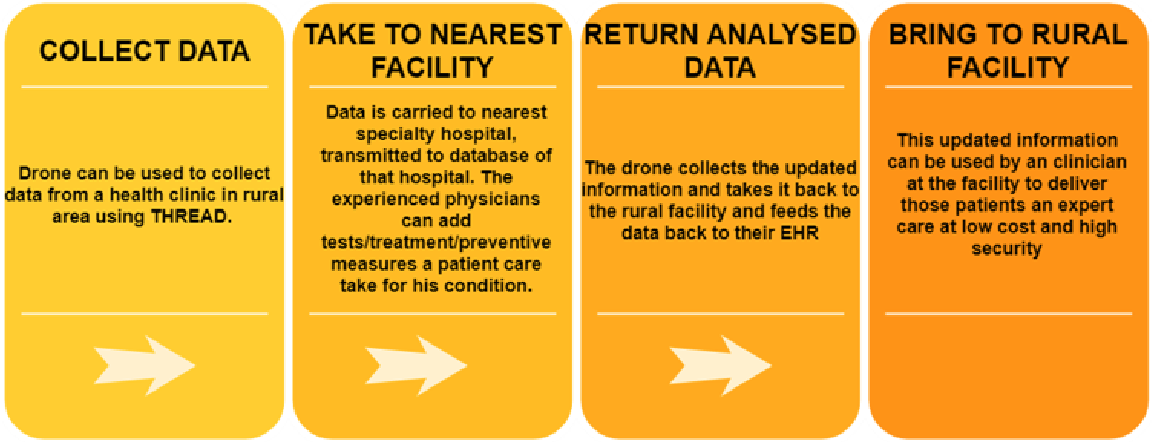

Fig. 6.1. Methodology

Initial Configuration

As mentioned in the Section and shown in Figure 6.2, the test case includes a remote PHC with a personal computer (PC) for a local doctor to manage offline EHR. Also, a HCD over the air server connected serially to the same PC for interoperability using Thread as shown in Figure 6.3. CCD which is a standard for interoperability of patient data from EHR, is usually an XML file which needs to be converted as binary prior to transfer. NXP Test Tool 12 helps in demonstrating the application. Test Tool is a Windows based graphical interface that communicates with various NXP development boards with different functionalities like Command Console, Script Server, Firmware loader, Protocol Analyzer, OTA Thread etc., It works with the Simple MAC (SMAC), the IEEE 802.15.4 MAC, Bluetooth Low Energy, Thread and ZigBee 3.0. Devices under test use this tool to pass commands that control the interfaces between the software protocol layers.

Test Tool is used to create a network as depicted in Figure 6.4, through command console using Thread specific command set. This set includes various other commands like network joining request, start and stop commissioning, adding joiners etc., Once the network is created, it is required to commission a REED OTA Client, that is

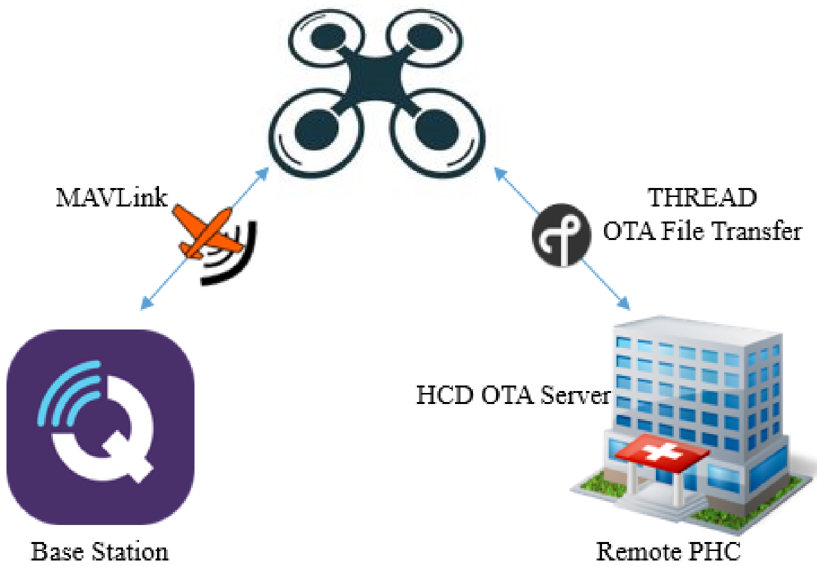

Fig. 6.2. System Implementation

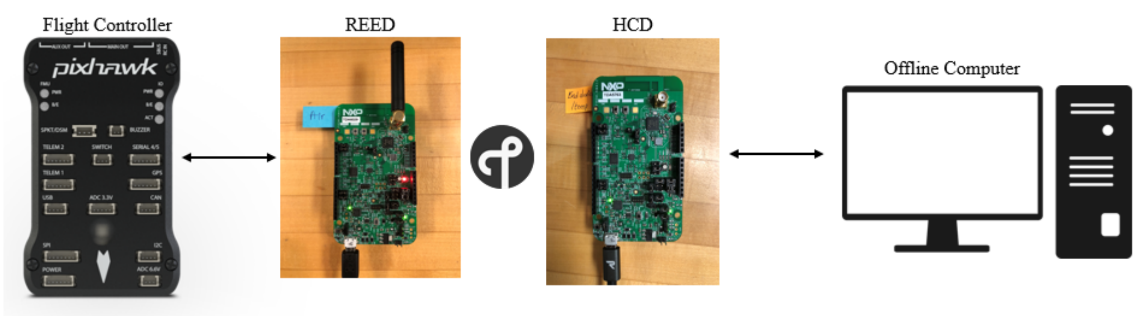

Fig. 6.3. System Interface

serially connected to Pixhawk as mentioned in the previous sections. This device is added to the network as joiner using EUI-64 (64- bit Extended Unique Identifier) and PSKd. Kinetis Thread stack shell interface allows us to get these two attributes using Thread network specific commands like *thr get eui* and *thr get pskd* [39]. After

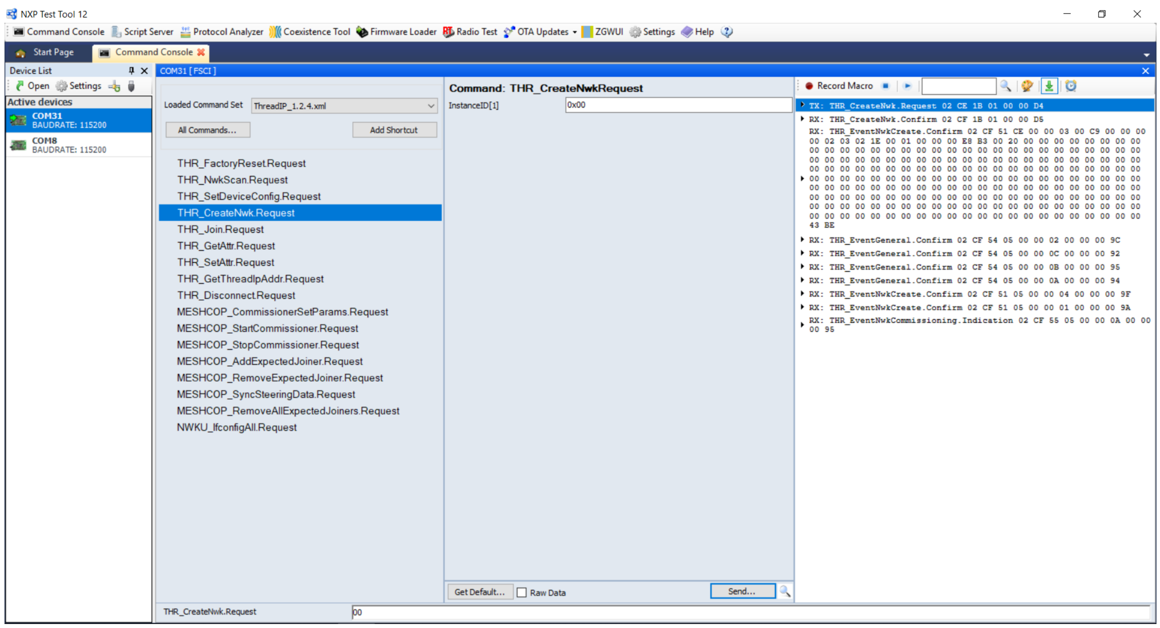

Fig. 6.4. Using NXP Test Tool to create a Thread Network with HCD

the device is added to the expected joiners list as shown in Figure 6.5, it is required to sync steering data from test tool for adding joiner to the bloom filter used for steering new devices onto the network [39]. Now, Thread network command *thr join* can be used at REED-OTA Client, to join the existing network where it is already commissioned as shown in Figure 6.6. Also, the status of joining will be logged on both the ends.

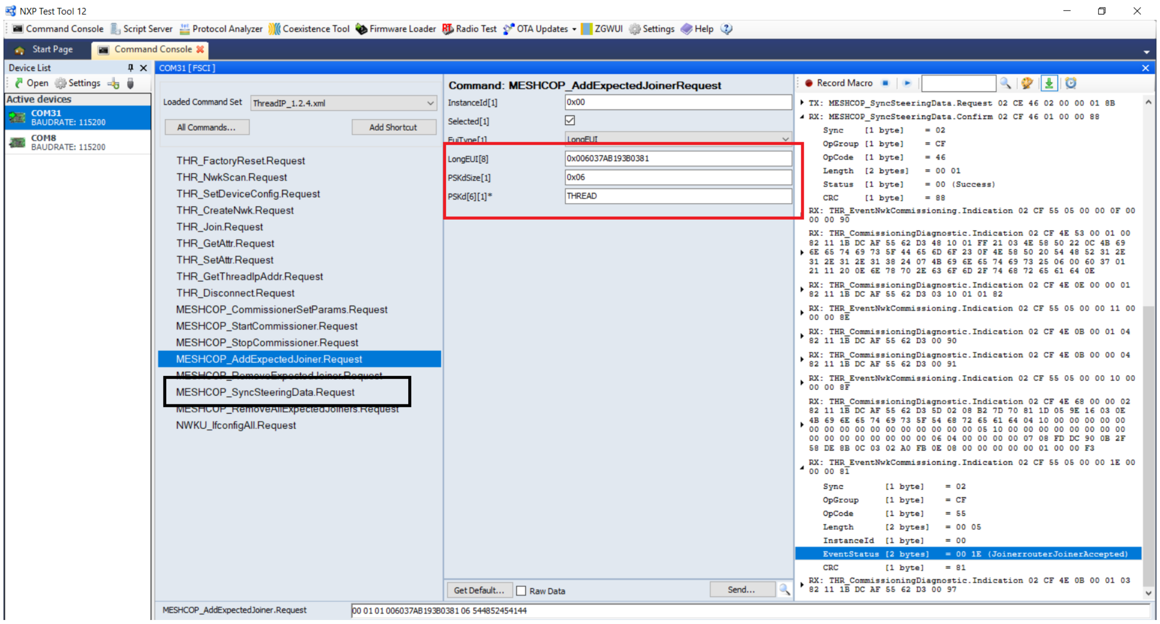

Fig. 6.5. Adding REED credentials to the Joiners list and Steering Network Data

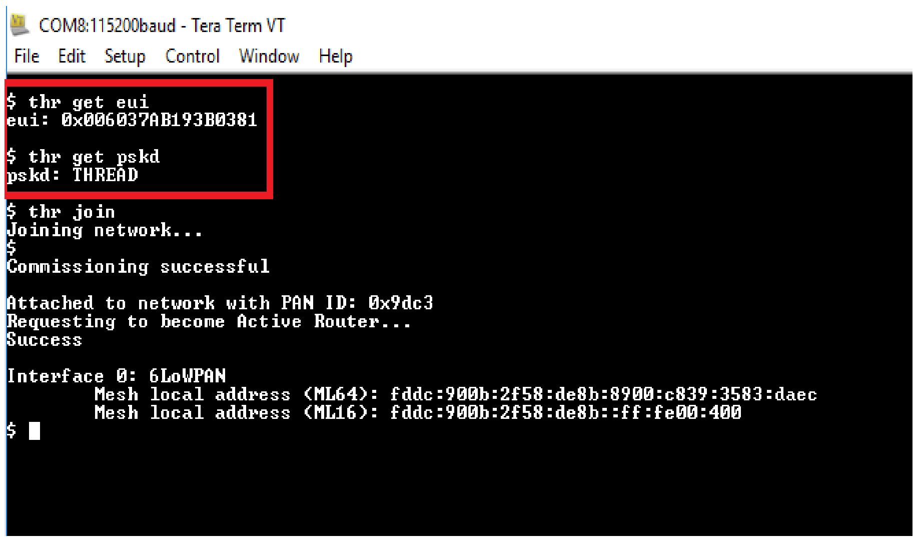

Fig. 6.6. REED Credentials

CCD Transfer OTA

From the above subsection, the drone is commissioned to the network created by a remote PHC. Figure 6.7 shows de-identified sample CCD file of a patient as provided by HL7, is used for demonstration purposes.

```
--->
<ClinicalDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <!--
********************************************************
  CDA Header
********************************************************
    -->
    <realmCode code="US"/>
    <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040"/>
    <!-- US General Header Template -->
    <templateId root="2.16.840.1.113883.10.20.22.1.1"/>
    <!-- *** Note:  The next templateId, code and title will differ depending on what type o
    <!-- conforms to the document specific requirements  -->
    <templateId root="2.16.840.1.113883.10.20.22.1.10"/>
    <id extension="999021" root="2.16.840.1.113883.19"/>
    <code codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" code="11490-0" displayNa
    <title>Discharge Summary (UD)</title>
    <effectiveTime value="20050329171504+0500"/>
    <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
    <languageCode code="en-US"/>
    <setId extension="111199021" root="2.16.840.1.113883.19"/>
    <versionNumber value="1"/>
    <recordTarget>
        <patientRole>
            <id extension="12345" root="2.16.840.1.113883.19"/>
            <!-- Fake ID using HL7 example OID. -->
            <!-- <id extension="111-00-1234" root="2.16.840.1.113883.4.1"/> -->
            <!-- Fake Social Security Number using the actual SSN OID. -->
            <addr use="HP">
                <!-- HP is "primary home" from codeSystem 2.16.840.1.113883.5.1119 -->
                <streetAddressLine>17 Daws Rd.</streetAddressLine>
                <city>Blue Bell</city>
                <state>MA</state>
                <postalCode>02368</postalCode>
                <country>US</country>
                <!-- US is "United States" from ISO 3166-1 Country Codes: 1.0.3166.1 -->
            </addr>
            <telecom value="tel:(781)555-1212" use="HP"/>
            <!-- HP is "primary home" from HL7 AddressUse 2.16.840.1.113883.5.1119 -->
            <patient>
                <name use="L">
                    <!-- L is "Legal" from HL7 EntityNameUse 2.16.840.1.113883.5.45 -->
                    <prefix>Mr.</prefix>
                    <given>Adam</given>
                    <given qualifier="CL">Frankie</given>
```


Fig. 6.7. CCD File in Clinical Data Architecture

For understanding the application, few intermediate steps are assumed to be processed. They are,

- CCD has been converted to binary (file.bin).
- Using QGround Control at base station, the drone approaches the remote location autonomously using a pre-planned mission.

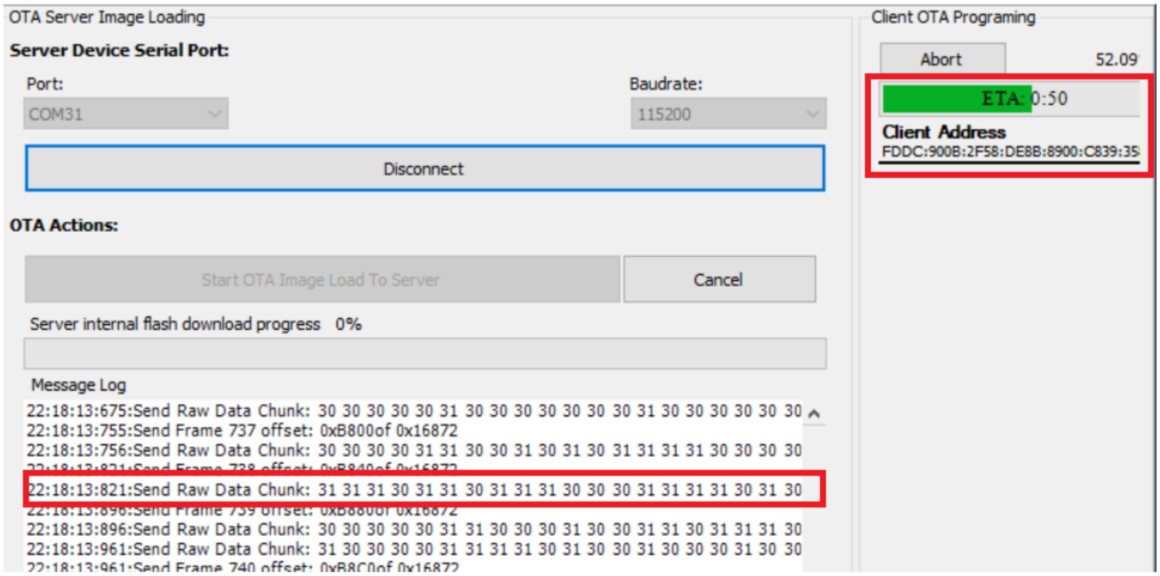

Fig. 6.8. Data transfer from NXP Test Tool

As per the process, connecting to OTA server device via Test Tool initializes the sequence. Path is provided to the file that is ready for transfer. As Thread is IPv6 based protocol, the file has to be fragmented into chunks of 1280 bytes which is the maximum size per packet as shown in the Figure 6.8. These packets are then transferred to the client. Also, test tool provides other information like estimated time left for transmission and the IPv6 address of the client. On the other hand, the drone with OTA Client, has an application running to store this data in a register on board as discussed in previous sections. This data can remain until it is transferred to a database.

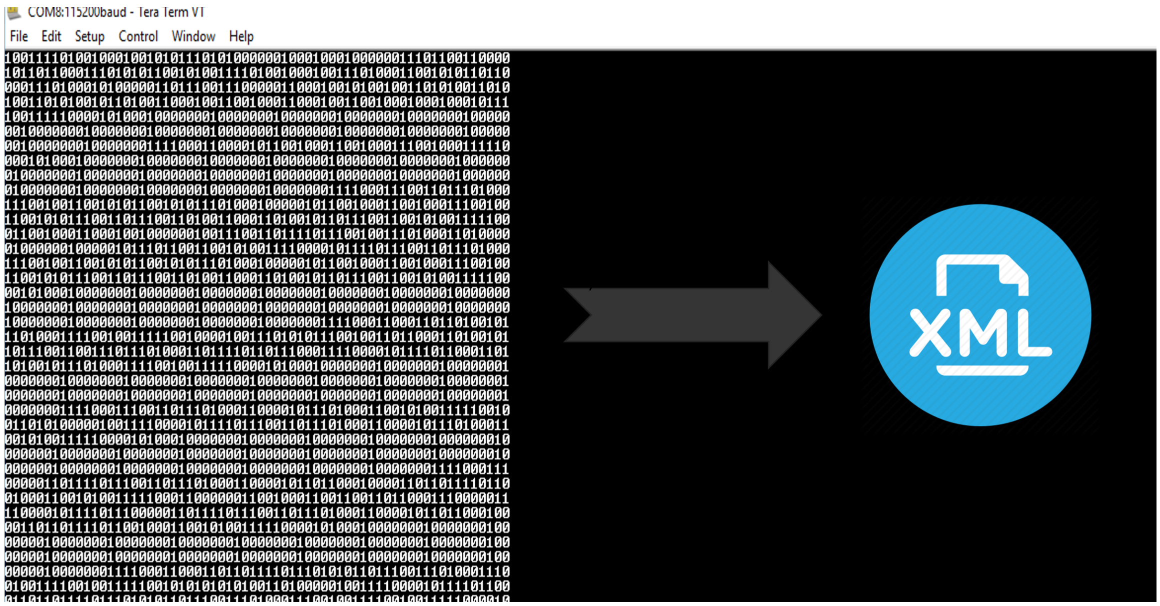

Fig. 6.9. Serial Terminal at REED OTA Client displaying received binary file

6.2 Remote WSN Range Extension:

6.2.1 Overview

As learned earlier, Wireless sensor networks are the most effective means of monitoring and controlling various parameters at both on-site and remote locations. Implementing WSN at remote locations is challenging due to limitations like power, maintenance, reliable gateways etc., and interoperability at such locations have never been easy. UAVs are being employed as a gateway due to their high mobility, for the remote sensor nodes that would use internet through the mobile network. Such implementation results in consequences like high power consumption by the gateway affecting the fight time, data loss due to poor network, reach ability due to lack of mesh connectivity etc., Thread can be used in an efficient way to address such concerns and barriers. Thread is Low-Powered IPv6 based mesh network that allows parent node to communicate with entire sensor network with a single child node in it that avoids the need of establishing communication with each node. UAVs with such advanced wireless protocol can save good amount of flight time, thus reducing the power consumption.

Furthermore, this implementation proposes an efficient way of data collection using MAVLink. It is a common practice to use MAVLink as a mode of communication with UAVs during their flight. It is light weight protocol that publishes vehicle information via radio and has a good range upto 200 KM depending the hardware. Utilizing such radio based protocol to retrieve remote sensor network data or toggle nodes that are connected to actuators, extends the range of the network besides providing control at the base facility with minimal human interventions.

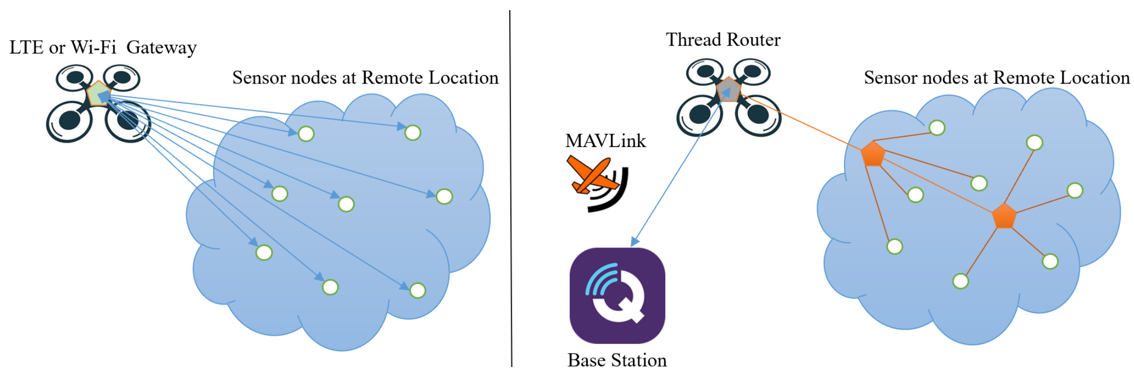

Fig. 6.10. Existing Methodology vs Proposed Methodology

As depicted in Figure 6.10, existing methods for sensor data retrieval using UAVs include limitations such as on board gateway that needs Wi-Fi or LTE connectivity at remote locations for interoperability, communicating with every single node that might reduce the reachability, more power consumption due to multiple way-points that affects UAVs flight time, security, etc. The proposed methodology overcomes above mentioned limitations by employing Thread Mesh Network at the remote locations, with sleepy end nodes and REEDs that consumes extremely low power besides featuring other key features as discussed in previous chapters.

6.2.2 Application of Proposed Method

The proposed system is composed of a Flight Management Unit (FMU) such as NXPhlite or Pixhawk, NXP KW41Z Thread radio module as a Router Eligible End Device (REED) interfaced with FMU. Also, multiple Thread sensor nodes at a remote location that are sensing Temperature, Light, etc., or connected to actuators.

UAVs are deployed from a base facility using QGC as GCS, to a required location using an autonomous mission. After arrival at target location, sensor nodes are requested to respond with the particular sensor data using CoAP transactions. The received information from end devices is transmitted back to GCS via MAVLink as discussed in the previous chapters. Also, end nodes can receive latest firmwares during maintenance Over The Air (OTA).

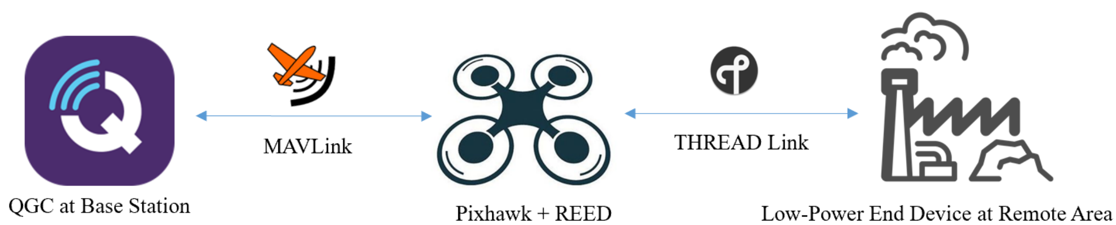

Fig. 6.11. System Implementation

Figure 6.11 depicts the implementation of the proposed system that is designed for extending the range of Thread based WSN at remote locations. The UAV system has a serially communicating Router Eligible End Device (REED) which provides a network for a Low-Power End Device at remote location to respond with sensor information as per the requests through CoAP transactions.

6.2.3 Power Consumption and Reliability Tests

A test case is implemented to evaluate the power consumption of REED, as it is an important concern of UAVs. Power consumption is monitored at the REED and the End Device in a Thread Network, where the CoAP transactions are executed 10 times with in 1 hour, requesting temperature sensor data from sleepy End Device. Besides power consumption, this experiment tests the reliability of the data transfer by monitoring data losses.

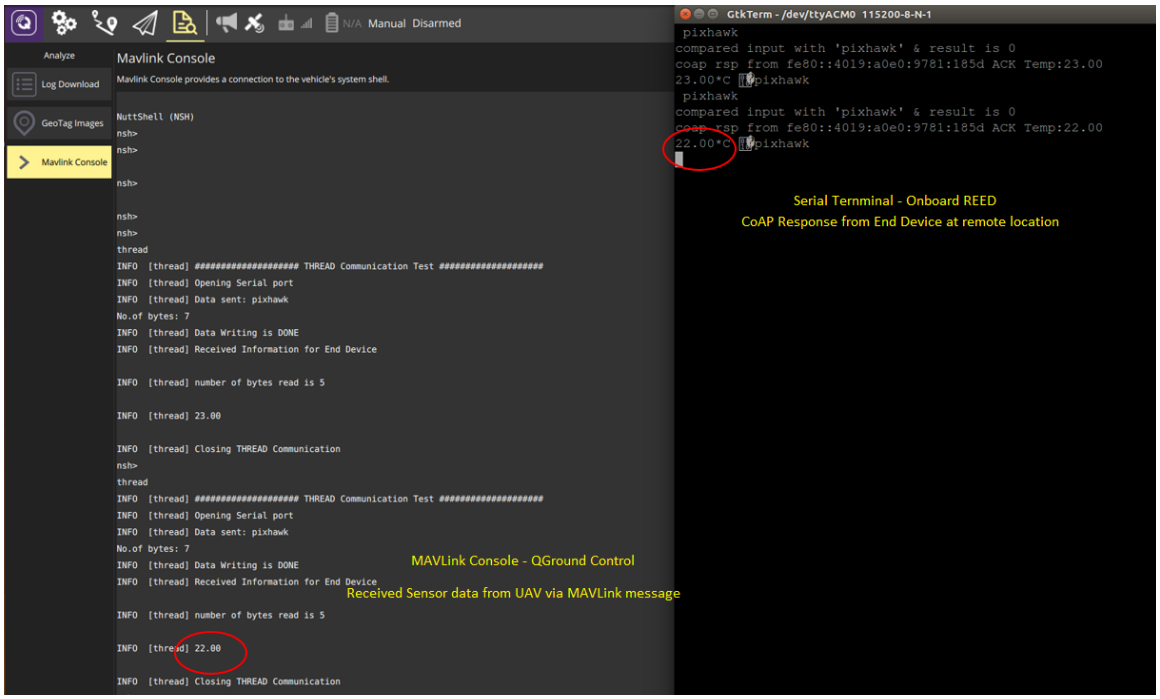

Fig. 6.12. MAVLink Console at GCS - Serial Terminal at REED

Figure 6.12 displays MAVLink Console in QGC, on the left and a serial terminal at UAV on the right. The custom application by the name ”thread” executes a CoAP Request, at predefined location, to the end node with an assigned IPv6 address and a predefined URI. The sleepy end device validates the request, responds back to complete the transaction and return to sleep mode. This response is sent back GCS through a MAVLink message. The Current test as shown in Figure 6.12, is fetching temperature data in Celsius from an end device.

7. RESULTS

Incorporation of Thread Protocol into UAVs is implemented as a part of NXP UAV Reference Design, using latest hardware like NXPhlite, Pixhawk, NXP KW41Z and software platforms like PX4 and tested in real-time environment.

Proposed modification in the Over The Air (OTA) file transfer retrieved non-firmware binaries like the Continuity Care Document (CCD) from Host Control Device (HCD) which is assumed to be at a remote location, without any loss of data. Multiple CCDs were transferred without affecting the system stability. Figure 7.1 shows the performance summary based on size of CCD and OTA latency.

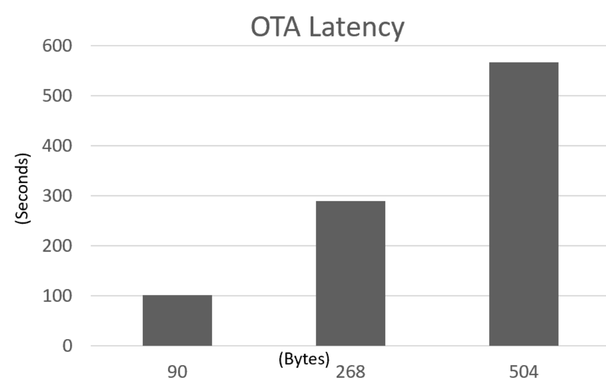

Fig. 7.1. Performance Summary - Latency vs Size

A test case is implemented to estimate the power consumption, latency, data reliability of the proposed system and compared with Bluetooth Low Energy Generic Attribute Protocol (BLE GATT). Average current consumption is monitored at the parent and end nodes of both Thread and BLE single hop network, where the temperature sensor information is acquired from end nodes by parent devices. Furthermore latency and loss of data is monitored to evaluate performance of the network. Figure 7.2 and Figure 7.3 depicts the comparison of current consumption, latency and data

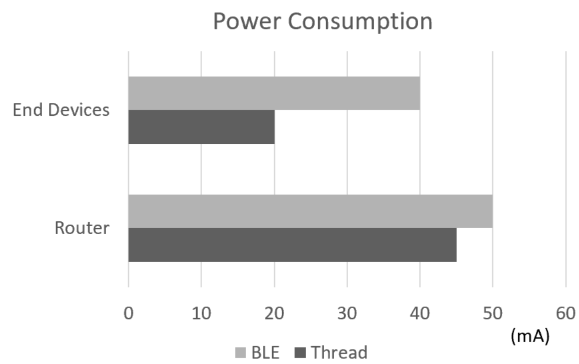

Fig. 7.2. Thread vs BLE - Average Current Consumption

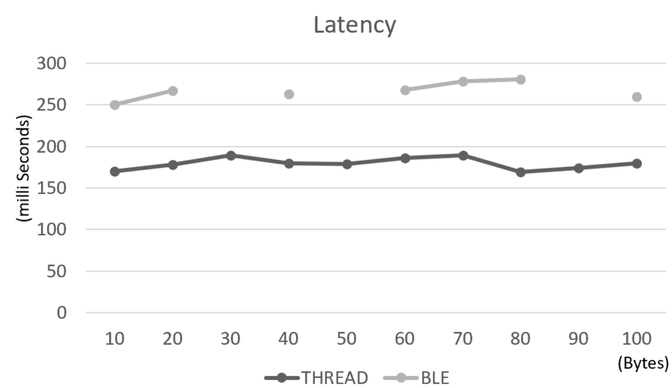

Fig. 7.3. Thread vs BLE - Latency and Data Losses

losses between Thread and BLE. It is observed that Thread has better latency of less than 200 milliseconds with no data losses and average current consumption of the REED and the Sleepy End Device are 45 mA and 20 mA respectively at 5 V.

8. CONCLUSION

We believe that the world is getting smaller and smaller as more people connect over the networked computer systems. But, this is true only in locations where there are resources like internet which makes it possible. It is said that there are only two million people in the world who use internet and that is just nearly one-third the total population. There are many developing countries with remote locations with limited access to resources of technology. Interoperability in such locations has always been overlooked. Data from such locations can be of great use.

Through this research, a system is proposed to address the interoperability issues at remote locations by extending the present scope of using Native IP to achieve connectivity in WSN by using Low-Power IPv6 based wireless network protocol-Thread, at the remote location without using internet. This system is designed to be integrated with UAVs that help to gain easy access to remote locations quickly even without human intervention. It also has advantages of commissioning, no single point of failure, power efficiency and OTA file transfer.

A huge scope of this system was envisioned in healthcare for interoperability enhancement of patient data at remote locations especially in developing countries which are at risk of outbreaks every year. All developed countries have health data interoperability to provide the best care to patient. However, the data from remote locations have always been overlooked owing to difficulty in acquiring them. This deprives such population from their rights of achieving health care. The proposed system through research can be applied in such locations. UAVs with Thread effectively collect data using OTA, thereby pruning the need of internet and transfer this data to desired location. This can help patients have better access to healthcare. This collected data, over a period can also act as a data repository for used predictive analysis of epidemics and big data analysis for innovative solutions in the community leading to health care revolution. This also helps to reduce the use of paper patient records which cannot be stored for a long period and be made available anywhere. The results on testing this system gave an excellent latency with no data loss. The system was stable on transferring large amount of data with no single point of failure with high end security, making it well suitable for healthcare application that involves huge patient record files. This system finds many applications and can have minor modification as per the need.

This, being a novel idea, its scope can be broadened by conducting other research analysis for various domains. However, the proposed schema is believed to have added a new vision for existing opportunities of interoperability at remote locations, efficiently at minimal expense and supervision. This research also made it possible to fulfil one of the visions of NXP of incorporating thread protocol in UAVs.

9. FUTURE WORK

The main focus of this research is to integrate Thread Protocol into UAV design, as a part of NXP Semiconductor's Industrial Drone Reference Design, and investigate the advantages. The results in this thesis provide a strong evidence as the proposed system is a viable solution to interoperability challenges at remote locations. Though the implementation is successful, it has laid a strong foundation for future work. There are few modifications and new integrations required that can improve the base platform in both design and reliability perspectives. As shown in the Figure 9.1, integrating Rapid IoT that supports Control Area Network (CAN), can transform the design modular with respect to connectivity besides improving

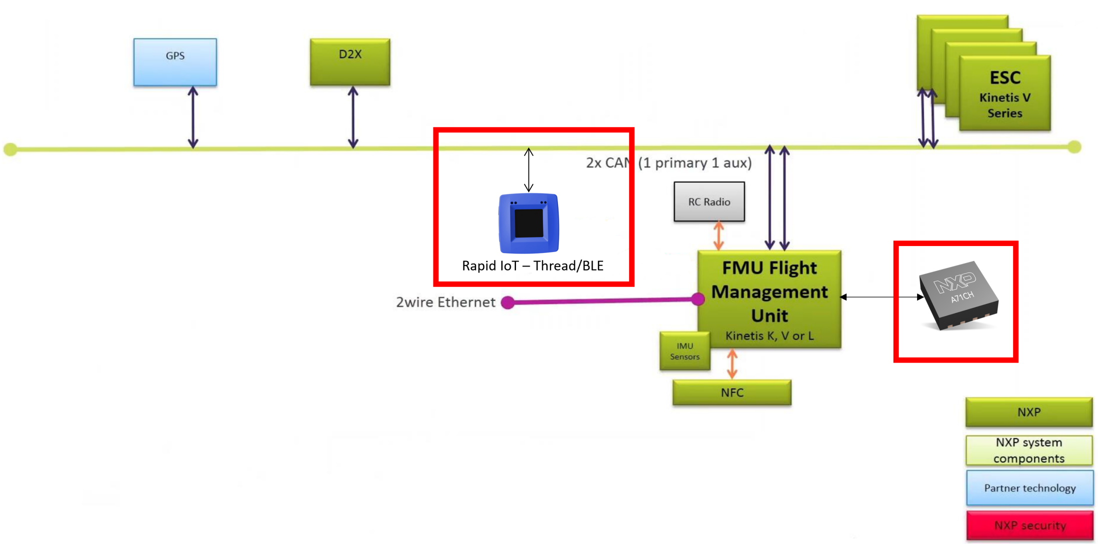

Fig. 9.1. Expected UAV Reference Design

inter-system data transfer rate and reliability and reducing the complexity in terms of communication. Furthermore, incorporating NXP A71CH secures the radio link to the Unmanned Aerial System by pre-injecting keys that are stored on-chip secure non volatile memory, in certified facility for establishing root of trust. Also, it ensures ePassport grade protection, through more than hundred hardware and software countermeasures [40] [41].

This design has other interesting interfaces like automotive grade 2-wire Ethernet (100Base-T1) that can be used for LIDAR or optical flow camera, NFC for secure identification of pilot, that could be researched.

REFERENCES

REFERENCES

[1] HIMSS, "What is interoperability?" October 2018, (Last Accessed: September 23, 2018). [Online]. Available: http://www.himss.org/library/interoperability-standards/what-is

[2] L. Cardoso, F. Marins, F. Portela, M. Santos, A. Abelha, and J. Machado, "The next generation of interoperability agents in healthcare," *International Journal of Environmental Research and Public Health*, vol. 11, no. 5, p. 53495371, 2014.

[3] LIAISON, "Understanding the benefits of edi (electronic data interchanges)," October 2018, (Last Accessed: September 23, 2018). [Online]. Available: https://www.liaison.com/blog/2017/01/26/understanding- benefits-electronic-data-interchanges-edi/

[4] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*, September 2011, pp. 1–6.

[5] C. Rotariu, R. G. Bozomitu, V. Cehan, A. Pasarica, and H. Costin, "A wireless sensor network for remote monitoring of bioimpedance," in *2015 38th International Spring Seminar on Electronics Technology (ISSE)*, May 2015, pp. 487–490.

[6] D. Lan, "Experimental study of thread mesh network for wireless building automation systems," Dissertation, KTH Royal Institute of Technology, Stockholm, 2016.

[7] AHIMA, "Clinical data exchange models," 2008, (Last Accessed: November 12, 2018). [Online]. Available: http://library.ahima.org/doc?oid=77057.WYO-ehKg2w

[8] J. Hui, D. Culler, and S. Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15.4 into the ip architecture," in *IPSO Alliance White Paper 3.*, 2009.

[9] IETF, "Compression format for ipv6 datagrams over IEEE 802.15.4-based networks," (Last Accessed: November 12, 2018). [Online]. Available: https://datatracker.ietf.org/doc/rfc6282/

[10] Thread-Group, "Thread In Commercial Network - Topology Explained," (Last Accessed: November 12, 2018). [Online]. Available: https://www.threadgroup.org/news-events/blog/ID/185/Thread-in-Commercial-Network-Topology-Explained#.W3DbCyhKhPY

[11] C. W. Group, "Constrained application protocol (CoAP) draft-ietf-core-coap-18," (Last Accessed: October 22, 2018). [Online]. Available: https://tools.ietf.org/html/draft-ietf-core-coap-18

[12] L. Zimmermann, N. Mars, M. Schappacher, and A. Sikora, "Development of thread-compatible open source stack," *Journal of Physics: Conference Series*, vol. 870, no. 1, p. 012001, 2017, (Last Accessed: November 29, 2018). [Online]. Available: http://stacks.iop.org/1742-6596/870/i=1/a=012001

[13] Designspark, "11 internet of things (IoT) protocols," (Last Accessed: November 21, 2018). [Online]. Available: https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about

[14] V. Sivateja and M. El-Sharkawy, "Interoperability enhancement in health care at remote locations using thread protocol in uavs," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, October 2018.

[15] O. Oluwaseun and I. Olaronke, "Strategies for managing information flow in nigeria healthcare system," *International Journal of Applied Information Systems*, vol. 9, no. 8, p. 1521, May 2015.

[16] Thread, "Security and commissioning," (Last Accessed: September 23, 2018). [Online]. Available: https://www.threadgroup.org/support#Whitepapers

[17] O. Thread, "Thread commissioning," (Last Accessed: September 23, 2018). [Online]. Available: https://openthread.io/guides/build/commissioning

[18] S. Labs, "Thread fundamentals," (Last Accessed: September 23, 2018). [Online]. Available: https://www.silabs.com/documents/public/user-guides/ug103-11-appdevfundamentals-thread.pdf

[19] Thread, "Thread overview," (Last Accessed: September 29, 2018). [Online]. Available: https://www.threadgroup.org/support#Whitepapers

[20] Google, "Battery friendy design," (Last Accessed: September 29, 2018). [Online]. Available: https://www.threadgroup.org/What-is-Thread

[21] Thread, "The value of low power," (Last Accessed: September 29, 2018). [Online]. Available: https://www.threadgroup.org/support#Whitepapers

[22] NXP, "Kinetis thread stack over-the-air firmware update users guide," (Last Accessed: September 29, 2018). [Online]. Available: https://usermanual.wiki/Document/Kinetis20Thread20Stack20OvertheAir 20OTA20Firmware20Update20Users20Guide.1760681171/html

[23] Consortiq, "A short history of unmanned aerial vehicles," (Last Accessed: September 29, 2018). [Online]. Available: https://consortiq.com/media-centre/blog/short-history-unmanned-aerial- vehicles-uavs

[24] M. Farhan, "Efficient data communication in unmanned aerial vehicles," Thesis.35, United Arab Emirates University, 2015.

[25] C. Tonoy, I. L. Rewzana, and G. Atul, "Evolution of flight control system & fly-by-light control system," in *International Journal of Emerging Technology and Advanced Engineering(IJETAE)*, vol. 3, 2013.

[26] Hackaday, "Droning on: Choosing a flight controller," (Last Accessed: September 29, 2018). [Online]. Available: https://hackaday.com/2014/06/06/droning-on- ight-controller-round-up/

[27] Pixhawk, "What is pixhawk," (Last Accessed: September 29, 2018). [Online]. Available: http://pixhawk.org/

[28] Dronecode, "PX4 architectural overview," (Last Accessed: September 29, 2018). [Online]. Available: https://dev.px4.io/en/concept/architecture.html

[29] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A system for autonomous flight using onboard computer vision," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2992–2997.

[30] Dronecode, "Mavlink developer guide," (Last Accessed: September 29, 2018). [Online]. Available: https://mavlink.io/en/

[31] D. Gagne, "Overview of qgc," (Last Accessed: November 24, 2018). [Online]. Available: https://docs.qgroundcontrol.com/en/

[32] M. Welcome, "The nigerian health care system: Need for integrating adequate medical intelligence and surveillance systems," *Journal of Pharmacy and Bioallied Sciences*, vol. 3, no. 4, p. 470, 2011, (Last Accessed: October 15, 2018).

[33] D. Dougherty and P. Conway, "The 3ts road map to transform us health care: The how of high-quality care," *JAMA*, vol. 299, no. 19, pp. 2319–2321, 2008, (Last Accessed: September 28, 2018). [Online]. Available: http://dx.doi.org/10.1001/jama.299.19.2319

[34] D. H. Nigeria, "Electronic health records (EHR): The death of the case note as we know it?" (Last Accessed: September 30, 2018). [Online]. Available: https://digitalhealth.com.ng/2017/03/09/electronic-health-records-ehr-the-death-of-the-case-note-as-we-know-it/

[35] S. O. Anie, "Impact of information computer technology on primary health care services to rural communities in niger delta region of nigeria," in *Library Philosophy and Practice*, 2011.

[36] WHO, "Primary health care systems (PRIMASYS)," (Last Accessed: September 30, 2018). [Online]. Available: http://www.who.int/alliance-hpsr/projects/alliancehpsr nigeriaprimasys.pdf

[37] ONC, "Implementing consolidated clinical document architecture (C-CDA) for meaningful use stage 2," (Last Accessed: September 30, 2018). [Online]. Available: https://www.healthit.gov/sites/default/files/c-cda and meaningfulusecertification.pdf

[38] TechTarget, "What is continuity of care document (CCD)? - definition from whatis.com," (Last Accessed: September 30, 2018). [Online]. Available: https://searchhealthit.techtarget.com/definition/Continuity-of-Care-Document-CCD

[39] NXP, "Thread stack," (Last Accessed: September 30, 2018). [Online]. Available: https://usermanual.wiki/Document/Kinetis20Thread20Stack20Shell20Interface 20Users20Guide.1222463066.pdf

[40] NXP-Semiconductors, "Nxp rapid iot kit," (Last Accessed: November 24, 2018). [Online]. Available: https://www.nxp.com/support/developer-resources/rapid-prototyping/nxp-rapid-iot-prototyping-kit:IOT-PROTOTYPING

[41] NXP, "A71CH: secure IoT connections," (Last Accessed: November 24, 2018). [Online]. Available: https://www.nxp.com/products/identification-and-security/authentication/plug-and-trust-the-fast-easy-way-to-deploy-secure-iot-connections:A71CH