REMOTE CARGO MONITORING SYSTEM SPECIFICATIONS WITH SMART TECHNOLOGIES

HAMED GHODSINEZHAD

Student number 39307

Master of Science (Technology) Thesis

Supervisor: Sebastien Lafond

Faculty of Science and Engineering

Åbo Akademi University

April 2020

ABSTRACT

Hundreds of containers are lost at sea every year due to poor maintenance and harsh weather conditions and a majority of environmental hazards and economic losses at seas is caused by containers that are lost. In addition to that, securing containers as a stack on ships in rough seas is a procedure that relies on different types of equipment. Thus, reliable technical guidance from vessels' officers is required to install equipment.

Port and shipping companies can significantly benefit from digitalization and new developments to optimize their existing solutions and create new opportunities.

The aim of this thesis is to determine the possibility to design a cargo monitoring system for containership operators and find the specifications based on the literature review. However, there are challenges in ship-to-shore communication, namely, uncertainty about communication methods, technological requirements and the operation cost.

In this thesis, I propose a communication model based on the MQTT protocol to inform ship crew about cargos' status and then define requirements to transfer this information between vessels and onshore bases. This model is divided into two layers: vessel and shore layer. The vessel layer is an internal system that monitors containers' status through sensors' data to assist crew. This information is stored at the ship bridge. The shore layer is to update onshore people about their shipments.

Moreover, the technical requirements in accordance with the maritime environment to implement this model are explored. We estimate the amount of data and required storage along with the approximate cost to send information from ship to shore through satellite internet.

Keywords: Internet of Things protocols, fog cloud, smart container ship

ACKNOWLEDGMENT

To my supportive parents, sister and lovely wife.

Hamed Ghodsinezhad

April-2020

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

OSI	Open Systems Interconnection
ICT	Information Communication Technology
VSAT	Very Small Aperture Terminal
LPWAN	Low-Power Wide-Area Network
PLC	Programmable Logic Control
IoT	Internet of Things
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
M2M	Machine to Machine
MQTT	Message Queue Telemetry Transport
QoS	Quality of Service
CoAP	Constrained Application Protocol
RFID	Radio-Frequency Identification
IIoT	Industrial Internet of Things
LAN	Local Area Network
P2P	Peer-to-Peer
KPI	Keep Performance Indicator
AIS	Automatic Identification System
TEU	Twenty-Foot Equivalent Unit
SYN	Synchronize Sequence Numbers
ACK	Acknowledgment
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
SSH	Secure Shell Protocol
M2M	Machine to Machine
URI	Uniform Resource Identifier
DTLS	Datagram Transport Layer Security

Chapter 1

Introduction

The first successful transatlantic telegraph communication occurred in 1856. It had eight words per minute capacity with a price of 10$ per word. However, in the shipping industry when a ship had left a port it was unable to communicate with the shore until the introduction of radio on ships at the beginning of the 20th century [1].

The early capabilities of ship-to-shore communication were limited to voice or telex for safety and navigational purposes up to the 1990s, when the Global Maritime Distress Safety System (GMDSS) introduced satellite communication and brought digital messaging to robust communication in the shipping industry.

Todays, with the help of satellite broadband systems the possibilities of communication in vessels are increased and, as a result, we now have connected vessels, which are equipped with technological infrastructure such as sensors. This revolution helps mariners to consider connected vessels as a data resource and allows them to have more insight into ship operations.

However, a few underlying questions will need to be answered:

- What type of data should be monitored for cargo security?
- What sorts of skills are required to turn data into actionable information?
- What type of technology should be used with respect to maritime environment?
- What are the hardware and software requirements?

To answer these questions, this thesis tries to define the specifications to design cloud-based software, which would be able to help people in ports and ship offices to monitor cargo status.

1.1. Thesis Structure

The *first chapter* presents the problem and solution requirements from an engineering and user points of view. This introductory chapter shares similar solutions to familiarize with users' need along with vessel communication methods. In the last section of this chapter, a general architecture of communication methods from vessels to shore is presented.

The *second chapter* explains the Open Systems Interconnection (OSI) model and introduces two IoT protocols, MQTT and CoAP. The implementation methods and security of these two protocols are discussed in this chapter.

The *third chapter* focuses on the utilization of cloud computing in the maritime industry and reviews the possibility to use fog computing on a ship. Besides, in this chapter, a maritime cloud model is studied and the specification of a cloud architecture is explained.

A hypothetical architecture model of a cargo monitoring system is presented in the *fourth chapter.*

The *last chapter* describes big data in the shipping sector and data usage from a user perspective. Moreover, this chapter demonstrates technical requirements to setup a monitoring system from the ship bridge and the shore. The amount of data for a specific voyage is roughly calculated and presented in this chapter.

1.2. Problem Description

In the marine industry and especially in cargo shipping, it has been a challenge to improve the security of shipping containers and increase the visibility of cargo transportation. In 2017, the World Shipping Council[1] (WSC) conducted a survey and found that, an average of 1,390 containers have been lost at sea from 2014 to 2017. The WSC is regarded as the best source for accurate information on the subject containers lost at sea. The WSC provides a nine-year period of containers lost at each year, which is presented in *Figure 1.*

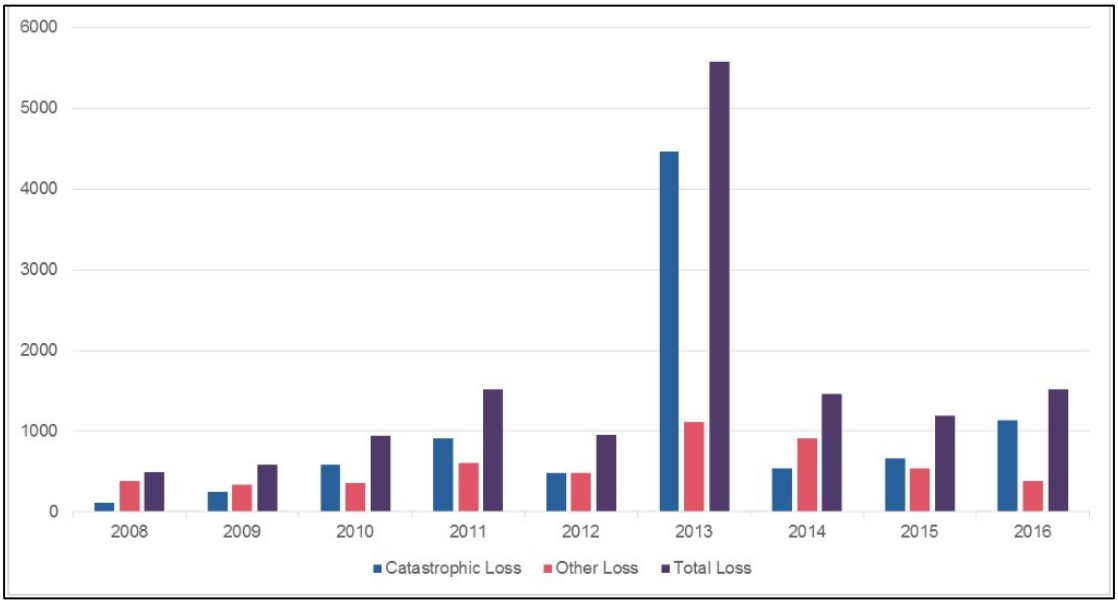


Figure 1 Summary of Containers Lost at Sea [2].

There are fundamental concerns in securing containers and cargos such as:

- Adverse weather conditions need various considerations and due to diverse geographical locations, it might be impossible.

[1] **http://www.worldshipping.org/**

- Lack of cargo lashing information and standards for each situation or fail to be prepared for future problems.
- Inadequate knowledge and time for crew to secure the containers properly before departure from a port.
- Lack of stability in containers/cargos and improper usage of equipment leads to catastrophes.

The essence of the mentioned concerns is absence of intelligence, since securing, tightening and checking processes of lashing are human tasks. In fact, these manual tasks are based on experience from the ship crew.

The importance of this problem is, despite the availability of technological solutions to this need, that ship operators suffer from lack of real-time condition monitoring systems. This gap needs to be studied precisely to determine the feasibility to implement IoT and cloud services based on the shipping environment and explore the shortages for further development. The benefit of this thesis work is not only for ship operators and cargo owners; it is also a basic step toward unmanned shipping and autonomous ships.

The target users are ship operators, who will be using monitoring applications to alert them to take required actions when there is an issue with goods, so that they can improve their safety. This will help them to find the exact root cause of accidents.

Another group who will benefit from advanced Information and Communication Technology (ICT) are cargo owners. Remote applications will give them an opportunity to receive shipment updates and track their goods.

In order to find the needed type of information, which is associated to cargo safety, we conducted three interviews with ship operators and a captain.

1.3. Requirements – Functional and Quality

An adequate number of technologies is required in order to collect data from data points, then transfer data to a central unit and finally process and analyze it. Besides, ship ports should be reviewed to find their current infrastructure to support remote monitoring solutions.

Therefore, this section will briefly explore requirements from technology and maritime perspectives.

1.3.1. Technology Requirements

In this part, technological requirements are categorized into three different aspects and we tried to explain briefly each category as follows.

A. What should be considered from a user point of view to trust and use a smart monitoring application?
    - The application should be convenient to use for all types of users with different knowledge.

- The application should present practical information and it should consider storing and maintaining large amounts of data during the voyage.
- The system should consider bandwidth fluctuation from port to port.
- Due to the nature of business, internet coverage is not guaranteed so that the system must have a minimum dependency to it.
- The system must identify practical data for different user groups. For example, crews need to know about the operation, whereas cargo owners need to know the delivery time.

B. What are the crucial characteristics to implement an IoT system?

- The capacity to connect a large number of heterogeneous elements with high reliability.
- Energy consumption should be considered since an intercontinental sailing can last at least 5 weeks.
- It is important to transmit real-time data with minimum delays.
- The security of a network should be considered wisely, as the system would share sensitive information.
- In the design architecture steps, the ability to configure applications must be thought-out thoroughly.

C. What are data management principles in maritime environment?

- The ability to provide both statistical and analytical information.
- Store data with respect to onboard data storage.
- An application should be scalable since ships have different sizes and accordingly numerous data points.

1.3.2. Maritime Requirements

Vessels

Vessel infrastructure plays an important role when integrating new technologies. New vessels are often designed and built with rudimentary structure to adopt smart technologies faster. A study among approximately 100,000 ships in a global fleet showed that 20,000-30,000 of them, which are already sailing have the basic infrastructure in place and they can justify the investment required to start taking advantage of new technologies [3]. This study showed that 3,000-7,000 ships are expected to consider solid technology infrastructure during construction annually.

Therefore, new vessels are often being designed with a robust technology infrastructure and significant sensoring so that performance and condition data assist the crew to operate and maintain equipment and increase their performance level with lower cost [3].

Ports

Based on the annual number of containers that are loaded and discharged at berth, in Asia, the Shanghai port in China and in Europe, the Rotterdam port in the Netherlands are the largest ones [4].

Following is the specification of these two ports; however, due to the lack of public data it is impossible to provide precise information.

Shanghai[2]:

Shanghai has the highest internet speed among cities in China [5] and the port should benefit from that, as it is located close to Shanghai.

The internet coverage in Shanghai port is built on Wi-Fi technology [6] and very small aperture terminal (VSAT); therefore, a satellite ground station provides accessibility to the internet.

Despite the availability of internet connection, in Shanghai like other Chinese cities, there is a restriction on websites, and consequently, services such as Google and social media such as Facebook are censored and VPN is needed to access. This barrier might affect the speed as well.

Rotterdam[3] :

The Netherlands has almost 87 ports and Rotterdam as the largest one has infrastructure like Shanghai port, Wi-Fi technology and VSAT satellite station. Even though these ports are very much alike, Rotterdam has started a collaboration with IBM on a multi-year digitalization initiative to transform the Rotterdam port operation environment by using Internet of Things and cloud technologies [7] [8] .

The aim of this collaboration is to build a smart port, which enables the entire port of the Rotterdam (42-kilometer) site to host connected ships. The essence of this project is to implement centralized dashboard applications to collect water (hydro), weather (Meteo) sensor data and process the data in real time, then analyze them through an IBM IoT platform. As a result, IBM's cloud-based IoT technologies will analyze the data and turn it into different data models so that the port of Rotterdam can make decisions to reduce waiting times, determine optimal times for ships to dock, load and unload and enable more ships into the available space. For instance, Rotterdam port will be able to predict the best time based on water level, to have a ship arrive and depart Rotterdam, ensuring that the maximum amount of cargo is loaded [7].

1.4. Similar Industrial Solutions

The use of IoT and cloud technologies in the shipping industry is wide and it can cover the entire logistics chain. Following are a few examples of existing applications that are using cutting-edge technologies either for entire supply chains or for specific purposes.

Case 1 - Maersk[4] :

[2] **http://www.portshanghai.com.cn/en/**
[3] **https://www.portofrotterdam.com/en**
[4] **https://www.maersk.com/en/solutions/shipping/refrigerated-cargo/fruit-and-vegetables**

Controlled atmosphere containers from Maersk which are called Starcare, extend the shelf life of fruits and vegetables by slowing down the ripening process during transit.

This solution provides visibility for cold chains and full access to reefer performance data. Thus, customers make decisions regarding reefer cargo and optimize the supply chain based on the data. For instance, transit time for avocados can be extended to as much as 34 days, so a ship can transfer the cargo without damaging them.

Starcare is designed to monitor avocados, bananas and asparagus particularly. Automatic ventilation systems in the container control and maintain the blend of oxygen and carbon dioxide within the reefer unit and goods respire at the suitable rate through the voyage. The basic architecture of the Starcare system is shown in *Figure 2.*

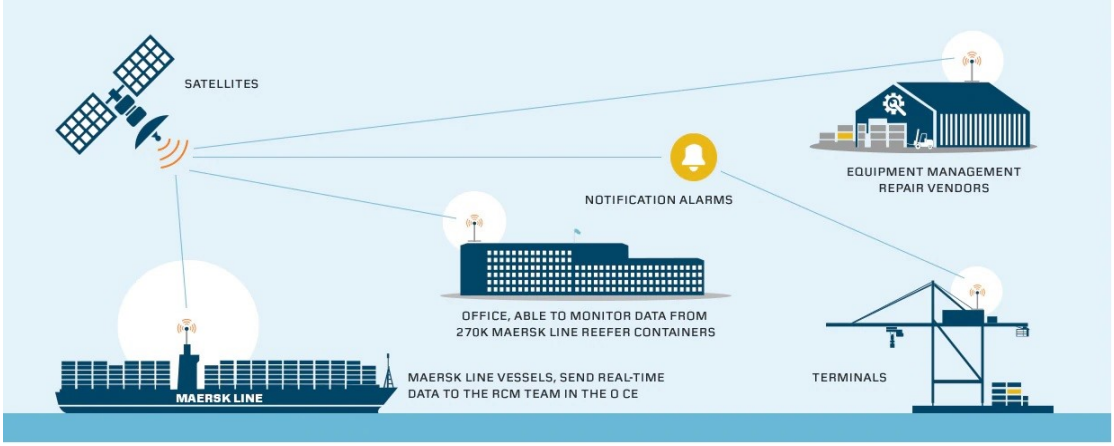


Figure 2 Cold chain transparency with Remote Container Management (RCM) [9].

Case 2 - Bosch[5] :

According to the Food and Agriculture Organization, 1.3 billion metric tons of all food is spoiled before it ever reaches the consumer [10]. This significant loss is mainly due to the quality of goods during transportation, and that is because temperature deviations compromise the quality of food.

Bosch proposed a wireless sensors network solution to control and monitor temperature inside the good packages. Therefore, fruits packs are equipped with sensors. It had been possible to establish remote access to the container via 3G when the container is on the truck or by satellite when the container is out at sea. Therefore, they developed an interface between the internal sensor network and external communication, which is called Freight Supervision Unit (FSU) and placed it on the container.

In addition to that, FSU provides a platform to have insight into goods and their condition. Bosch mentioned that the platform could receive software bundle to assist in decision support. *Figure 3* illustrates the Bosch smart container model.

[5] **https://blog.bosch-si.com/**

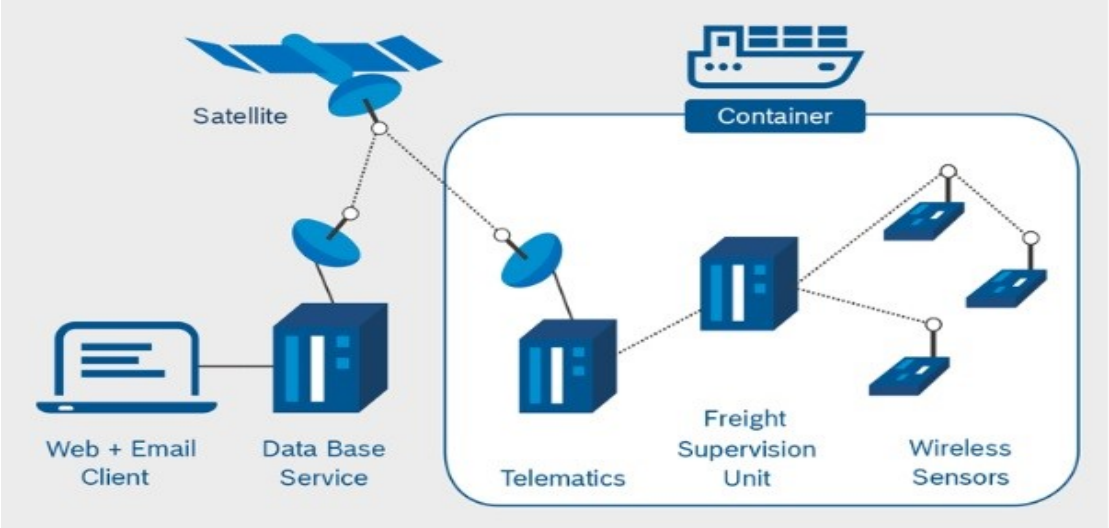


Figure 3 Smart container model [11].

Case 3 - MOST[6] :

Mobile Sensory Technology (MOST) is a monitoring sensor device, which is equipped with five digital sensors to detect location, temperature, light, humidity and shock without installing hardware or software. Then, through GSM network connection, those sensor data will be uploaded to the internet. In fact, there is a web-based interface to read those data which are saved in the cloud and provide monitoring and alert options to check the status of cargo. Finally, REST and PUSH services (JSON SOAP or XML) are integrated as an API to send and receive data. The device power is supplied by a lithium with 100-day battery life.

Case 4 - LPWAN network[7]:

A low-power wide-area network (LPWAN) from Sigfox[8] provides intercontinental coverage so that containers can be tracked. On average, one intercontinental shipment involves more than 200 interactions and more than 20 actors, such as shipping lines, freight-forwarders, in-land transportation companies and port handling.

This leads to difficulties receiving the real-time visibility of the sea-going container shipments and lack of visibility reduces supply chain agility and has a huge effect on services for the final customer. The LPWAN network assists IoT so that sea-freight containers can be tracked in real time. The solution is to place sensors in the container during the loading process and then enable real-time geo-localization from the initial warehouse to the final port of delivery, which includes all transit ports. In addition to that, the solution would be able to detect when the container is unloaded on arrival at the port, if the Sigfox operator's coverage supports the port area. *Figure 4* shows the general network architecture of Sigfox. Since sensors are based on the LPWAN network, there is no need to install software or establish an infrastructure.

[6] **https://most.tech/**
[7] **https://www.sigfox.com/en/news/benefits-tracking-shipping-containers**
[8] **https://www.sigfox.com/en**

Figure 4 Sigfox network architecture [12].

1.5. Ship Data and Communication

In order to understand ship communication and data that are being communicated, in this section, we briefly explain different types of data sources on board a ship and the necessity of them, as well as current communication methods along with their use cases.

1.5.1. Type of Data

Based on container ship operations, data can be divided into two parts. The first part is mainly related to vessel navigation and status and the second part is about cargos status.

For the first part, the vessel data type is categorized into three major sections and each of them along with a few use cases as below [3] [1].

Vessel data type

I. Navigational data
   a. Radar
   b. Gyro Compass
   c. Voyage Data Recorder (VDR)
   d. Electronic Chart Display Information System (ECDIS)
   e. Automatic Identification System (AIS) e.g. position, speed, course etc.
   f. Auto pilot

II. Engine data
   a. Main engine data
   b. Propulsion engine data
   c. Tank and ballast water monitoring
   d. Machinery data e.g. performance and condition monitoring
   e. Fuel consumption

III. None sensor data
   a. Voyage plan

b. Ship flags
c. E-mail services
d. Weather forecast
e. Ships general description

Cargo data type

A variety of measurements needs to be used to secure containers but these are beyond the scope of this thesis. However, based on the analysis of the business process along with an interview that was conducted with a sea captain, basic mandatory data to secure the goods and to have the information about the goods status as follows:

- Humidity
- Temperature
- Vibration
- Pressure

1.5.2. Vessel Communication

The Maritime Transport Program and DNV-GL[9] strategic research and innovation conducted research regarding ship connectivity and the future landscape of maritime technology [1]. This section shortly presents the current available network technology, which makes internet services accessible to the marine industry.

Terrestrial Radio

This communication system is based on medium frequency (MF), high frequency (HF) and very high frequency (VHF). These are well-known radio frequencies in maritime ecosystems for ship-to-shore communication. *Table 1* shows the difference between frequencies band and coverage area [1].

Table 1 Typical coverage of radio systems [11]

System	Typical coverage from earth station
VHF	40-60 nautical[10] miles
MF	150-200 nautical miles
HF	worldwide (given appropriate condition and frequency)

Terrestrial mobile systems

[9] DNV-GL is an international quality assurance and risk management company, which provides classification, technical assurance, software and independent expert advisory services to the maritime industry. (https://www.dnvgl.com/)

[10] A nautical mile is a unit of measurement used in both air and marine navigation, and for the definition of territorial waters. Historically, it was defined as one minute (160) of a degree of latitude. (https://en.m.wikipedia.org/wiki/Nautical_mile)

A communication network, which is distributed over different zones, is called a cellular network. Although the coverage range is shorter than VHF in terrestrial radio, the cellular systems provide useful data connectivity for smaller vessels, such as yachts and fishing vessels [1].

Mobile Satellite System (MSS)

This system refers to the network of satellites' communication intended for use with mobile and portable wireless telephones. The MSS satellite communication market suffers from poor competition, since Inmarsat with 90% of the market share has been the main provider [1] .

Satellite VSAT

A larger type of terminal operating towards geostationary satellite on C-, Ku- and Ka-bands. A Very Small Aperture Terminal (VSAT) access satellite to relay data from small remote Earth stations (terminals) to other terminals in mesh topology or master Earth station in star topology [13].

Figure 5 presents an overview of the frequency bands of different technologies that are mentioned above.

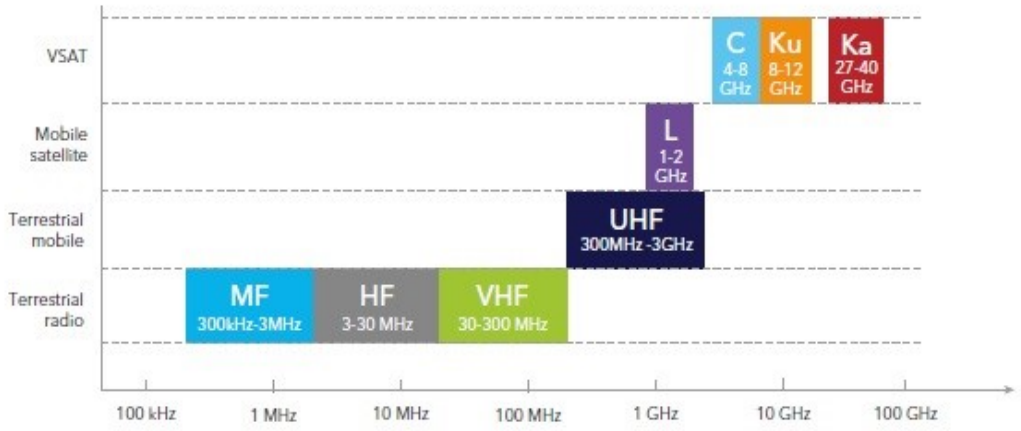


Figure 5 Frequency bands relevant for maritime communications [1].

Common use cases of communication

These methods are being used for navigational aid, as well as reporting to authorities, and they are the most common historic methods and current drivers in maritime communication [1].

- Two-way voice communication (by radio or satellite):

To exchange information with other vessels or shore; for instance, to check weather, to receive updates regarding navigational hazards or route choice and a crucial aid to facilitate navigation.

- Automatic identification system (AIS):

AIS is a messaging system on defined channels in the VHF band, which contains vessel ID, speed, position and course to avoid collision. This messaging system is received by nearby ships and, in recent years, satellite AIS (S-AIS) has been added to enhance coverage.

- Long-range identification and tracking (LRIT):

To report ship position ID to their flag administration[11]. This process is done four times a day through a satellite.

- Vessel traffic service (VTS):

Like air traffic control, this system is designed to monitor marine traffic that is established by harbor or port authorities. Typical VTS systems are based on CCTV, AIS, radar and VHF two-way radio communication. This helps to keep track of vessel movements and assist them with navigational safety in limited geographical areas.

1.5.3. PLC

A programmable logic control or programmable controller is called PLC, and this electrical unit is an important part onboard a ship to capture data, because both navigational data and engine data that are mentioned in vessel data type in *section 1.5.1* are usually managed and controlled by PLCs.

There is a plethora of use cases of PLC in vessels, for instance in the engine room PLC units can be installed as an alarm system, a controlled boiler system, a power management system, a generator control system etc. These controllers are digital computers and primarily used for automation of electromechanical process. PLCs are resistant to vibration and immune to electrical noise [14].

PLCs are being programmed to control the equipment and monitor the process. The main language to program a PLC is C; however, C++ and Pascal are also being used. *Figure 6* shows an ABB plc (AC500); it has 500-megabyte memory with different communication possibilities like Ethernet, Internet, EtherCAT and support OPC UA, MQTT and Modbus protocols [15].

[11] **Flag state administration is the responsible nation and ships comply with rules and regulations. (https://www.marineinsight.com/maritime-law/what-are-flag-states-in-the-shipping-industry-2/)**

Figure 6 AC500 Key features [15].

1.5.4. General Architecture

Figure 7 represents the basic architecture of current communication methods from vessels to shore. Based on current vessel communication technologies, which are discussed *in section 1.4.2*, a vessel can send and receive information with either terrestrial radio/mobile or mobile satellite system or VSAT satellite.

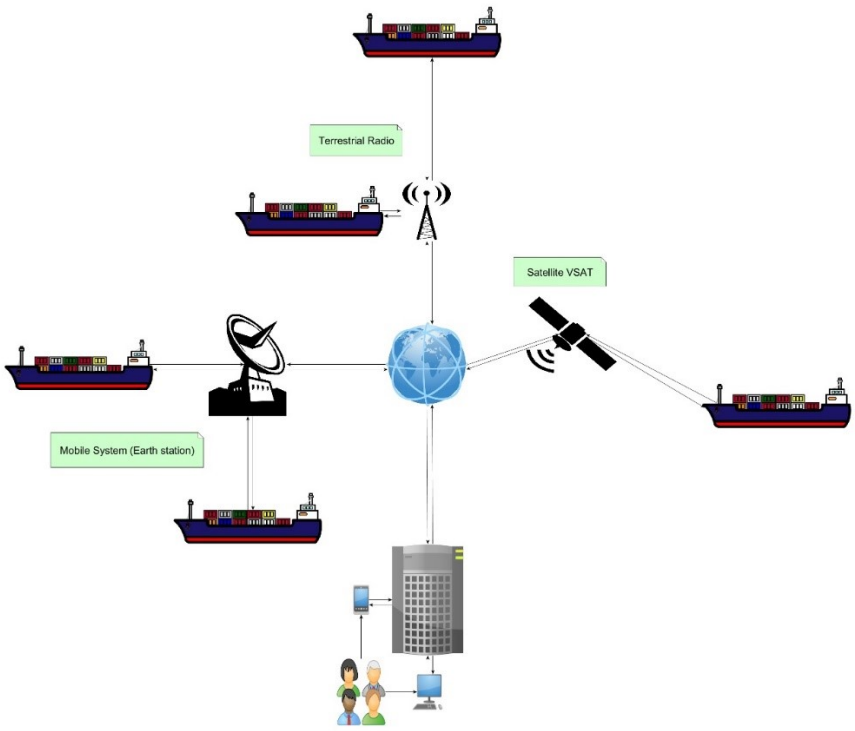


Figure 7 General architecture of container vessels communication.

Chapter 2

Internet of Things

The definition of Internet of Things (IoT) is connecting physical objects to communicate over a secure network without human interaction. Based on IoT Fundamentals Networking Technologies book [16], this term is explained as:

"A world where just about anything you can think of is online and communicating to other things and people in order to enable new services that enhance our life. From self-driving drones delivering your grocery order to sensors in our clothing monitoring your health, the world you know is set to undergo a major technological shift forward. This shift is known collectively as the Internet of Things (IoT)."

However, the maritime industry faces major challenges compared to land-based IoT applications to develop a communication network. The most important barrier is how to create a low-cost and high-speed communication system [17].

Considering the mentioned challenge, the TRI-media Telematics Oceanographic Network (TRITON) project has identified and described a mesh communication infrastructure based on the IEEE 802.16 system that is able to provide high bandwidth and acceptable quality of control (QoS) in narrow water channels [17].

The focus of this chapter is to identify requirements of an IoT application to transfer status data of cargos to the ship bridge by introducing a conceptual framework to understand the relationship between digital devices, internet and their connection methods.

2.1 Standard Communication Model

The Open Systems Interconnection (OSI) is considered as a reference model to show how an application should communicate over a network. The OSI model characterizes a conceptual model to understand communication between digital devices and software in seven layers.

Figure 8 presents this model and its relation to each layer. The top three layers Application, Presentation and Session are grouped together, which simplifies the model.

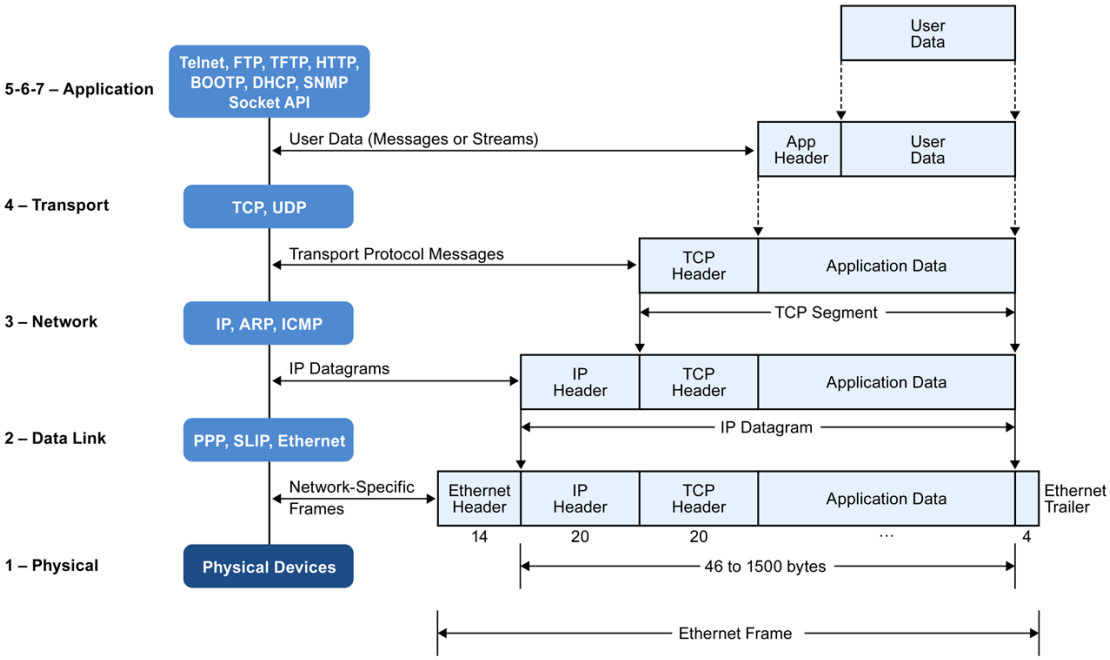


Figure 8 TCP/IP stack reference model [18].

2.1.1 Physical and Data link Layer

Generally, there are three different types of connectivity technologies with respect to the physical layer, namely, short/medium wireless range, cellular and Low-Power Wide-Area Network (LPWAN). These categories are based on the peak data rate in kilobit per second per maximum range in kilometer.

Figure 9 illustrates a theoretical picture of wireless technologies and a comparison of them based on the mentioned category. However, this comparison does not mean that the two aspects (peak data and range) can be obtained at the same time because when the highest data rate is considered the lower communication range is achievable.

In addition to that, different technological parameters are required to implement a network through the wireless technologies. However, due to the vast geographical location, this need in the shipping industry should be more focused to achieve the maximum coverage.

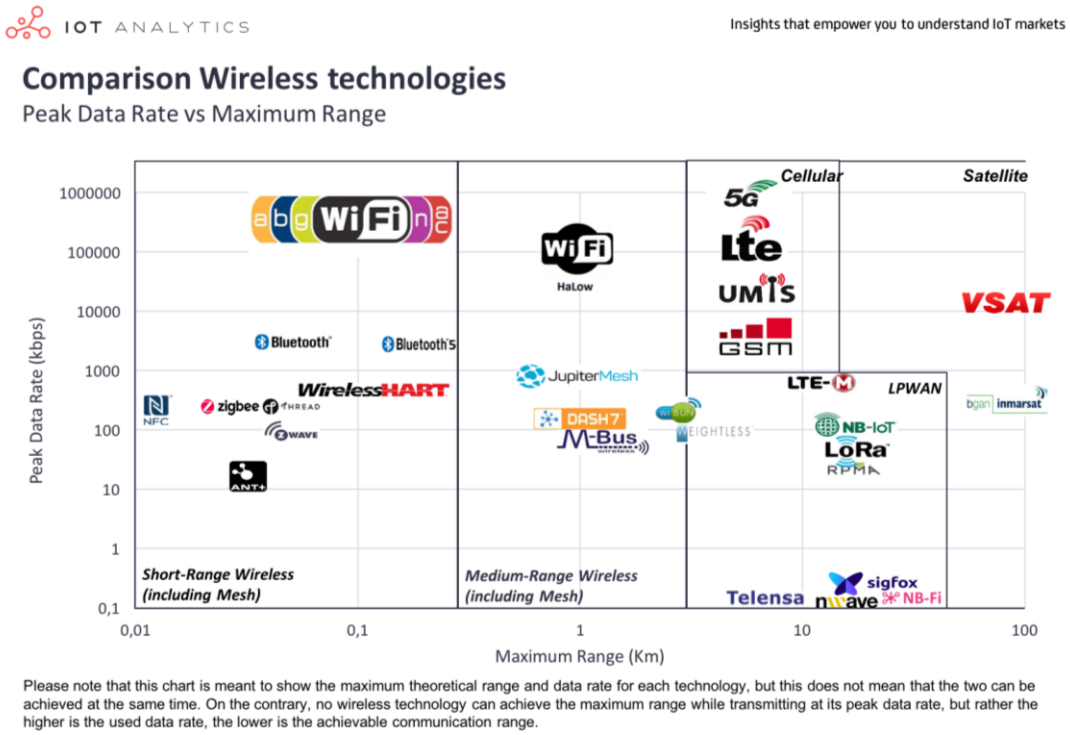


Figure 9 Comparison Wireless technologies [19].

As discussed in *section 1.5.3*, one example of common physical layer usage is a PLC that can be connected to a network through the Ethernet cable.

2.1.2 Network Layer

This layer provides a connection between networks and physical devices to transfer data sequences which is called packet. The data packet is sent from a source to a destination host through the internet protocol address (IP address).

2.1.3 Transport Layer

Most IoT protocols are based on Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). In the application layer, the corresponded protocol to them should be considered and, accordingly, CoAP is designed over the UDP protocol, while MQTT is built on top of TCP. In addition to that, using TCP instead of UDP is theoretically feasible, but not recommended and standardized [20] . A brief explanation of TCP and UDP structures is provided below.

Transmission Control Protocol (TCP)

The TCP protocol is used in the majority of interactive applications with the Web, for instance web browsing and e-mail. TCP also guarantees that the data is received in order and completed, and if it is failed, this protocol will repeat the process to send the data.

First, the TCP protocol acknowledges a session between two hosts, which are trying to communicate. To do that, it will pass through a three-way handshake. *Figure 10* shows the simple form of three-way handshakes. Host A starts a connection to host B by sending Synchronize Sequence Numbers (SYN) and host B realizes that host A is trying to set up a connection. Then host B responds with Acknowledgment (ACK) and SYN. Finally, host A sends the ACK receipt to B and transfers data [21] .

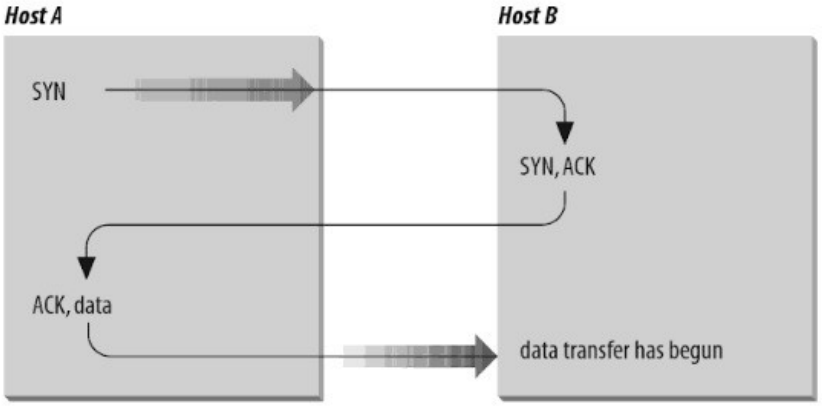


Figure 10 Three-way handshake [21].

After the third phase has taken place, data can be delivered. If a data packet goes astray and does not arrive, then TCP will resend it.

User Datagram Protocol (UDP)

The UDP protocol is called connectionless because it does not establish a session, and either cannot guarantee the delivery. UDP is faster than TCP due to lesser overhead and it is known as a Fire-and-Forget protocol since the originator sends the data and does not need to know anything about the recipient. Besides, Fire-and-Forget is most effective with asynchronous communication channels, as it does not require the sender to wait until the message is delivered to the receiver. Instead, the originator can pursue other tasks as soon as the messaging system has accepted the message [22] .

TCP and UDP Header

Figure 11 shows the header segmentation comparison between TCP and UDP protocols and demonstrates why UDP has a lower overhead compared to TCP.

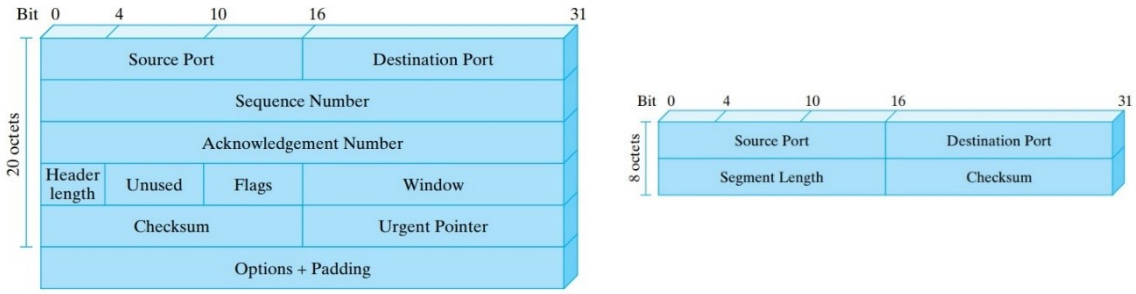


Figure 11 TCP and UDP Headers [23].

The TCP header on the left side with 20 octets or 160 bits[12] provides more fields for flow control and error control, while the UDP header on the right side with 8 octets or 64 bits has fewer fields and, due to this fact, UDP has lower overhead.

In contrast, the TCP and UDP headers contain a 16-bit port number for source (sender) and a 16-bit port number for destination (receiver).

A brief explanation of each segment for both protocols is given in *Table 2.*

Table 2 TCP and UDP header segmentation [23]

TCP segment	UDP segment
• Source port: 16-bit port number of the source	• Source port: 16-bit port number of the source
• Destination port: 16-bit port number of the destination	• Destination port: 16-bit port number of the destination
• Sequence number: 32-bit sn	• Length: 16-bit number representing the length in bytes of the udp datagram (including the header)
• Request number: 32-bit rn	• Checksum: 16-bit checksum used for error detection (later)
• Hdrlen: length of header in 32-bit words, needed because of options, also known as offset field	• Data: the message
• Flags: 6 bits for syn, fin, reset, push, urg, and ack	
• Advertised window: 16-bit number used for flow control (later)	
• Checksum: 16-bit checksum computed over the tcp header, the tcp data, and the pseudoheader (same algorithm as for udp)	
• Urgent pointer: when urg flag is set, urgent pointer indicates where the urgent	

[12] In computer and network technology, an octet represents an 8-bit quantity. Octets range in mathematical value from 0 to 255.

data ends (it starts at the first byte of data)
- Option: variable
- Data: the message

2.1.4 Application Layer

The top layer of Internet Protocol (IP) is where applications create the user data, which are provided by the lower layers. An application layer contains Hypertext Transfer Protocol (HTTP), as the foundation protocol of the World Wide Web (WWW), and in addition to that, covers the higher-level protocol such as File Transfer Protocol (FTP) and Secure Shell Protocol (SSH).

Although, HTTP is widely used as a client-server model, it is not an ideal protocol to build an IoT device due to its special needs, such as security and privacy, scalability, constantly listening to events, pushing information over unreliable networks etc. [24]. There have been various studies on why HTTP is not a proper protocol, for example, a study from IBM [24] revealed crucial facts about HTTP where it was compared to Message Queue Telemetry Transport (MQTT) as a candidate protocol.

The hardware specification for the experiment in [24] was: Android 2.2.2 phone and battery/hour refers to the percentage of the fully charged capacity of a phone battery that is used per hour. The test was done by sending and receiving 1024 messages of 1 byte each.

The result of this experiment was:

- The HTTP protocol consumes more battery in both aspects (battery/message and battery/hour).
- The HTTP protocol is less reliable since it received 240 messages with 3G and 524 Wi-Fi out of totally 1024 messages.
- The HTTP protocol is slower because of a smaller number of messages that were transferred per hour.

Table 3 displays the result and founding.

Table 3 Why HTTP is not enough for the Internet of Things [24]

Characteristics		**3G**		**Wi-Fi**	
		HTTPS	***MQTT***	***HTTPS***	***MQTT***
Receive Messages	*Messages/Hour*	1,708	160,278	3,628	263,314
	Percent Battery/Hour	18.43%	16.13%	3.45%	4.23%
	Percent Battery/Message	0.01709	0.00010	0.00095	0.00002
	Message Received (Note the losses)	240/1024	1024/1024	524/1024	1024/1024
	Messages/Hour	1,926	21,685	5,229	23,184

Send Messages	*Percent Battery/Hour*	18.79%	17.80%	5.44%	3.66%
	Percent Battery/Message	0.00975	0.00082	0.00104	0.00016

2.2 Comparison of IoT Application Protocols

The Internet of Things has become the basis of digital transformation to develop a new business offering and improving the way of working. The maritime industry is not an exception but selecting the appropriate type of IoT protocols is a complex task to start a project.

Before that, it is important to describe two prerequisites and common terms, latency and overhead, which are frequently used to differentiate the advantages and disadvantages of IoT protocols. A simple definition of them is as follow:

Latency

This term is used to determine how fast the content within a network can be transferred from a client to a server and back. This can be measured by the exact time that it takes for a request to travel from a sender to a receiver and the receiver to process the request and send it back to the sender.

Overhead

Overhear is to describe additional or indirect parameters such as bandwidth, memory, computational time or other resources that are certainly required to obtain a precise goal.

2.2.1 Evaluation of IoT protocols

The comparison can be done from different perspectives prior to selecting any protocols, for instance, legacy protocols, different layer methodologies and use cases.

Nevertheless, the focus of this part is on the reliability of IoT protocols and data transmitting speed and to investigate their character, as they are the basis for designing the network infrastructure [25].

IoT data protocols are being used jointly or alternatively to solve different needs of communication among machines, thus, simplicity and low overhead are vital for IoT and Machin-to-Machine (M2M) devices.

The overhead factor in IoT data protocols for WebSocket, CoAP and MQTT has been discussed based on the mathematical model when sending an arbitrary number of data packets to find optimal utilization [20]. This overhead empirical validation was via Wi-Fi network with IPv4-based, a client as Raspberry PI and local laptop as a server.

The result of this test, as it is presented in *Figure 12*, shows that CoAP with non-confirmable requests and responses performs at best by sending up to 100 communication slots (data packets) with average 128 package size and with no packet loss [20].

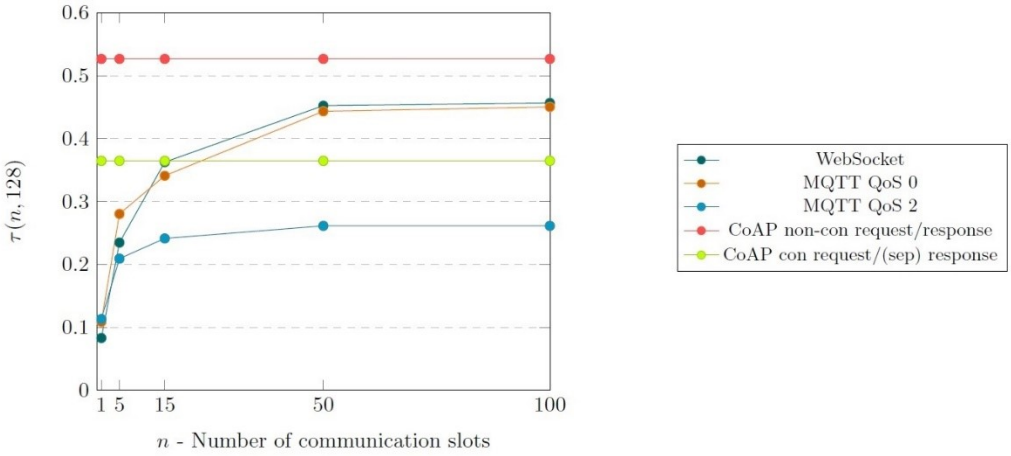


Figure 12 Comparison of IoT Data Protocol Overhead [20].

The next experiment was done with 20% of packet loss and, as illustrated in *Figure 13* yet CoAP with non-confirmable requests/responses scenario is the best throughput and WebSocket and MQTT with QoS 0[13] (quality of service) are sharing the second place.

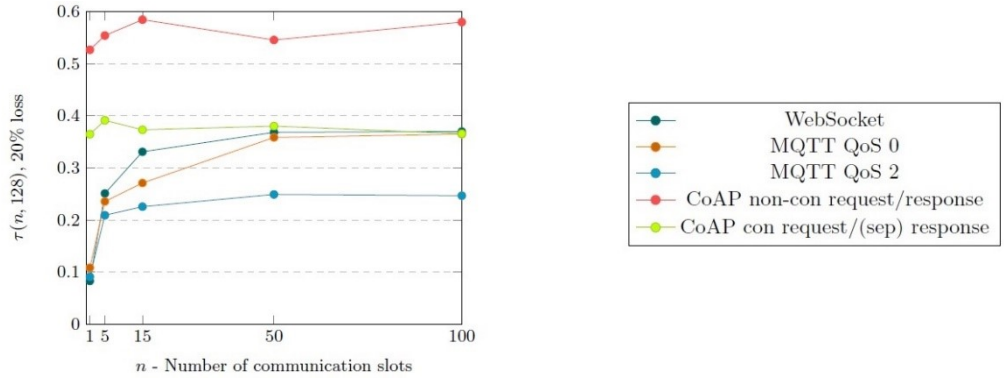


Figure 13 Comparison of IoT Data Protocol Overhead [20].

The characteristics of IoT protocols are varied, thus picking the appropriate one depends on a use case along with the availability of infrastructure in a business and customers' need. In this regard, *Table 4* shows the IoT protocols landscape from Cisco's view.

[13] **QoS 0 where messages are assured to arrive at most once, hence can be lost when connection problems occur. In most cases, QoS 0 is enough since MQTT can take advantage of TCP's connection reliability mechanisms** [20]**.**

Table 4 Beyond MQTT: A Cisco View on IoT Protocols [26]

PROTOCOL	CoAP	XMPP	RESTful HTTP	MQTT
TRANSPORT	UDP	TCP	TCP	TCP
MESSAGING	Request/Response	Publish/Subscribe Request/Response	Request/Response	Publish/Subscribe Request/Response
2G, 3G, 4G SUITABILITY (1000S NODES)	Excellent	Excellent	Excellent	Excellent
LLN SUITABILITY (1000S NODES)	Excellent	Fair	Fair	Fair
COMPUTE RESOURCES	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash	10Ks RAM/Flash
SUCCESS STORIED	Utility Field Area Networks	Remote management of consumer white goods	Smart Energy Profile 2 (premise energy management/home services)	Extending enterprise messaging into IoT applications

A survey was conducted in [27] to acquire an overview of application layer protocols for the Internet of Things and their possible alternative protocols with pros and cons. The key protocols that are being used today to implement the IoT which were considered in this survey are: CoAP, MQTT, XMPP, RESTFUL services, AMQP and WebSocket. This survey indicates that:

- CoAP is the only protocol that runs over the UDP transport layer that makes this protocol the most lightweight.
- If battery consumption and constrained communication has less priority, RESTful service can be implemented.
- As demonstrated in *Table 3*, MQTT has proved to be more efficient for battery-run devices.

Table 5 is showing an overview of the major difference of the aforementioned protocols [27].

Table 5 Major differences among protocols [27]

Protocol	Transport	QoS options	Architecture	Security
CoAP	UDP	Yes	Request/Response	DTLS
MQTT	TCP	Yes	Publish/Subscribe	TLS/SSL
XMPP	TCP	No	Request/Response Publish/Subscribe	TLS/SSL
REST	HTTP	No	Request/Response	HTTPS
AMQP	TCP	Yes	Publish/Subscribe	TLS/SSL
Web socket	TCP	No	Client/Server Publish/Subscribe	TLS/SSL

We discussed the IoT technology requirements in *section 1.3.1* and defined a few necessities, in which the first criterion is battery consumption in the application layer and the second criterion is Quality of Service (QoS) option to have reliable communication and guarantee the delivery.

In addition to that, the overhead factor in IoT data protocols is compared and then protocols are reviewed to find a suitable one for the maritime application. Nevertheless, there are more than a few factors that affect the selection of IoT protocols on the application layer. For instance, battery consumption has a huge influence on the maritime applications as well as communication methods of the devices.

To conclude, the focus of this thesis is on MQTT and CoAP as the nominated protocols to explore more and present the implementation process.

2.3 MQTT[14]:

MQTT stands for Message Queue Telemetry Transport, and it is designed for constrained devices and low-bandwidth or unreliable network. This principle makes this protocol ideal for “A machine-to-machine (M2M) or "Internet of Things" connectivity where bandwidth and battery power have top priority. It was designed as an extremely lightweight publish/subscribe messaging transport.

2.3.1 MQTT Architecture

MQTT is based on TCP/IP, which works on top of the transport layer to transfer messages. This protocol was initially developed by IBM, however, now it is an open standard. The architecture of MQTT is a client/server model, where each sensor is a client and connects to a server known as broker. The *MQTT architecture in Figure 14* shows the publish/subscribe model.

MQTT Quality of Service (QoS) supports three levels of service as:

- 0 “Fire and forget” means at most once delivery tries.
- 1 guarantee to deliver the message at least once.
- 2 is highest level and means exactly once.

[14] **http://mqtt.org/**

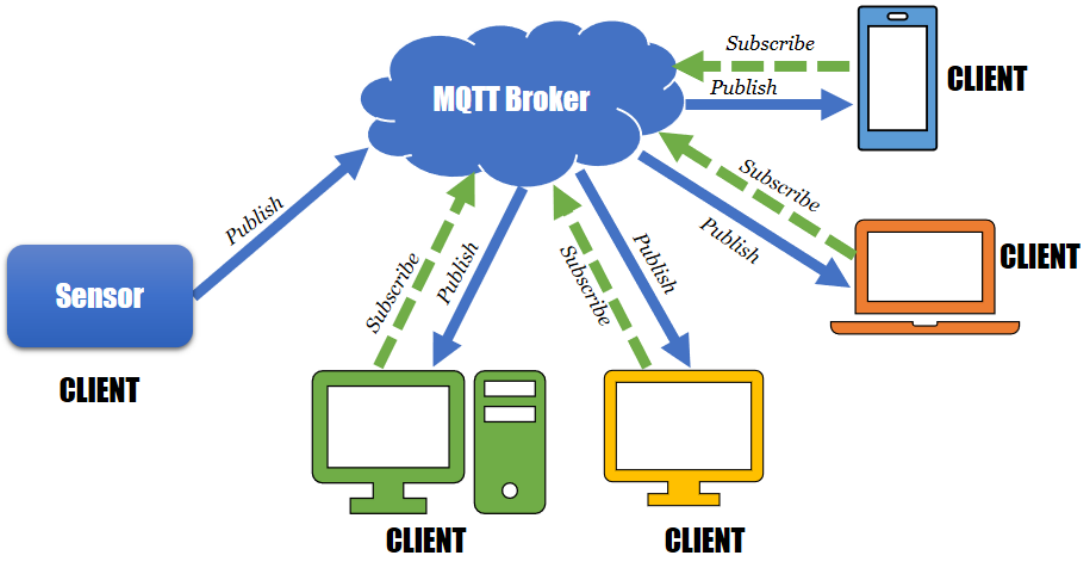


Figure 14 MQTT Publish/Subscribe Architecture [28].

Following, the brief explanation of MQTT terms associated to each role in the architecture is provided [29].

Broker: The broker is the heart of every publish/subscribe protocol. It is responsible to receive, filter and determine who is subscribed to what message.

Client: Any device like a PLC or micro controller that runs MQTT library and connects to a broker over a network to either publish or subscribe to a topic or both.

Publish/Subscribe: The device can publish messages to other devices and subscribe means the device can subscribe to a specific topic of messages.

Messages: Information exchanged between two or more devices. It can be either command or data.

Topic: It is a case-sensitive UTF-8 string to address the location of published messages. The topic is separated by slashes "/" which is also indicating the topic level.

2.3.2 MQTT Implementation

Eclipse Mosquitto is used to implement a light version of this protocol locally. Mosquitto's website[15] defined this tool as an open source message broker that implements the MQTT protocol and provides a lightweight method.

Mosquitto is suitable for use on all devices from low power single board computers to full servers. This protocol provides the publish/subscribe model and it makes it is suitable for Internet of Things messaging in sensors, mobile phones or microcontroller [30].

[15] **https://mosquitto.org/**

In order to run the Mosquitto MQTT broker, we need to first download [16] its module and then install it. Then, to start the Mosquitto service we need to open the command prompt and navigate to the path, where Mosquitto is installed (by default it will be in C:\Program Files (x86) \ mosquitto).

Now, two terminals are required to monitor publish and subscriber messaging. On the first terminal, we create a message-topic by –t parameter and the message by –m.

In the below sample and *Figure 15*, we demonstrated how to publish the cargo temperature as part of cargo safety data, which is explained in *section 1.5.1*. Therefore, "temperature" is the topic and multiple temperature degrees are the messages.

```
mosquitto_pub -t 'temperature /topic' -m "33"
mosquitto_pub -t 'temperature /topic' -m "30"
mosquitto_pub -t 'temperature /topic' -m "14"
mosquitto_pub -t 'temperature /topic' -m "45"
```

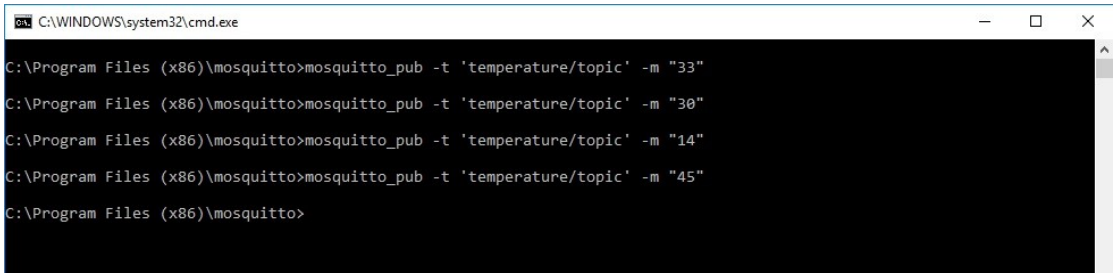


Figure 15 MQTT publish temperature.

On the second terminal, it needs to be subscribed to the "temperature" topic to receive the message. In order to specify the topic name after -t we should write the desired topic name and in order to receive the value -v should be added.

The following command and *Figure 16* shows how to subscribe to the "temperature" topic on the second terminal as well as the received messages:

```
mosquitto_sub -t 'temperature/topic' -v
```

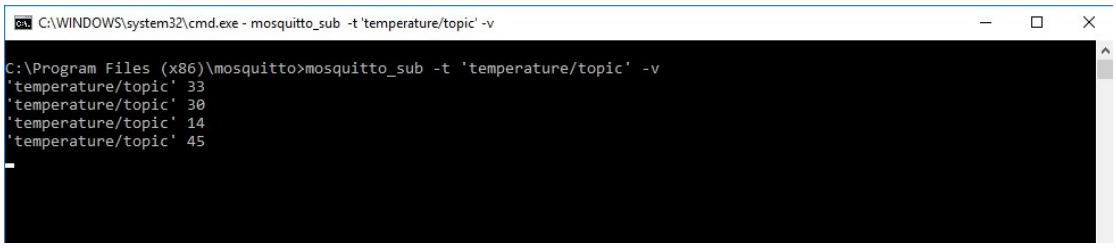


Figure 16 MQTT subscribe to temperature.

[16] **https://mosquitto.org/download/**

A simple Python script that provides a client class which enables an application to connect to a MQTT broker to publish a message and subscribe to a topic is presented in *Appendix I*.

2.4 CoAP[17]:

Based on the Internet Engineering Task Force (IETF) definition, The Constrained Application Protocol (CoAP) is a standard web transfer protocol which can be used in constrained nodes and constrained networks. CoAP is designed specifically for machine-to-machine (M2M) applications. Besides, this protocol provides a request/response communication between applications by including the core concept of Web such as URI (Uniform Resource Identifier) and internet media types [31].

2.4.1 CoAP architecture

Like HTTP (Hypertext Transfer Protocol), CoAP is based on the REST architecture request/response model, but unlike HTTP, CoAP operates over the UDP network layer and it is designed for constrained devices. CoAP provides URI, REST methods such as GET, POST, PUT, DELETE. CoAP packets are much smaller than HTTP, therefore they are simple to generate and consume extra less RAM in devices.

CoAP Quality of Service (QoS) support two levels of delivery, thus, request and response messages may be marked as confirmable or nonconfirmable.

- Confirmable messages must be acknowledged by the receiver with an acknowledged packet [32].
- Non-confirmable messages mean unreliable and transporting without acknowledgement (fire and forget) [32].

Electrical control devices such as PLC can benefit from the low-cost and lightweight characteristic of the CoAP protocol and can setup their system with either Local Area Network (LAN) or Internet.

Figure 17 is an example of an energy control system that each data collection node can exchange and interact with other nodes. In this example, there are three nodes, which are collecting data and send it to the control server over the CoAP protocol and on the other side of this system, there is a web server to share the collected data to different end users through the web application.

[17] **https://coap.technology/**

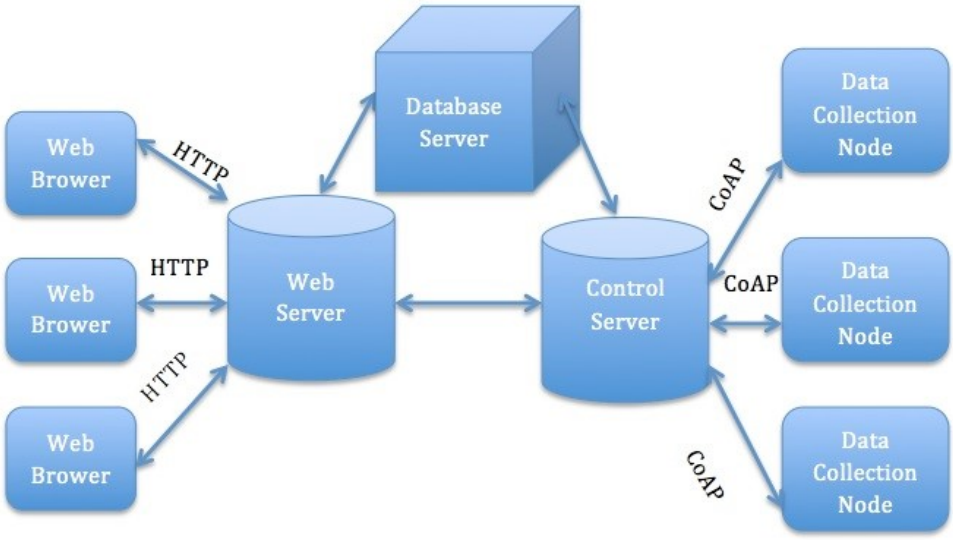


Figure 17 Energy control system [32].

2.4.2 CoAP implementation

Like the REST architecture, CoAP is very similar to HTTP request methods (GET, POST, PUT and DELETE) with a specific response code. These methods are being used to create, update, fetch data and delete the resource on the server, which are presenting an IoT application. Frequent response codes are shown in *Table 6.*

Table 6 CoAP response code

CODE	RESPONSE
2.01	Created
2.02	Deleted
2.04	Changed
2.05	Content
4.04	Resource not found
4.05	Method not allowed
5.XX	Server error

In this part, we demonstrate how to use the HTTP-CoAP proxy to request the resources from the CoAP server, which are accessible via HTTP[18]. There is a testing URI *coap://coap.me:5683* on the CoAP server that allows submitting requests by simulating a request and response model.

Figure 18 shows the "hello" GET request and receive "world" as a reponse.

[18] **http://coap.me/**

Figure 18 CoAP GET request.

The DELETE request with the response 2.02 as deleted is presented in *Figure 19*.

Figure 19 CoAP DELETE request.

Figure 20 depicts a successful POST request with the response code 2.01 that shows the location of the updated post.

Figure 20 CoAP POST request.

The PUT request with the response code 2.04 demonstrates a successful data creation and it is presented in *Figure 21*.

Figure 21 CoAP PUT request.

2.5 Security Mechanism

Ships are increasingly using systems that rely on digitalization. As the technology continues to develop, the systems onboard ships are more frequently connected and this will introduce the greater risk of cyber security. During the recent years, several examples of IoT incidents with different devices have been reported[19]. Security concerns in the shipping industry can specially be reflected in different examples, such as the corruption of chart data in an Electronic Chart Display and Information System (ECDIS) or as a failure during software maintenance or unauthorized access to the ship's data.

[19] **https://www.zdnet.com/article/5-nightmarish-attacks-that-show-the-risks-of-iot-security/**

Generally, variety of applications with different architectures might suffer from poor security. *Table 7* presents the top ten most security concerns in IoT [33].

Table 7 Top ten vulnerabilities in the IoT system [33]

Security concerns	Interface layer	Service layer	Network layer	Sensing layer
Insecure web interface	✓	✓	✓	
Insufficient authentication/authorization	✓	✓	✓	✓
Insecure network services		✓	✓	
Lak of transport encryption		✓	✓	
Privacy concerns		✓	✓	✓
Insecure cloud interface	✓			
Insecure mobile interface	✓		✓	✓
Insecure security configuration	✓	✓	✓	
Insecure software/firmware	✓		✓	
Poor physical security			✓	✓

Shancang Li in [33] explained the security mechanism in IoT and defined a four-layer architecture in this regard. Each layer is responsible to provide security controls such as access control, device authentication, data integrity and confidentially in transmission. These layers are defined as [33]:

- Sensing layer: To sense and acquire data from the end components.
- Network layer: To support the wireless or wired connection.
- Service layer: To provide and manage required services for users or applications.
- Interface layer: To deliver interaction methods with users or applications.

Security in the sensing layer is extremely important because this is the first layer to sense and acquire the data from end components. Therefore, the focus of this section is to understand how to establish a secure connection in the MQTT and CoAP protocols.

Table 8 summarizes the possible potential threats and security vulnerabilities in the sensing layer [33].

Table 8 Security threats and vulnerabilities in the sensing layer [33]

IoT end-node threats and vulnerabilities	IoT end-devices	IoT end-node	IoT end-gateway
Unauthorized access	✓	✓	✓
Selfish threat		✓	✓
Spoofing attach		✓	✓
Malicious code	✓	✓	✓
Dos	✓	✓	✓

Transmission threats			✓
Routing attach	✓	✓	✓

2.5.1 MQTT Security

The security specification of the MQTT protocol is based on the OASIS[20] (The Organization for the Advancement of Structured Information Standards) standard and the document is publicly available[20]. This protocol has three security mechanism methods and the main structure is to verify the identity of the MQTT client.

Each of three methods is briefly explained as follow [34]:

Client identifiers:

The Client Identifier (ClientID) identifies the MQTT client to the MQTT broker. The broker uses ClientID to identify the client and states of it. Therefore, when a client is subscribed to a particular topic, the ClientID associated to the client is sent to the broker. Thus, ClientID should be a unique number per client and broker.

Username and password:

The MQTT broker can request a valid username and password from a client before a connection is granted. The username and password are transmitted in a clear text. A Virtual Private Network (VPN) between clients and servers can establish a secure connection and guarantees that the data is only being received from the authorized clients.

Client certification:

The most secure but also the most difficult method to implement is the client authentication, as it needs to deploy and manage certificates on each client. Because of that, this form of authentication is really only suited to a small number of clients that needs a high level of security.

2.5.2 CoAP Security

The CoAP security is based on the DTLS (Datagram Transport Layer Security) protocol, which is an enhanced version of the TLS (Transport Layer Security) protocol but the major difference is that DTLS runs over UDP instead of TCP. In [35] Security analysis of CoAP in IoT has been discussed. Authors explained that, DTLS provides authentication, data integrity, confidentiality and automatic key management. The DTLS protocol also supports the wide range of different cryptographic algorithms, and defines four type of security methods, these methods are:

- NoSec

In this mode, there is no protocol-level security as the DTLS protocol is disabled. The system sends the packets over normal UDP and it is indicated by the coap scheme and the CoAP default port. The *coap://* scheme has been discussed in *section 2.3 CoAP* implementation.

[20] **https://www.oasis-open.org/standards**

The rest of following security modes are accomplished by the DTLS protocol over *coaps://*.

- PreSharedKey(PSK)

A shared key between all nodes that will be used during the communication with the CoAP nodes.

- RawPublicKey(RPK)

The device has an asymmetric key pair without a certificate and the DTLS protocol is enabled.

- Certificate

The device has an asymmetric key pair with a X.509 certificate along with the DTLS protocol that binds it to its authority name and it signed by some common trust roots.

Chapter 3

Maritime Cloud

One of the most popular technological terms in the recent decade is cloud computing. The fundamental concept behind cloud computing is that the location of infrastructure does not concern the services and users. In fact, cloud computing is the delivery of on-demand computing services over the internet to the users.

Accordingly, the cloud computing services have a crucial role for the future connectivity in the shipping industry. Connectivity can boost dynamic routing when major factors such as weather, efficient route and traffic are identified. In addition to that, the complete shipping process time can be enhanced when cargos arrive on time.

Having said that, a secure, reliable and seamless cloud framework is still lacking in this respect. Therefore, in this chapter we explain a few challenges behind the marine industry cloud structure and propose a possible method in order to design a framework.

3.1 Definition of Cloud Computing

According to the National Institute of Standards and Technology (NIST), a general definition of cloud computing is as follows [36]:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

Based on the above definition, a cloud model is composed of five essential characteristics, three service models and four deployment models. The NIST model can be found in *Table 9* [36].

Table 9 NIST Cloud Model [36]

Essential Characteristics	Service Model	Deployment Model
On-demand self-service	Software as a Service (SaaS)	Private Cloud
Broad network access	Platform as a Service (PaaS)	Community Cloud
Resource pooling	Infrastructure as a Service (IaaS)	Public Cloud
Rapid elasticity		Hybrid Cloud

Measured service

Since the focus of this chapter is on the cloud capabilities, in the following each of five essential characteristics of the NIST cloud model is shortly explained [36].

Five essential characteristics:

- On-demand self-service: Computing resources such as server time and network storage can be delivered automatically and do not require a human to interact with each resource.
- Broad network access: Cloud capabilities are available over the network and accessible through different platforms such as mobile phones, tablets and laptops.
- Resource pooling: Computing service providers are able to serve physical and virtual resources dynamically. In the meantime, customers have no knowledge or control over the exact location of the provided service, but they might be able to specify country, state or datacenter at the high level.
- Rapid elasticity: It is possible for consumers to scale outward and inward the cloud capabilities in any quantity at any time.
- Measured service: Cloud resources can be monitored, controlled and reported to both cloud service providers and consumers and based on that resources can be optimized.

3.2 Cloud computing in the shipping industry

Digitalization in the shipping industry has many ways of implementation, for instance vessel operation and navigation or asset tracking application. This means the vast amount of data should be stored and analyzed from the different perspectives, and this can be various data points, such as information related to a voyage, or data from vessel equipment. Data diversity has been discussed in more details in *section 1.5.1*.

The Danish Maritime Authority (DMA) proposed a cloud infrastructure model in this industry [1] and the model illustrated in *Figure 22*. In this proposal, a cloud service facilitates secure interoperable information exchange between services and stakeholders. Moreover, the proposed cloud model consists of three components, i.e.

- “Maritime Service Portfolio Registry” that holds information about capabilities of services and associated users.
- “Maritime Identity Registry” that maintains the identity and holds the authentication and confidentiality.
- “Maritime Messaging Service” to provide the unified way of communication between ship-to-ship and ship-to-shore.

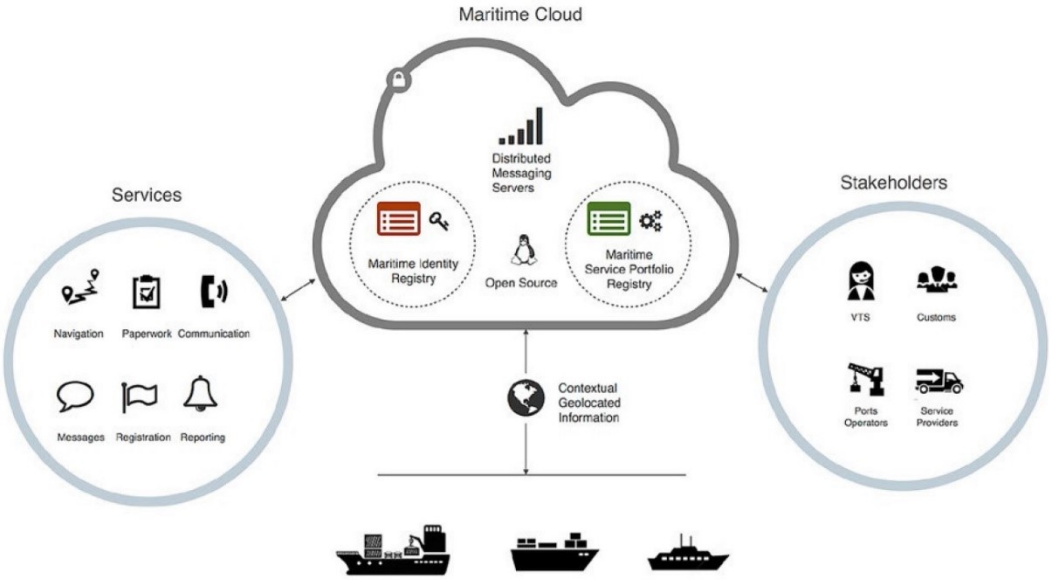


Figure 22 Maritime Cloud Infrastructure [1].

Even though the mentioned cloud model provides a clear method, there are some limitations to adopt cloud solutions on a vessel, a few of which have been discussed in *section 1.3* as part of requirements.

In addition to that, as mentioned in *section 3.1,* a cloud model consists of five essential factors, in which on-demand self-service and broad network access are relatively expensive due to the use of VSAT communication and international roaming charges.

Another obstacle is the difficulty of connectivity in an open area such as the Pacific Ocean. In that case, ship operators prefer to minimize the need for communication and try to communicate only to receive the crucial information.

Thus, the list of challenges with highest priority in the shipping industry according to this thesis is:

- Lack of access to the internet during the voyage due to the cost.
- Restricted bandwidth due to the cost.
- Large amount of data because of various data points.

3.2.1 Offshore ship service model

In order to tackle the above challenges, the framework proposed in Hao Wang [37] for offshore support vessels[21] (OSVs) has been studied.

This study depicts that most of the technology providers for offshore vessels are targeted the land-based applications. Additionally, an OSV operates in a special and difficult environment and relies mostly on satellite communication which is very

[21] **Offshore vessels are vessels that specifically support the operational purposes and construction work at the high sea. These types of ships are mainly used for excavation, oil exploration and oil drilling.**

expensive. Considering these facts, it is unrealistic to use the cloud-based big data analytics (BDA) solution and install powerful computing facilities onboard an OSV.

Because of these challenges, this study introduced a two-layer BDA-IIoT framework for OSV in which the first layer associated to the vessel and the second layer linked to land [37].

Figure 23 shows the vessel BDA layer and the land BDA layer. In this framework, BDA and industrial IoT (IIoT) integrated on a hybrid CPU/GPU high performance computing platform.

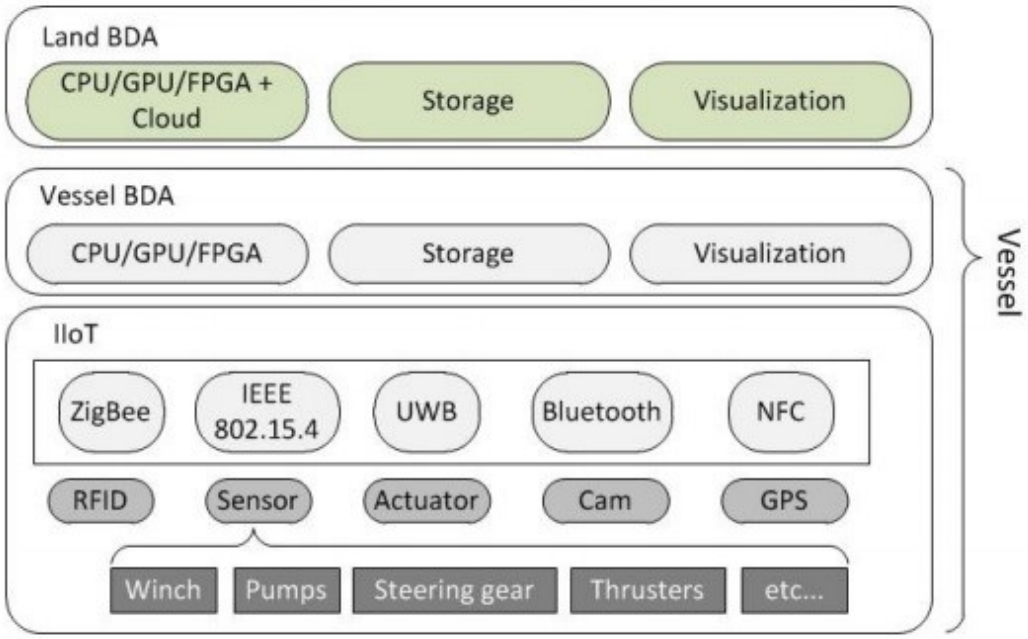


Figure 23 Conceptual 2-layer BDA-IIoT Framework for OSV [37].

The vessel layer consists of the vessel's local data processing and analyzing facilities for real-time BDA need. Due to the limited availability of computing resources onboard a ship, the vessel layer focus is on real-time descriptive, predictive and prescriptive analytics to support the ship operation.

On the top layer, the land BDA emphasis is on analytic tasks of the large historic vessel data to facilitate the future ship design and maintenance. [37]

3.2.2 Container ship service model

In "The implementation of cloud computing in shipping companies" [38], Pančo Ristov has argued what is the proper service model to improve the efficiency and security of the business in the container ships.

According to the NIST cloud model which is discussed in *section 3.1,* there are three different service models, software as a service (SaaS), platform as a service (PaaS) and

infrastructure as a service (IaaS), and between these services Pančo Ristov suggested the SaaS model is an acceptable model to present the cloud services [38]. This suggestion is based on the analysis of business processes on board ships and shipping companies. Besides, in this study it is recommended that the SaaS model should include the following functional modules [38]:

- Ship/Fleet management – this module helps crew to schedule and execute their tasks effectively.
- Maintenance management – this module allows planning and executing the ship maintenance procedure and spare parts management.
- Document management – this module helps to manage the document like document editing, archives, distribution and document versions.
- Reporting – this module enables a dynamic standard on reports according to various criteria.

3.3 Cloud model proposal

According to the described cloud service models in *section 3.2.1* and *section 3.2.2* and also the mentioned challenges in *section 3.2*, the most important condition for a remote monitoring application is to have a decentralized computing system. It means that the part that mainly supports the ship operation should be on board the ship and another part that focuses on analytic tasks of large and historic vessel data should be on land. Based on the given facts, fog computing has been considered as a suitable model and in the following section we will discuss how fog computing can assist the marine industry to facilitate their process.

3.3.1 Fog computing architecture

Fog computing emphasizes processing data close to the edge of network rather than sending the information to the cloud or data center. This helps to reduce the amount of data by processing information beforehand and increase the performance of critical applications [39]. Therefore, fog computing is trying to analyze data close to where the devices are receiving them.

Cisco believes that the current cloud models are not designed for the four Vs of big data for IoT applications, instead fog computing is designed to answer this need [39].The four Vs in big data are Velocity, Variety, Value and Volume and we will explain them in *section 5.1* .

In this regard, an open reference architecture for fog computing from OpenFog Consortium[22] aimed to standardize and promote fog computing in various applications. This consortium is an independent and open membership ecosystem, which is founded by Cisco, ARM, Microsoft, Intel, Dell and Princeton University[23].

[22] **https://www.openfogcosortium.org/**
[23] **https://www.openfogconsortium.org/membership-information/#member-companies**

Fog computing overcomes the aforementioned challenges in *section 3.2* and provides location awareness, low latency as well as Quality of Service (QoS) for the real-time applications [40].

Lately, many architectures have been used for fog computing, but in [40] Gohar explained the three-tier architecture in which the cloud layer at the top, the fog layer is in the middle and the nodes like IoT devices and the sensors are located at the bottom layer. A short explanation of each layer follows.

- Cloud layer: the top layer can store and process a massive amount of data and it is responsible for analyzing data and store it permanently.
- Middle layer: the middle layer or the fog layer is deployed between the cloud and IoT devices and it is responsible for transmitting data between them. The fog layer contains network devices such as routers, access points, gateways and switches. This layer is explained in more detail *in 3.3.2.*
- Device layer: this layer contains IoT devices and end devices such as mobile phones and they are distributed geographically to sense information from different physical objects or events.

3.3.2 Fog layer

Applications that are developed through the fog layer have common requirements: firstly, the solution should be able to collect information from end devices and secondly perform real-time analyses. These applications are mainly based on an open Linux that enables port management of IoT applications for a smooth communication.

Fog API

Figure 24 demonstrates how a Cisco router with fog API works. This figure shows an SaaS cloud model and in that, Cisco IOx hosts applications in a Guest Operating System (GOS) that runs on a hypervisor on a Cisco fog node. IOx usually runs over Yocto Linux, however, it is possible to use any OS [41]. Furthermore, the communication aspect in a fog SaaS model is based on an Internetwork Operating System (IOS) and a Linux operating system, so together it creates IOx.

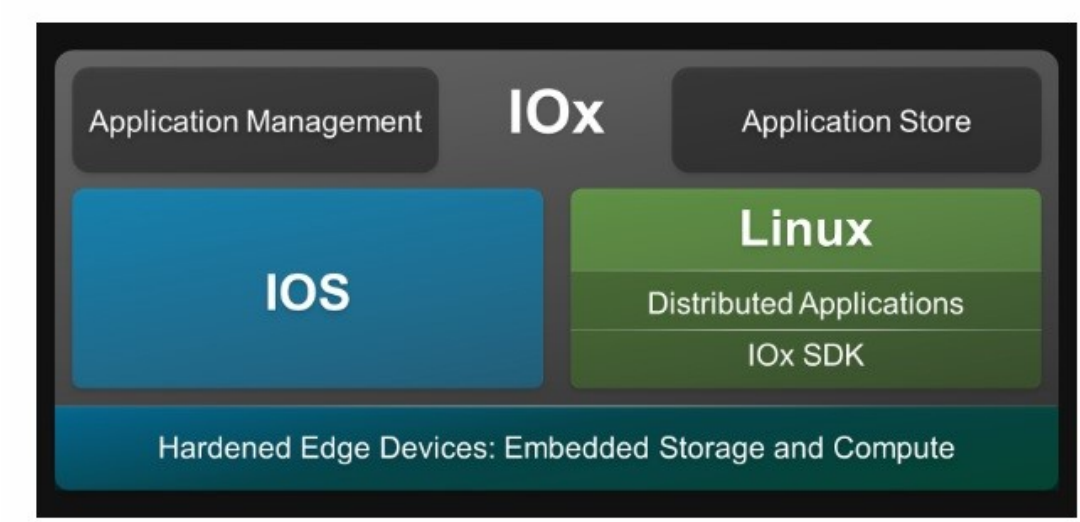


Figure 24 How Cisco IOx Works [42].

Data management

Another important aspect of a fog application is real-time analysis and the purpose of that is to manage four Vs of big data (Volume, Velocity, Variety and Value) by processing and analyzing information before transmitting them to the cloud. By that, it will minimize the latency and network traffic and instead of sending sensitive data to the network, it will keep the data inside the node [41].

To understand the data flow through a fog application, *Figure 25* presents a general view that shows fog nodes collecting data from the end devices and then transferring it to the fog data services. Finally, a cloud platform and the fog data services navigate data through the REST APIs.

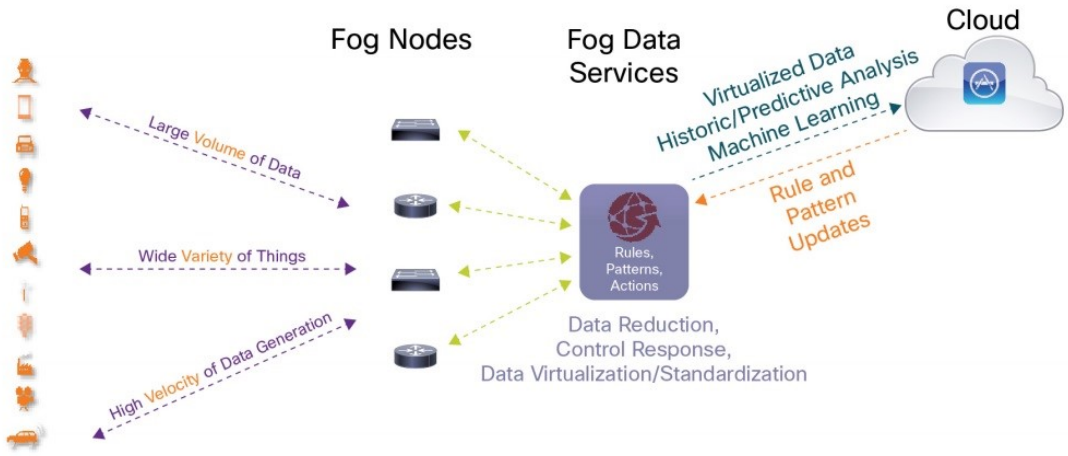


Figure 25 Fog Data Services Coordinate the Movement of Data from Fog to Cloud. [41].

3.3.3 Fog and Edge Computing

The comparison between fog and edge computing is out of the scope of this thesis, but these terms are often used interchangeably, so a short explanation of edge computing and the key difference will be provided here.

Fog and edge computing are frequently used in Industrial Internet of Things (IIoT) through an architecture layer and are considered to bring the intelligence and processing closer to the data creation points, however, the key difference is the location of intelligence and computer power.

Fog computing relies on local area network (LAN) to place the intelligence and transmit data from endpoints to a central unit, whereas the edge computing places intelligence and processing power in devices [40].

Embedded devices such as PLC controllers with small memory storage which are connected based on peer-to-peer (P2P) technology is a common example of edge computing.

Chapter 4

High-level Architecture Design

This chapter will demonstrate a hypothetical architecture of a cargo monitoring system built on fog computing. This solution is based on a two-layer architecture in which the first layer demonstrates how containers communicate through a central unit and the second layer presents ship-to-shore communication.

4.1 Vessel layer

The vessel layer aimed to provide real-time awareness for the ship crew through a SaaS service, and to do that each container is armed with a sensor and an UHF technology[24] then they are able to update their status based on the MQTT protocol.

This means that, a broker is subscribed to the cargo safety data and receive their condition on a regular basis, after that the broker publishes the information to a central unit at the ship bridge. Ultimately, a fog server at the bridge collects data and informs crew about goods status. This layer is demonstrated in *Figure 26*.

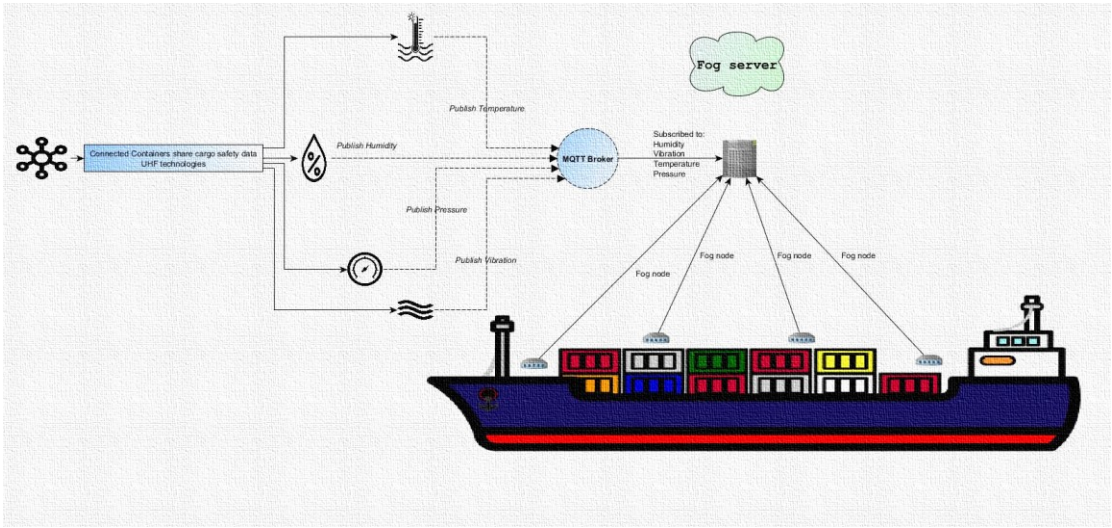


Figure 26 Cargo Monitoring Architecture - Vessel layer.

[24] **UHF technology is discussed in** *section 5.3.1*.

4.2 Shore layer

Ship-to-shore communication can be performed in two ways with respect to the ship location.

One situation is when a ship is sailing and it can connect to a satellite, hence it sends information to the shore to share general information about the goods status and updates the related stakeholders like cargo owners and ship owners. This information can be presented through a KPI report on a daily basis, for instance.

Another situation is when a vessel is at port therefore it can use the port network like Wi-Fi to send the voyage data, which are accumulated from the past sailing. This data can be used for the future development based on the operation performance or ship maintenance according to the equipment data.

The general design of the mentioned communication methods can be found in *Figure 27*.

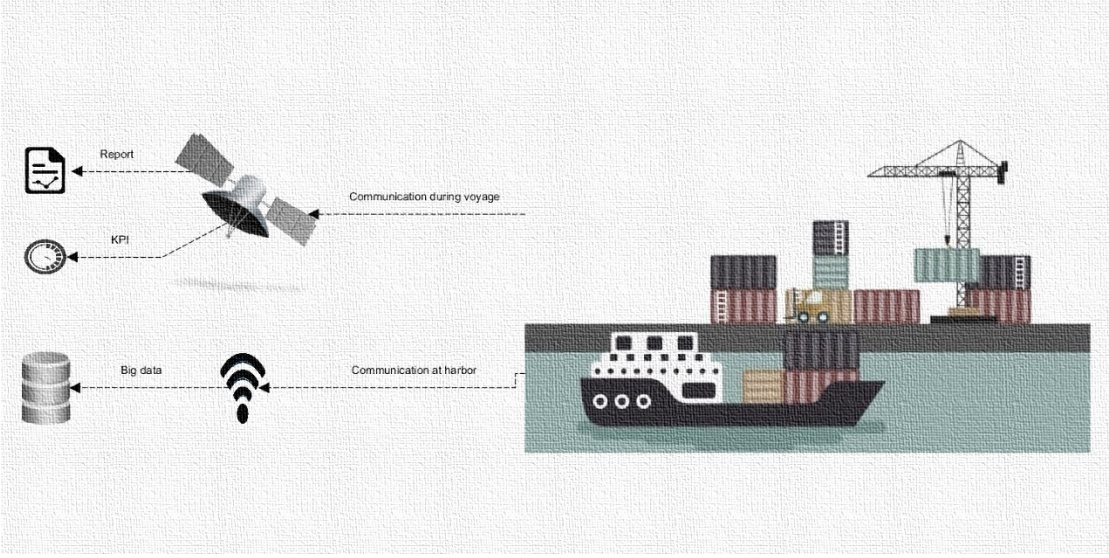


Figure 27 Cargo Monitoring Architecture - Vessel layer.

Chapter 5

Data Management

Data management aspect in the shipping industry is highly dependent on organization maturity and data quality. Maturity in this context means how companies respond to the unwanted events and how well they can prevent those events or minimize the probability of accidents. These knowledge-based decisions are directly related to information and facts about the business process. It is required to have a precise understanding on the digital capabilities of an organization to setup proper technologies as well as architecture to extract data with high quality and accuracy.

In the vessel technology sector, the outcome values for a company by having standard data models will be deep understanding of vessel operation and cargo safety. Moreover, these fact-based approaches will help corporations to optimize the logistic and anticipate future failures.

Through the Inmarsat research program [43] in 2018, which was focused on understanding the way that Industrial Internet of Things (IIoT) is being adopted by organizations in different sectors, they found that, majority of respondents in the maritime industry use or will use data for "monitoring and improving health and safety" standards. Respondents were from different organizations either decision-maker or responsible for IIoT initiatives.

Figure 28 is presenting the percentage wise on how operators in the maritime industry are intending to use the collected data through IIoT-based solution.

Figure 28 Data Usage by percentage [43].

5.1 Key principles and Challenges

Generally, four V's of big data are the well-known principles to classify data management. These four V's are Volume, Velocity, Variety and Value. A brief explanation of each of them as follow [44].

- Volume stands for all types of data generated and published from different sources continuously.
- Velocity is used to define the transmission speed of data whereby data is generated.
- Variety is referring to different types of collected data from different channels. The channel can be digital devices such as smart phones, sensors and in different context like audio, video, image etc.
- Value is interpreted as the most significant aspect of big data since it is the process to find how worthy is the data.

Nevertheless, to evaluate the reliability of data management we have to consider different characteristics. The evaluation should be more focused on the nature of business and circumstance of a vessel. Similar to the definition of cloud computing in *section 3.2*, it is crucial to understand the challenges and the importance of data management on board a ship.

Big Data Challenge

One of the challenges is to find a clear definition of big data on the shipping industry since each party and stakeholder in this ecosystem is interested in different perspectives of information. For instance, ship operators are mainly concerned about the efficiency while ship builders are more interested to know the new methods to reduce the operation cost. Besides, maritime authorities like DNV-GL[25] concentrates on how to robust the ship structure. Therefore, the needs and interests are varies based on each stakeholder.

Therefore, in order to answer the mentioned challenge and understand the major factors in big data, the paper from DNV-GL [45] has been studied. In [45] DNV-GL has shared the main areas that should be considered in the big data development. These factors are as follows.

- Technical operation and maintenance
- Energy efficiency (cost and environment)
- Safety performance
- Management and monitoring of accident and environmental risk from shipping traffic
- Commercial operation (as part of logistic chains)
- Automation of ship operations (long-term)

Consequently, service providers and stakeholders need to approach the above factors and data as an asset through a flexible framework to execute the projects. Thus, it is required to combine knowledge from domain experts and data scientists to a

[25] **DNV-GL is an international quality assurance and risk management company, which provides classification, technical assurance, software and independent expert advisory services to the maritime industry. (https://www.dnvgl.com/)**

framework. In *Figure 29* mandatory components to have an efficient and flexible framework is presented [45].

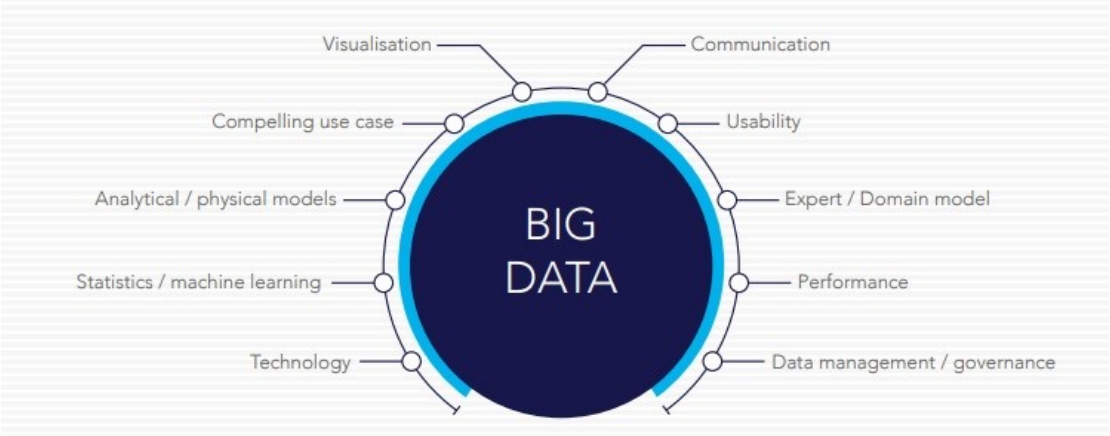


Figure 29 Key components/capabilities for successful big data application [45].

Capacity Limitation Challenge

Another challenge is data swamping and it happens when a system encounters with an overwhelming amount of raw data. Assuming that 1000 sensors on board a vessel a 1 Hz sampling rate creates roughly 86 million bytes daily and 31 billion bytes annually. If each data point is 4 bytes to store humidity, temperature, pressure and vibration, accumulated data will be about 126 GB[26] of sensor data every year. In addition to that, this amount of data on a fleet that consists of N number of ships will be multiple to that. We will answer this challenge in the *data size section*.

5.2 Scope Definition and Limitation

The aim of this chapter is to find the general requirements for a cargo awareness system for the both ship and shore layer. To achieve that, in this section we assumed and estimated key factors such as departure port, sailing period and ship size are identified and measured.

5.2.1 Route

In order to calculate the amount of data and evaluate the cost according to the proposed IoT and cloud model in *section 3.3*, we need to have a voyage route and the ship size to determine the number of data points. The assumption is a voyage from the Shanghai port to the Rotterdam port, both is introduced in *section 1.3.2* .

[26] **The gigabyte is a multiple of the unit byte for digital information.**

5.2.2 Voyage

The exact amount of time for a container vessel to travel between Shanghai to Rotterdam or vice versa depends on the exact route and the number of stop on the way to the final destination. Nevertheless, this journey is almost between four weeks to five weeks to complete.

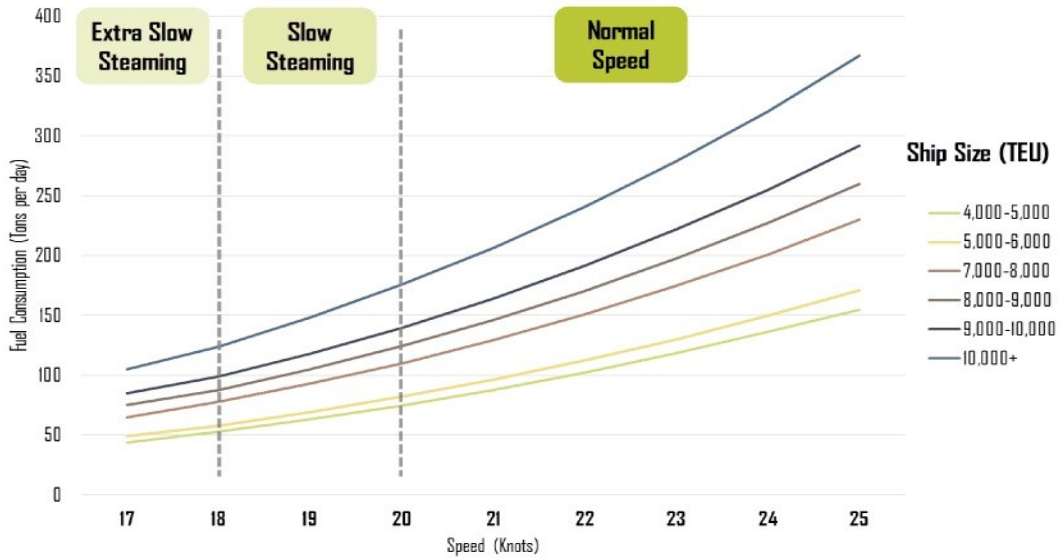
For instance, we calculated the estimated delivery time for some freight through the Searates[27] website, and based on the origin port (Shanghai) and the destination port (Rotterdam) with the average speed it takes approximately 31 days to deliver the cargo.

5.2.3 Ship size and speed

In the shipping industry, fuel consumption is mostly associated to the ship size and the cruising speed. In normal speed, a containership of around 9,000 TEU[28] at 25 knots[29] (46.3 kilometer/hour) speed would consume about 280 tons of bunker fuel every day.

However, by reducing the speed from 25 knots to 21 knots consumption drops to 150 tons, but clearly the shipping time would be longer. In *Figure 30,* we can see how fuel consumption changes with respect to speed and containership.

Therefore, the slow streaming approach to reduce fuel consumption in vessel operations during the financial crisis of 2008-2009 was introduced. This approach introduced by Maersk Line, and it proved that by adopting this practice and decrease the speed from 25 knots to 18 knots fuel consumption can be dropped by 65.27% [46]. Slow streaming applies when a vessel sailing at 18 knots and if speed is dropped to under 18 knots it is being called extra slow steaming. Thus, vessel size does not affect the arrival time and majority of vessels are sailing with the same speed so the assumption for the ship size is a 10k TEU.



[27] **https://www.searates.com/**

[28] **The twenty-foot equivalent unit is approximately a unit size of cargo and it is equal to 6.1 meter long.**

[29] **The knot is a unit of speed equal to one nautical mile per hour, exactly 1.852 km/h.**

Figure 30 Fuel Consumption by Containership Size and Speed [47].

5.2.4 Assumption data

In *Table 10,* we presented all aspects that are considered in our assumption according to the *route, voyage* and *ship size* sections.

To simplify the scope, each container is counted as a data point to collect cargo status information per second. This means that, a ship with 10,000 containers capacity has 10k sensors installed to collect humidity, temperature, vibration and pressure per second.

Table 10 Assumption information

Key	Description
Route	Shanghai – Rotterdam
Voyage	31 days
Speed	18 knots
Ship size	10,000 TEU
Number of sensor	10 k
Status information	▪ Humidity ▪ Temperature ▪ Vibration ▪ Pressure
Data interval	Per second

5.3 Vessel layer

In the vessel layer, first we determined general requirements for a cargo status system and then explained each of them in details.

In this layer, the focus is on the safety of cargo and on-time notification. The crew on board a ship need to know the status of cargo early enough to be able to take needed actions. This principle is the essence of a cargo monitoring system and *Table 11* shows the basic requirements and description to implement it.

Table 11 Onboard requirements

Requirement	Description
Radio frequency	To understand the acceptable radio waves and technologies.
Coverage	To find the coverage range in a vessel.
Onboard storage size	To keep and maintain data during a voyage.
Data packet / energy consumption	To find the lightweight and efficient interaction protocols. This part has studied in *chapter 2.*

5.3.1 Radio Frequency

Maritime classification societies are the non-governmental organizations that define and establish rules and technical standards for the shipping industry. Regarding the internal communication requirements, Global Maritime Distress and Safety System (GMDSS)[30] and DNV-GL have provided a guideline of general ship requirements for the internal communication [48].

Based on the mentioned guideline [48], Ultra High Frequency (UHF) is the acceptable frequency range onboard a vessel. Current UHF technologies are shown in *Table 12*. It is beyond the scope of this thesis to determine which of them should be considered in the development phase, but in general, Wi-Fi and Bluetooth are commonly used on board the ships.

Table 12 UHF Technologies

Ultra-High frequency technologies samples (in satellite communication and radio service)
GPS
Wi-Fi
Bluetooth
Walkie-talkie

5.3.2 Radio wave coverage

To understand the required coverage of radio waves, it is important to know the ship dimensions. The dimension of a vessel is defined during the ship design phase and it is based on capacity (TEU), desire speed, sailing route, type of cargo etc.

In fact, the total number of possible containers to load on a ship is directly related to its dimension, even though two ships might have the same TEU. For instance, a vessel with 10 thousand TEU can load almost 9000 containers whereas another vessel with the same TEU might be loaded with 5000 to 8000 containers. Generally, in *Table 13* a 10,000 TEU vessel dimension is shown [49].

Table 13 10k TEU Container ship size

Angel	Size
Length	366 m
Draft (depth)	15.2 m
Beam (width)	49 m

[30] **http://www.ccg-gcc.gc.ca/eng/CCG/SAR_Gmdss**

5.3.3 *Data size*

The role of physical data storage is crucial since it is used to store and maintain information onboard a vessel and in order to tackle the data swamping challenge that we mentioned in the *challenges section,* it is vital to calculate the amount of data in advance to setup the necessary hardware. Hardware storage can be integrated into sensor devices such as memory stickers or it can be a central unit to collect data from all nodes.

Based on the *assumption table* we considered a sensor that senses humidity, temperature, pressure and vibration. In the meantime, sensors' data are inaccurate due to the varying geographical location during a voyage, and it might be impossible to determine a fixed range, therefore, in *Table 14,* we mentioned a possible range for each sensor and a short description as below.

- Humidity is calculated based on water in dry air by percentage.
- The possible temperature is calculated based on the historic data for a voyage from Shanghai to Rotterdam [50] [51].
- The maximum and minimum pressure at sea is considered in data range [52].
- Since it is complicated to determine a range for vibration, we assume vibration in percent.

Table 14 Data type and range

Sensor data	Size	Range
Humidity	1 byte	0 to 100%
Temperature	1 byte	-60 to +60
Pressure	1 byte	923.6 to 1067.1 bar[31]
Vibration	1 byte	0 to 100%

To sum up the vessel layer, according to the *assumption table* we found that accumulated data for a 10k TEU vessel in one day would be 3.4 GB and as a result, data size goes up to roughly 107 GB for 31 days. The calculation is based on 10k sensors that collect 4bytes data per second and the details are presented in *Table 15.*

Table 15 Data size

Time	Seconds	Minutes	Hours	Days	31 days (voyage)
Data points					
10 k	40,000 B	2.4 MB	144 MB	3.456 GB	107.136 GB

[31] **Pressure unit: 1 bar= atmospheric pressure at sea level**

5.4 Shore layer

There are two requirements in the shore layer. First, we estimated the ship-to-shore communication cost and then explained why big data analytics knowledge is important for the future improvement from cargo owners' and shipping companies' perspective. In *Table 16,* the mentioned requirements and their short definitions are shown.

Table 16 Shore requirements

Requirement	Description
Cost	To find approximate cost to receive data from ships.
Data analytic	To understand values behind big data.

5.4.1 Cost estimation

The vessel communication methods are discussed in *section 1.5.2* and in *Figure 5,* the frequency band of different technologies is shown. Nevertheless, in order to have a clear view of the communication cost, a further study on major differences in VSAT system is provided in this section.

In the maritime market, VSAT solutions are typically delivered as a package that includes satellite space segments, equipment and phone and internet services. Besides, there are various different techniques to implement a maritime broadband network onboard a vessel and each has its advantage in cost, coverage and signal strength. Although VSAT radio frequency bands that communication operates within are C-Band, Ku-Band, Ka-Band and X-Band, the most commercial VSAT networks are C-Band and Ku-Band frequencies. In *Table 17,* these two frequency bands are demonstrated with the key differences.

Table 17 VSAT radio frequency

	C-BAND	KU-BAND
Frequency range	3-6 GHz	12-18 GHz
Data rate	Up to 4 Mbps	Up to 4 Mbps
Advantage	▪ Continent-wide coverage ▪ Allows operation regardless of weather conditions so most suitable for tropical	▪ Small antenna to operate. Antenna size is 0.6m - 1.8m. ▪ Higher frequency that provides stronger signals ▪ Less costly and easier installation.

	regions with heavy rainfall	
Disadvantage	▪ Large antenna for operations. Antenna size is 1.8m - 2.4m	▪ Costal and near global coverage ▪ Affected by weather conditions such as rainfall especially in tropical areas
Vessel type	Large vessels such as container ships	Small to large vessel
Pricing model	Flat rate	Flat rate

Since there are numerous manufacturers in the global maritime VSAT market, it is not feasible to provide an accurate price for this technology. Yet, there are two primary expenses that should be considered to setup VSAT networks: hardware/equipment costs and subscription/monthly service fee.

The average equipment cost in the majority of VSAT service providers[32] is presented in *Table 18* [53] [54] [55].

Table 18 VSAT hardware cost

	C-BAND	KU-BAND
Satellite antenna and router	$96	$54
Installation	$15,000	$5,000
Monthly maintenance cost after the guarantee period	$3,000	$1,600

The next expense is the monthly service fee and the cost of it is based on either bandwidth (speed) or the volume of data (amount of megabytes). It is also challenging to find an accurate price for both types because this calculation is mainly made according to the ship usage. In spite of the mentioned fact, in *Table 19* a few examples of typical minimum costs are provided. In this table, all measurements are based on a low usage profile, which means checking email, downloading weather files, internet browsing[33] [53] [54] [55].

Table 19 Bandwidth cost for low usage

Provider	Bandwidth	Data volume	Average cost per-MB
Iridium pilot	128 kbps	150 MB	from $0.26
Vt idirect	256 kbps	385 MB	from $1.70

[32] Manufacturer suggested retail price in the VSAT market such as Inmarsat, KVH Industries, VT iDirect etc.
[33] Around 20 web sites in a month with 10kb streaming size.

Inmarsat (fleet broadband model)	432 kbps	150 MB	from $0.70

We found in *section 5.3.3* that our hypothetical cargo monitoring system would accumulate 3.5 GB data every day. It is not logical and affordable to transfer this amount of data based on bandwidth cost; instead, this information can be analyzed and then turn to a simple KPI dashboard to update the shore layer about cargo status within past 24 hours so the KPI file size would be a few megabyte. Therefore, we estimated the cost in *Table 20* for a 5MB file size.

Table 20 Bandwidth cost

File size	Provider	Daily cost	Monthly cost
5 MB	Iridium pilot	$1.3	$39
5 MB	Vt idirect	$8.5	$255
5 MB	Inmarsat (fleet broadband model)	$3.5	$105

5.4.2 Data analytics

The shipping lines transport a huge volume of cargo every day and one single delivery requires multiple organization communications for this operation. Multiple operations during the shipping process generate a massive amount of data that includes shipping time and cost, freight information and information of vessels. Therefore, there are immense possibilities for data analytics in this industry and with proper knowledge and applications this can bring new insights. Valuable insights can help the maritime industry to improve operations, asset utilizations and reduce the cost and time. For this reason, it is important for the shipping lines to comprehend how data and data analytics knowledge are crucial.

To understand the benefits from big data, a DNV-GL survey [56] has been reviewed. The survey was conducted in February 2016 and involved 1,189 professionals across industries to understand the value of big data from their business point of view [56].

In this survey, it is mentioned that 52 percent see big data as an opportunity rather than a threat, however, there are two highlighted shortages in that report. Firstly, between the respondents, only one in four has a clear strategy on big data and secondly only one in four is able to leverage on big data to boost productivity and value creation. An in-depth view of big data benefits is presented in *Table 21*.

Table 21 Benefits from big data [56]

Increased efficiency	**23%**
Better business decision making	**16%**
Improved customer experience and engagement	**16%**
achieved financial savings	**11%**

In addition to the DNV-GL survey, Global Marine Technology Trends 2030 (GMTT) published a report in November 2015 and examined 56 critical technologies that might possibly be developed and implemented around 2030 by the commercial ships, naval and ocean sectors [57]. Then, they selected 18 generic technologies for further analysis in which one of them is big data analytic. It is mentioned in the GMTT report that big data analytics is the main key to have the interrelationship model between all three sectors [57].

Thus, data analytics knowledge can support the shipping lines to create value from data. Value can be details information that can be used not only by crew, also by decision-maker to enhance shipping operations. As a result, data analytics skills can facilitate the transformation of raw data into actionable information.

Chapter 6

Conclusion

In this thesis, the rudimentary steps toward a remote cargo application were discussed. The goal of this study was to find and explore the specification of this application according to the maritime environment. To do that, the current communication ways in the shipping industry along with their usage were studied. Then, we reviewed similar industrial solutions to obtain the holistic knowledge as well as requirements for a remote cargo monitoring system from the different perspectives.

The characteristics of IoT in a general concept as well as comparison between common IoT protocols and their advantages and disadvantages were studied. As a result, MQTT and CoAP were proposed and discussed as a publish/subscribe architecture for communication between devices.

I found that the lack of internet access and expensive infrastructure in this industry are the major challenges and due to that, in this thesis fog computing was proposed for ship-to-shore data transfer. In this proposal, there are two main layers, the first layer is the ship layer and it is responsible to collect sensors' data and store it to the bridge, and the second layer, which is the shore layer, should transfer only necessary information with respect to customers. As a consequence, ship operators will have a real-time monitoring system onboard the vessel and cargo owners will be updated about their goods' status.

In future, the growth of communication and especially satellite technologies will provide more accessibility to information from the shore. However, regulators should set standards on the new challenges that will be introduced in the near future, for instance data policy and privacy, cyber security and data management, and hardware requirements onboard a ship etc.

To conclude the discussion, in comparison to other transportation industries, the maritime sector lags behind, and it is mostly due to the environmental challenges and lack of clarity. Nevertheless, a remote cargo monitoring system can enhance the transparency and visibility to the operation and consequently helps the maritime sector.

BIBLIOGRAPHY

[1] S. c. DNVGL, "Ship connectivity," [Online]. Available: https://www.dnvgl.com/publications/ship-connectivity-28107. [Accessed 04 February 2019].

[2] World Shipping Council, "World Shipping Council," [Online]. Available: https://gcaptain.com/number-of-containers-lost-at-sea-falling-survey-shows/. [Accessed 08 April 2019].

[3] K. K. Rob Bradenham, "Bringing the industrial internet to the maine industry and ships into the cloud," [Online]. Available: https://www.hel.fi/static/kanslia/elo/bringing-the-industrial-internet-to-the-marine-industry-and-ships-into-the-cloud.pdf. [Accessed 23 January 2019].

[4] [Online]. Available: http://www.worldshipping.org/about-the-industry/global-trade/top-50-world-container-ports. [Accessed 23 january 2019].

[5] [Online]. Available: https://www.scmp.com/news/china-insider/article/1427947/chinas-average-internet-speed-highest-shanghai-slowest-qinghai. [Accessed 23 January 2019].

[6] [Online]. Available: https://www.vesseltracker.com/en/ports.html?country=46 . [Accessed 23 January 2019].

[7] port of rotterdam, "Port of Rotterdam teams with IBM Internet of Things to digitize operations," [Online]. Available: https://www.portofrotterdam.com/en/news-and-press-releases/port-of-rotterdam-teams-with-ibm-internet-of-things-to-digitize-operations. [Accessed 23 January 2019].

[8] IBM, "Turning Rotterdam into the “World’s Smartest Port” with IBM Cloud & IoT," [Online]. Available: https://www.ibm.com/blogs/think/2018/01/smart-port-rotterdam/. [Accessed 23 January 2019].

[9] "maersk.com," [Online]. Available: https://www.maersk.com/en/solutions/shipping/ocean-transport/refrigerated-cargo/fruit-and-vegetables. [Accessed 12 February 2019].

[10] United nation, "Goal 12: Ensure sustainable consumption and production patterns," UN, [Online]. Available: https://www.un.org/sustainabledevelopment/sustainable-consumption-production/. [Accessed 4 April 2019].

[11] [Online]. Available: https://blog.bosch-si.com/industry40/container-4-0-smart-transport-high-seas/. [Accessed 12 February 2019].

[12] [Online]. Available: https://www.sigfox.com/en/solutions/sigfox-bubble. [Accessed 12 February 2019].

[13] wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Very-small-aperture_terminal. [Accessed 04 February 2019].

[14] S. G. P. P. A. S. Tushar Jamsutkar, "PLC BASED SYSTEM FOR CONTROLLING AND MONITORING PARAMETERS IN SHIP," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. 3, no. 4, pp. 940-944, April 2014.

[15] ABB, "Programmable Logic Controllers PLCs," [Online]. Available: https://new.abb.com/plc/programmable-logic-controllers-plcs. [Accessed 19 February 2019].

[16] G. S. P. G. R. B. J. H. David Hanes, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press, 2017.

[17] M.-t. Zhou, V. D. Hoang, H. Harada, J. S. Pathmasuntharam, H. Wang, P.-y. Kong, C.-w. Ang, Y. Ge and S. Wen, "TRITON: high-speed maritime wireless mesh network," vol. 20, no. 5, pp. 134-142, 2014.

[18] Micrium, "People Internet vs. Device Internet," [Online]. Available: https://www.micrium.com/iot/internet-protocols/. [Accessed 13 February 2019].

[19] [Online]. Available: https://iot-analytics.com/iot-segments/iot-connectivity/. [Accessed 04 February 2019].

[20] V. Sarafov, "Comparison of IoT Data Protocol Overhead," in *Network Architectures and Services*, 2018.

[21] C. Hunt, TCP/IP Network Administration: Help for Unix System Administrators, O'Reilly, 2002.

[22] S. Suehring, Linux Firewalls: Enhancing Security with nftables and Beyond, Addison-Wesley Professional, 2015.

[23] W. Stallings, "PART OF THE PICTURE: The TCP/IP Communications Architecture," [Online]. Available: https://pdfs.semanticscholar.org/7483/19aeb75f6786ec6b501d35f7b32c62547a74.pdf. [Accessed 27 February 2019].

[24] D. Aditya, "Why HTTP is not enough for the Internet of Things," [Online]. Available: https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things. [Accessed 15 February 2019].

[25] A. I. G. M. Luigi Atzori, "The Internet of Things: A survey," vol. 54, no. 15, pp. 2787-2805, 2010.

[26] P. Duffy, "Cisco," [Online]. Available: https://blogs.cisco.com/digital/beyond-mqtt-a-cisco-view-on-iot-protocols. [Accessed 15 February 2019].

[27] P. C. F. V.-G. J. A.-Z. Vasileios Karagiannis, "A Survey on Application Layer Protocols for the Internet of Things," in *Transaction on IoT and Cloud Computing 2015*, 2015.

[28] "http://embeddedlaboratory.blogspot.com/2018/01/getting-started-with-mqtt-using.html," Embedded Laboratory, 26 January 2018. [Online]. Available: http://embeddedlaboratory.blogspot.com/2018/01/getting-started-with-mqtt-using.html. [Accessed 18 February 2019].

[29] R. G. Andrew Banks. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html. [Accessed 18 February 2019].

[30] MQTT, "Eclipse Mosquitto An open source MQTT broker," [Online]. Available: https://mosquitto.org/. [Accessed 6 April 2019].

[31] Z. Shelby, "CoAP," Internet Engineering Task Force (IETF) , [Online]. Available: https://tools.ietf.org/html/rfc7252. [Accessed 20 Feruary 2019].

[32] X. Chen, "Constrained Application Protocol for Internet of Things," wustl.edu, 2014.

[33] L. D. X. Shancang Li, Securing the Internet of Things, Syngress, 2017.

[34] "MQTToasis," MQTT, [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. [Accessed 25 February 2019].

[35] A. L. M. A. Thamer A. Alghamdi, "Security Analysis of the Constrained Application Protocol in the Internet of Things," in *2nd International Conference on Future Generation Communication Technology*, London, 2013.

[36] T. G. Peter Mell, "The NIST Definition of Cloud Computing," NIST, 2011.

[37] O. L. O. L. L. H.-N. D. ,. W. Z. Hao Wang, "Big Data and Industrial Internet of Things for the Maritime Industry in Northwestern Norway," in *IEEE*, Macao, 2015.

[38] M. P. V. T. Pančo Ristov, "The implementation of cloud computing in shipping companies," *Scientific Journal of Maritime Research,* no. 28, pp. 80-87, 2014.

[39] Cisco, "Cisco and/or its affiliates," 2015. [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. [Accessed 5 March 2019].

[40] C. C. W. Gohar Rahman, "Fog Computing, Applications , Security and Challenges, Review," *International Journal of Engineering & Technology,* vol. 3, no. 7, pp. 1615-1621, 2018.

[41] Cisco, "Cisco Fog Computing Solutions: Unleash the Power of the Internet of Things," 2015. [Online]. Available: [8] [Online]. Available: https://blog.bosch-si.com/industry40/container-4-0-smart-transport-high-seas/. [Accessed 12 February 2019].. [Accessed 6 March 2019].

[42] R. D. L. Mora, "Cisco IOx: Making Fog Real for IoT," cisco, [Online]. Available: https://blogs.cisco.com/digital/cisco-iox-making-fog-real-for-iot. [Accessed 6 March 2019].

[43] Inmarsat, "Industrial IoT on land and at sea," 2018. [Online]. Available: https://safety4sea.com/wp-content/uploads/2018/09/Inmarsat-IoT-on-land-and-at-sea-2018_09.pdf. [Accessed 13 March 2019].

[44] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani and S. U. Khan, "The rise of "big data" on cloud computing: Review and open research issues," *Elsevier,* vol. 47, pp. 98-115, 2015.

[45] DNV-GL, "Big data - The new data reality and industry impact," DNV-GL, 2014.

[46] Maersk, "Slow steaming the full story," [Online]. Available: http://www.maersk.com/Innovation/WorkingWithInnovation/Documents/Slow Steaming - the full story.pdf. [Accessed 19 March 2019].

[47] "Fuel Consumption by Containership Size and Speed," [Online]. Available: https://transportgeography.org/?page_id=5955. [Accessed 19 March 2019].

[48] DNV-GL, "GMDSS AND INTERNAL COMMUNICATION," Det Norske Veritas DNV, Høvik, 2001.

[49] wikipedia, "Container ship," [Online]. Available: https://en.wikipedia.org/wiki/Container_ship. [Accessed 22 March 2019].

[50] "World Sea Temperatures," [Online]. Available: https://www.seatemperature.org/. [Accessed 30 March 2019].

[51] "List of weather records," [Online]. Available: https://en.wikipedia.org/wiki/List_of_weather_records#Europe. [Accessed 30 March 2019].

[52] ipfs, "List of atmospheric pressure records," [Online]. Available: https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/List_of_atmospheric_pressure_records_in_Europe.html. [Accessed 30 March 2019].

[53] marinesatellitesystems, "Internet at sea," [Online]. Available: http://www.marinesatellitesystems.com/index.php?page_id=113. [Accessed 25 March 2019].

[54] Ground control, "The Maritime VSAT Advantage: A cost analysis of VSAT broadband versus L-band pay-per-use service," Ground control.

[55] Global Marine network, "Satellite Internet At Sea: Hardware, Airtime, and Pricing," Global Marine network, [Online]. Available: http://www.globalmarinenet.com/satellite-internet-at-sea-hardware-airtime-and-pricing/. [Accessed 28 February 2020].

[56] DNV-GL, "Are you able to leverage big data to boost your productivity and value creation?," DNV-GL, 2016.

[57] GMTT2030, "Global Marine Technology Trends 2030," August 2015. [Online]. Available: https://eprints.soton.ac.uk/388628/1/GMTT2030.pdf. [Accessed 25 March 2019].

[58] S. Mneimneh, "Computer Networks UDP and TCP," Hunter College of CUNY, New York.

[59] P. C. F. V.-G. J. A.-Z. Vasileios Karagiannis, "A Survey on Application Layer Protocols for the Internet of Things," in *Transaction on IoT and Cloud Computing 2015*, 2015.

Appendix I

To download and install Python client.

```
pip install paho-mqtt
```

A sample code that subscribers to the broker $SYS topic and prints out the resulting

```
import paho.mqtt.client as mqtt

# The callback for when the client receives a CONNACK response from the server.
def on_connect(client, userdata, rc):
    print("Connected with result code "+str(rc))
    # Subscribing in on_connect() means that if we lose the connection and
    # reconnect then subscriptions will be renewed.
    client.subscribe("$SYS/#")

# The callback for when a PUBLISH message is received from the server.
def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("iot.eclipse.org", 1883, 60)

# Blocking call that processes network traffic, dispatches callbacks and
# handles reconnecting.
# Other loop*() functions are available that give a threaded interface and a
# manual interface.
client.loop_forever()
```