Creating Decentralised SMART Manufacturing Units with a Newly Implemented Standard Communication Protocol

by
Gareth A. Gericke

Dissertation submitted in fulfilment of the requirements for the degree

MASTER of ENGINEERING in ELECTRICAL ENGINEERING (MEng.)

in the

Department of Electrical, Electronic and Computer Systems Engineering

of the

Faculty of Engineering, Built Environment and Information Technology

at the

Central University of Technology, Free State

Supervisors:

Dr R.B. Kuriakose

Prof. H.J. Vermaak

Prof. O. Madsen

Bloemfontein

2021

© Central University of Technology, Free State

Declaration

I, Gareth Andrew Gericke (Identity Number: [redacted], Student Number [redacted]) do hereby declare that this research project which has been submitted to the Central University of Technology, Free State for the degree Master of Engineering: Electrical Engineering, is my own independent work; and complies with the Code of Academic Integrity, as well as other relevant policies, procedures, rules and regulations of the Central University of Technology, Free State; and has not been submitted before by any person in fulfilment (or partial fulfilment) of the requirements for the attainment of any qualification.

Signed: ____________________ Date: 04/03/2021

© Central University of Technology, Free State

Acknowledgements

I wish to extend my sincere gratitude to the following persons for their contributions to the completion of this study.

Thank you to Dr Rangith Baby Kuriakose, Professor Herman Jacobus Vermaak, and Professor Ole Madsen, my supervisors, for their willingness to guide and motivate me though this trying time the world has experienced amidst the COVID-19 pandemic. Their support through all hours of the days and weeks gave me the perseverance to produce this study that I hold close to my heart. I truly believe the experience and knowledge gained throughout these short years have produced a master's study any old scientific beliefs would deem worthy. I could not have completed this study without your help.

I would like to thank the Central University of Technology, Free State (CUT) for providing me with financial support to prepare me for undertaking this study by attending workshops, and allowing me to attend conferences (for the first time in my life) that assisted me in gaining further knowledge on the study.

I would like to thank the Center for Sustainable SMART Cities (CSSC) and all of the colleagues during the time of this study, for the effort and willingness to help in any part, no matter how small. Their response to my call for help returns a gratitude unlike any other.

To my parents, Mr. Kevin Michael Gericke (deceased) and Ms. Stacey Gericke, and my brother Marcuse Wade Gericke, I would like to express my honest and warmest heartfelt gratitude for your constant support and recompense of your love. My dedication to pursue this level of study could not have been accomplished if it were not for their willingness to care for me through the most difficult of times.

Finally, I would like to thank in a special way my significant other, Janique Jade Dean. Your constant and unselfish acts plainly saved my life through gripping medical scares and mental support. You amaze me every day through our kind heart and loving personality.

A very big thank you to all.

Abstract

Communication protocols drive the flow of information from machine to machine. However, with the requirements of the Fourth Industrial Revolution and machines set to become more mobile, flexible and intelligent, greater communication practices are required to fully incorporate this revolution. Machines are required less to be able to communicate in fixed networks with numerical representation, but rather to have the ability to communicate with machines swopping in and out of manufacturing lines and networks, all while adhering to real-time communication regulations. Furthermore, with machines swapping in and out, a greater need is placed on this communication protocol to operate in a purely decentralised manner, something not available on the market today. All these requirements aid each other. While machines are able to become decentralised and self-efficient, they rely on more crucial information, which is used to create intelligent decisions and execute them with timely precision. This dissertation entails an extensive research in communication protocols and their use in the manufacturing scene. From this, a newly developed communication protocol is designed with a heavy focus placed on real-time execution in a decentralised manner. The communication protocol is tested in a constructed water bottling plant to gauge its effectiveness and performance. Finally, conclusions are drawn from these results, detailing and elaborating the use of the communication protocol and the practices it ensues.

© Central University of Technology, Free State

Table of Contents

List of Figures

© Central University of Technology, Free State

List of Tables

© Central University of Technology, Free State

List of Equations

© Central University of Technology, Free State

Chapter 1:

Introduction into Industry 4.0

1.1. Background

Industry 4.0 [1] (I.4.0) has established itself as the latest trend in manufacturing. The four pillars [2] of I 4.0 are Cyber Physical Systems (CPS), Internet of Things (IoT), Internet of Services (IoS) and SMART Manufacturing (SM). However, the development in this field has been rapid and as a result left gaps, especially in defining standardised communication protocols between the different components of SMART Manufacturing.

SMART Manufacturing [3] is concerned with the machines or factory working seamlessly to increase productivity and efficiency. If SMART Manufacturing Units can work independently of central servers, then the key for decision making becomes the necessary task not only with the SMART Manufacturing Unit (SMU) itself, but with all other attached machines in the production line. Such decision making can only be achieved with direct communication between machines.

Directional communication between SMART Manufacturing Unit's (SMUs) is referred [4] to as decentralised operation. Many researchers have set out to establish what would classify as SMART manufacturing and cloud manufacturing in other ways [5], and this could prove the use of their own protocol for standardising the communication between SMUs.

However, there remains a lack of research done in this field to lay a solid foundation for other projects to classify as SMART and standardise the process for communication between SMUs. The challenge is accentuated by the ever-growing complexity that a SMART Manufacturing Unit [6] should possess in order to complete production tasks of Industry 4.0.

1.2. Problem Statement

The fast pace of development in Industry 4.0 has resulted in several gaps, especially with regards to communication between SMART Manufacturing Units. This is characterised by the lack of scientific literature and experimentation, specifically on developing decentralised communication protocols between one or more Smart Manufacturing Units [7]. Communication protocols currently in use [8], although offering robustness between different machine vendors, suffer in the inability to realise real-time communication necessary for Industry 4.0 - SMART Manufacturing.

1.3. Research Hypothesis and Objectives

1.3.1. Research Hypothesis

On developing a decentralised communication protocol between Smart Manufacturing Units, it is envisioned that SMUs will have the ability [9] to be scalable, modular and have context awareness. This will allow these units to be adaptable, flexible, integratable and decentralised. The newly developed communication protocol could also be used for future benchmarking of Smart Manufacturing Units. Identifying the best means of creating a decentralised communication protocol between machines can be approached through the interoperability characteristic of the machines themselves, as expressed in Figure 1.

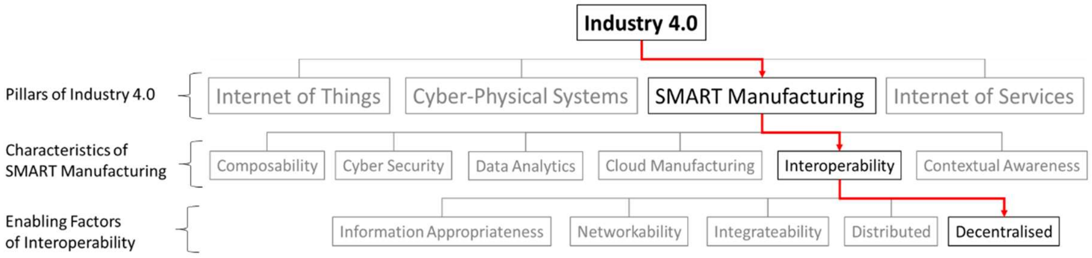

Figure 1: Conceptual Framework

1.3.2. Research Aim

A SMART Manufacturing Unit (SMU) needs to possess certain characteristics [10] to be classified as "SMART". These characteristics include context awareness, interoperability, cloud manufacturing, data analytics, cyber-security and composability. The aim of this research is to develop a communication protocol between Smart Manufacturing Units to decentralise them from one another and central servers. With this protocol in place, decentralised SMART Manufacturing Units will be realised. The SMUs will also be making informed decisions by themselves, by determining the type of medium used to transmit, and the message to be transmitted having a decreased latency before communication, lower transit time during communication and allowing for an effective exchange of information between machines for SMART decision making. The lower latency and transit time, along with the suitable communication method will enable SMART manufacturing with real-time communication.

1.3.3. Research Questions

- What is the classification of real-time communication for machine-to-machine communication?

- Where would the best location for a communication protocol lie, within all network layers, to achieve real-time and decentralised communication?

- What are the classifications of decentralised communication?

- What are the advantages of decentralised communication?

1.3.4. Research Objectives

- Perform a literature review on decentralized and centralized SMART Manufacturing.
- Develop a case study on a water bottling plant
- Develop a case study analysing decentralised and centralised SMART Manufacturing
- Create SMART Manufacturing Units for a water bottling plant
- Communicate the SMART Manufacturing Units through an Open Platform Communications United Architecture (OPC-UA) server for centralised communication
- Create a decentralised communication protocol between SMUs
- Create SMUs to work independently, together with other SMUs and in cohesion with a cloud server
- Analyse the latency and transit time of the developed decentralised communication protocol between SMUs
- Compare the protocol between SMUs against individual SMUs to a central server and cloud server in terms of transit time and latency
- Analyse the decentralised aspect of SMUs

1.4. Demarcation

This study is limited to explore and evaluate the decentralised aspect of the SMART Manufacturing Units with a new implemented communication protocol. Although other aspects such as contextual awareness, data analytics and cloud manufacturing will be incorporated into the SMART Manufacturing Units to enable the units to create SMART and informed decisions, these aspects will not be evaluated.

© Central University of Technology, Free State

Production with SMART Manufacturing Units will be benchmarked, whereby the machines will use different means of communication between each other, without changing the units and/or their coding. Although different communication mediums could be used, the latency and transit time will be evaluated over an Ethernet connection across all communication protocols in testing. Although SMART Manufacturing does avail itself to customisable designs, a water bottling plant will be used in manufacturing evaluation.

In this water bottling plant, a set limit is used for the type of bottles used for filling, capping and packaging, as to fully evaluate the communication protocol in a set environment. The decentralised characteristic can then be evaluated to see how the machines respond to changes in the production line, and what decisions are taken to counter a shift in production.

1.5. Layout of Dissertation

Chapter 1: Introduction into Industry 4.0: This chapter gives an introduction of the research problem; contains the problem statement and hypothesis; and highlights the objectives and research methodology of the project.

Chapter 2: Literature Review of Industry 4.0, SMART Manufacturing and Communication Protocols: This is a literature study dedicated to acquire preliminary knowledge in all four aspects of Industry 4.0, and the communication protocols used in SMART Manufacturing and SMART Manufacturing characteristics.

Chapter 3: Methodology: This chapter reveals the methodologies undertaken to develop the system under study. These methodologies include identifying the system hardware and software components, a description of how to implement these components, and how these components integrate together. All considered, this chapter provides the methods to develop the system in its entirety.

Chapter 4: Decentralised Communication Protocol Performance: This chapter identifies the tests that needs to be performed to verify the system. It also identifies the setup of each test, the procedures to perform each test, and what results to expect.

Chapter 5: Discussion of Decentralised Communication Protocol: A concluding chapter is presented on the discussion on the results obtained from the tests done. It provides analysis on each test by comparing expected to obtained results; identifies corrections and improvements in each case; and includes additional results obtained during the project which were not part of the testing procedures.

Chapter 2:

Literature Review of Industry 4.0, SMART Manufacturing and Communication Protocols

2.1. Introduction

This chapter consists of the literature review undertaken prior to this study. The aim of this chapter is to establish the research gap which motivated this study. An introduction of Industry 4.0 main pillars [11] is provided to highlight the differences between these pillars, with a deeper focus on subheadings for SMART Manufacturing [12], namely SMART Manufacturing Characteristics [13] and Communication Protocols [14].

2.2. Industry 4.0

Industrial revolutions [15] started occurring around the mid-18th century with the First Industrial Revolution [16] involving production evolution into new manufacturing processes with the introduction of steam- and water-powered mechanics.

The second Industrial Revolution [17] then came about during the late 19th century with a popularity in business seeking out mass production, and assembly lines with the aid of gas, petrol and electric powered engines and motors, with a need for mass production still being in use for its low cost and high production rates.

The Third Industrial Revolution [18] appearing in the late 20th century with Computer Automation and Programmable Logic Controllers automating much of the Second Industrial Revolution with a very tight coding design, reliant on timing and switching circuits with little room of flexibility. The Forth Industrial Revolution [19], debated to debut from 2020 – 2030 [20], is the age of data-controlled production lines, where more information is being collected from machines, products and users to further streamline production.

The Fourth Industrial Revolution introduces with it a means for self-controlled production lines with thousands of connected networks communicating for efficiency and improvement. With this newest revolution a greater need for processing and control is to be imbedded in production lines, machines and the factories they involve.

2.2.1. Internet of Things

Internet of Things (IoT) [21] is the interconnection and communication between internet-enabled devices sharing information [22] between each other. Such sharing of information is usually for the purpose of data analytics or direct control. For instance, if a homeowner would want to control the electricity usage of their home, he/she could attach a microcontroller to the lights at their home light dependant resistors.

The homeowner could then manually turn on/off the lights at their home with their smart phone communicating to the microcontroller on the local Wi-Fi connection. A daylight time setting could also be created to turn on the lights during the night, and off again during the day. The key feature is that the homeowner is able to monitor the status of the lights at their home and have direct control of the lights over an internet connection.

2.2.2. Cyber-Physical Systems

Cyber-Physical systems (CPS) [23] are a combination of a physical machine with a fully replicated digital counter-part, but what sets CPS apart from simulations is the role of the digital counter-part in controlling [24], automating or analysing data of a process in real time. CPS differ from IoT in that IoT would normally comprise of more than one physical component in a network, whereas CPS may comprise of one physical component with a digital counter-part. CPS can be expanded with multiple physical and digital counterparts working in a network, with communication occurring in physical machine to physical machine, physical machine to digital twin, physical machine to further digital counterparts, and digital counter-part to digital counter-part.

2.2.3. SMART Manufacturing

SMART Manufacturing (SM) [25] is the trend of using computer-integrated manufacturing to create fully customisable production orders with integration of adaptability and rapid design changes. This invokes SMART Manufacturing Systems to have SMART/intelligent decision-making [26] functionality programmed in the factory.

SMART decision making becomes apparent with products and the factories being layered with sensor technology to gather data during production. Such data are larger than binary decision values and require data analysis to decide upon the needed course of action for production processing.

2.2.4. Internet of Services

Internet of Services (IoS) [27], although similar to IoT in the sense of having internet-enabled devices communicating to one another, differs in that there is usually a monetary service being offered that is regularly updated, and although more complex, an IoT application can usually be offered to larger groups of people. In the example of self-driving cars, manufacturers can offer web-based services for tracking, driving and GPS services for their cars that are regularly updated to provide improved intelligence and performance. These updates are of a scheduled payment rather than a once-off purchase, as is the case with IoT.

2.3. SMART Manufacturing

2.3.1. Introduction

As expressed previously in 2.2.3, SMART Manufacturing is concerned with introducing highly flexible and adaptive manufacturing concepts in single production line concepts, in order to avail the production line to complex and customer-changing specific orders [28]. With the introduction of sensors in the production line and on products, a full evaluation of the production line takes place.

However, for the machines to understand how to complete orders in a changing scenario, they need to possess some sort of intelligence to make SMART decisions during production. With machines needed to create intelligent decisions with attached sensors all through the production line, it will involve huge amounts of input and output data at each station of the production line.

This is devised as big data and regarded as one of the main challenges of SMART Manufacturing. However, with certain design characteristics and network topologies, these can be realised effectively and in real time. These characteristics are outlined next with an introduction to network communication to follow.

2.3.2. SMART Manufacturing Units

SMART Manufacturing Units (SMUs) [29] are systems that are able to process decisions and be utilised in Industry 4.0 (I.40). Industry 4.0 demands that systems be more integrated and co-operative with each other, to for product customisation and better decision making in production [30].

This study focuses on how a SMU is able to achieve the afore-mentioned attributes, and what some of the challenges experienced currently are [31]. The characteristics of a typical SMU and the communication protocols that govern the transmission of information from cloud servers to SMUs are initially defined. The characteristics [32] of a typical SMU are covered in the next section.

2.3.3. SMART Manufacturing Characteristics

SMART Manufacturing can be created from importing SMART characteristics to machines, enabling SMART Manufacturing Units (SMU). The characteristics to classify a machine as “SMART” include, but are not limited to, context awareness, modularity, scalability, autonomy adaptability, robust, flexible, interoperability, sustainability, decentralised proactivity and integratability.

Context Awareness: Defined as the ability of machines to have a digital presence broadcasted and received by other machines. [33] Such information can include the status of a machine. By broadcasting the status of a machine, other machines are able to sense this presence and make a decision based on its surroundings.

With the sensing of surrounding machines, other machines are able to decide whether to stop, slow down or speed up production or even request the start of production. If a machine is able to assess its own status, it is only needed to broadcast the state of itself and not the entire information needed to arrive at the status of machines.

Modularity: Occurs when a machine is able to have any parts swapped and replaced. This includes extensions of the machine where new external fixtures can be attached to the machine. [34] The controller of the machine should also be modular to allow for easy maintenance or reproduction of the same machine.

Scalability: The ability of SMU to have new components attached onto it to allow for extra functionality or increase in production size. Scalability can also extend into other SMUs where multiple machines can be attached together, in series or in parallel. [35]

Autonomy: Allows SMUs to work independently. SMUs are therefore able to work continuously with constant connections existing between production list servers which an SMU is able to read from, [36] decide on the necessary steps and start production. Current third industrial revolution factories have prioritized autonomy to be their driving aspect for performance and efficiency in today’s world.

Adaptability: The degree to which adaptability can be extended to go hand-in-hand with contextual awareness, scalability and modularity. If a machine is able to sense its surroundings and attached hardware, it is able to intelligently make a decision on what its capabilities are, and to decide if it is able to perform certain actions or productions. [37]

Proactive: The process whereby a machine is able to evaluate itself and help prevent failures. [38] This can be in the form of step verification, where a product in production can be checked at steps to ensure it meets the requirements. This method also aids in quality management. The other method may include predictive maintenance, where a loaded number of uses is put into the system, and a counter system can warn users when it nears the end of its lifetime.

Robustness: Described as machines being well built and have a long-lasting lifetime. This is often a complicated task when one considers that the machine also needs to be modular, scalable and adaptable. [39]

Flexibility: Flexibility [40] is different from adaptability, as a machine is flexible if it is able to work in different areas with different machinery, whilst adaptability refers to the machine's ability to perform different tasks either through new hardware or new software.

Sustainability: Seen as the machine's capability to avoid the usage of natural resources to the point of depletion. This can be in the form of renewable energy supply or even renewable resource usage. Although a relatively easy fixture can turn machines to run off renewable energy, it can be regarded as a management aspect of the machine and not a requirement of construction. [41] Whilst sustainability has been brought into has one of the most in demand needs for factory spaces, solutions in these spaces are often from external sources, with their implementations still being seen worldwide.

Interoperability: Handles all of the SMU's communication. [42] This communication can exist through networkability, but can also include attributes such as decentralisation, integrateability and distribution. Interoperability also handles information appropriateness where only relevant information is sent to receivers that require this information. Although many newer factory spaces have included interoperability as staple factory spaces to monitor production, the true extent needed for Industry 4.0 feels lacking in order to totally capture the potential of future and Smart factories.

Decentralised operation: A key aspect of SMUs is not only the ability to be intelligent, [43] but also the ability to work independently of other machinery. This means that SMUs connected to a central server to handle incoming orders should still be able to work independently of that server once a production list is saved. Thereafter, every SMU should be able to work independently and not need confirmation from the central server to signal events.

System Integration: When systems have the ability to have add-on machinery or equipment attached to them and then use such equipment for production or service means. [44]

For a manufacturing unit to be classified as SMART it does not have to possess all of the above characteristics, rather a majority of these characteristics will suffice. The characteristics are grouped and organised in Figure 2 in their respective categories.

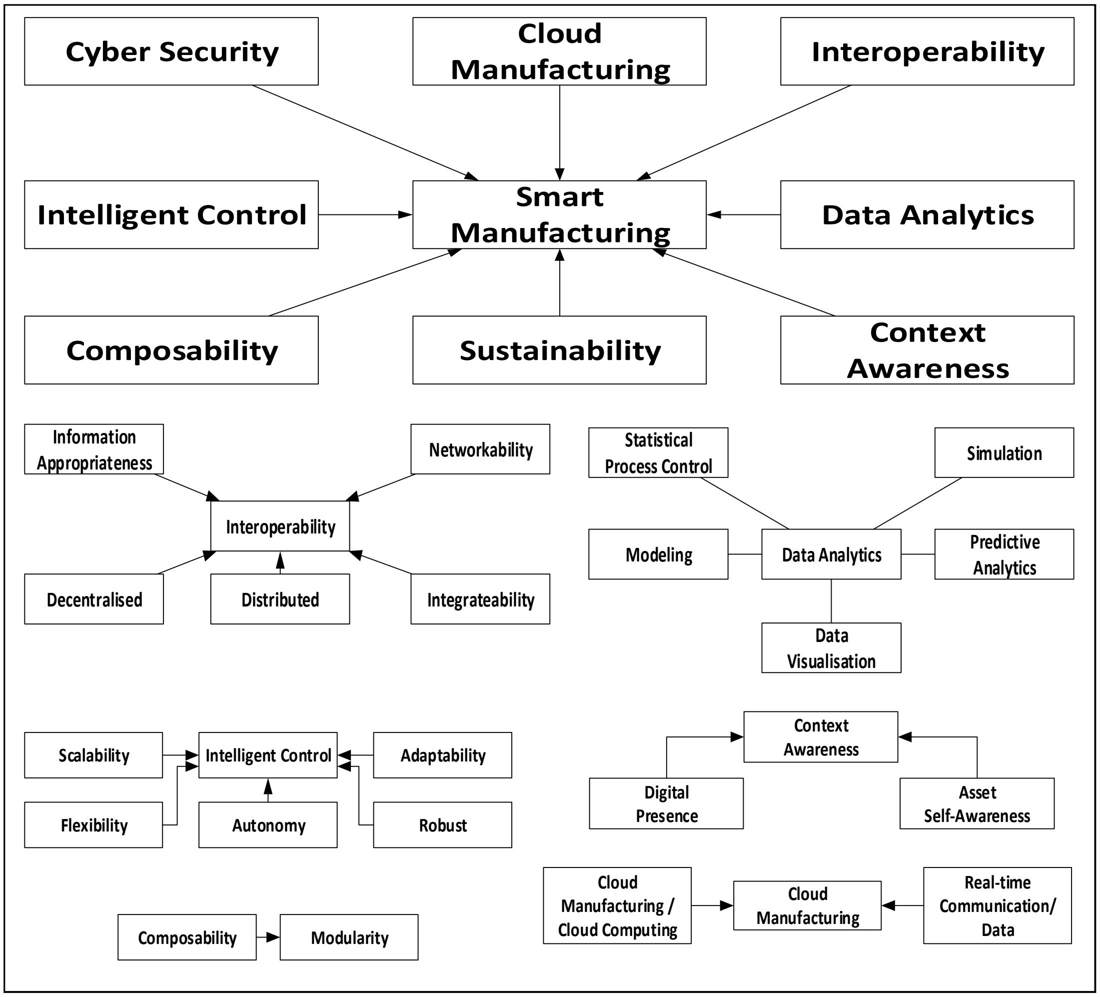

Figure 2: SMART Manufacturing Characteristics [38]

Most SM operations include a network that is controlled by a central server that links between the Cloud Manufacturer and SMUs, as can be seen in Figure 3 as layers 1 and 2, respectively. This results in a few challenges. The first challenge occurs when a single SMU would like to have its information of contextual awareness. In this scenario, the SMU will then need to request information from the central server about surrounding machines. Only after the first SMU has completed its request, can the second SMU submit its request, and so on.

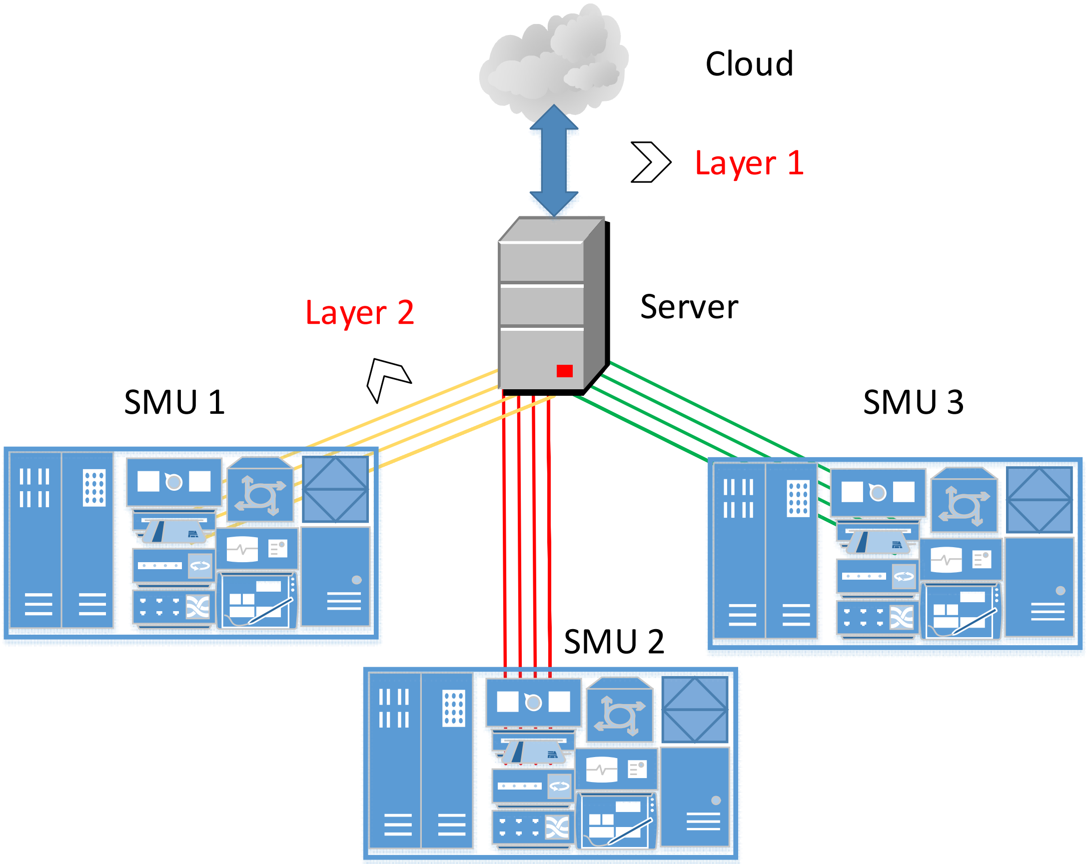

Figure 3: SMART Manufacturing Networks

The queuing of SMUs will increase the number of non-parallel communication requests to the central server delaying information processing. Since this process uses the hardware resources of the single central server and not that of the multiple SMUs processors, a lower byte of information can be sent at once, decreasing the communication time out of real-time communication and removing the context awareness characteristic of the SMUs all together, which is needed to be classified as SMART.

The next challenge is the characteristic of decentralisation. As mentioned previously, an SMU should be able to work independently from the central server and cloud manufacturer. Such decentralisation will only be realised if every machine in a production line is able to communicate with one another, independently of the central server, to complete a production request.

2.4. Communication Protocols

2.4.1. TCP/IP Communication

Historically the first internet connections where those of hard-line peer-to-peer connections, [45] and later different host connections were possible with the automation of telephonic switchboard operators. Host-to-host connections and switch packet networks were realised through these means. This was later coined as TCP/IP [46], which was the combination of the Transmission Control Protocol (TCP) and Internet Protocol (IP).

The TCP/IP protocol integrated redundancy checking, address transmission, information organisation and high transmission rates with requests to resend missed packets. It is important to know how information is sent on requests through a network topology. For this each layer of the communication network must be identified. For general message transmission the common layers, referred to as TCP/IP layers, are used.

2.4.2. TCP/IP Layers

The Cisco TCP/IP layers [47] define how information is sent, generally from web-based applications, and received. There are four layers of the TCP/IP communication, however more modern communication layouts include more layers to include security during transmission or encoding for encapsulation. As seen in Figure 4, the standard TCP/IP layers are shown. These layers show the minimum process required to send information through a scalable topology.

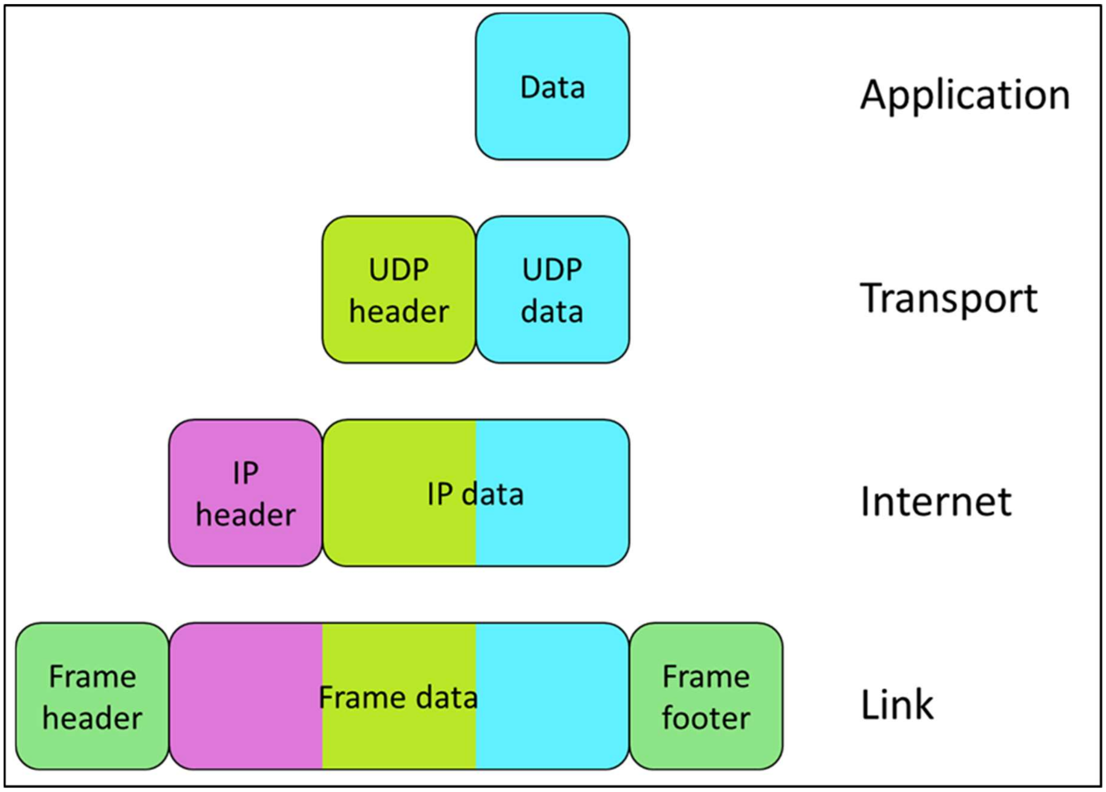

Figure 4: TCP/IP Layers [47]

2.4.2.1. *The Application Layer*

During transmission the Application Layer [48] is responsible for the collection of the data to be transmitted. The application layer also defines the protocol used for the application to exchange data. These protocols can include Telnet, HTTP and DNS.

2.4.2.2. *The Transport Layer*

The Transport Layer [49] is responsible for decoding the source destination in a network. For instance, if PC A wants to send PC B a message, then PC A and PC B would have their own unique address in the network. PC A can then send the message to the connected router, where the router can forward the packet based on the destination address placed by the UDP header. The packet can then be forwarded to PC B without the router needing to decode the packet data. However, the transport layer can also transmit packets form peer-to-peer connections with the aid of session hosting and datagram communication. This is specified by the previous application layer.

2.4.2.3. *The Internet Layer*

The Internet Layer [50] is responsible for the transmission of packets in different networks. The layer then adds on top of the transport layer its own IP header that will allow for routers to forward packets to other connected routers in different networks. Every network has its own unique address range and routers that operate in this layer. It is therefore the responsibility of the internet layer to correctly transmit the packets to the correct network address.

2.4.2.4. *The Link Layer*

The Link Layer [51] consists of the physical components, normally referred to as the topology, of the network. It can consist of switches and routers as well as physical cables or signals being transmitted. Although there is a conception that protocols can exist in this layer, it is a misconception. As with different cable connections, different protocols can be used to trigger signals across links. Similarly, with Bluetooth and Wi-Fi signals needing their own pre-agreed on communication medium, communication protocols govern how a signal can propagate over physical networks.

Therefore, a protocol can exist in any of the four TCP/IP layers, where the protocol can govern data structure, hand-shake methods, transmission time and permissions as well as paths and propagations.

The four different TCP/IP layers can then also be seen as each protocol being an encapsulation of data in the previous layer during transmission. At the receiver's end, each router during the propagation of the packet does not need to decode the data at the application layer, however, the router node only needs to forward the packet based on the frame header from the link layer. Furthermore, only the destination host needs to decode the data of the application layer, saving on overhead during transmission.

2.4.3. TCP/IP Packet Structure

The majority of the internet uses the Internet Protocol version 4 (IPv4) communication protocol [52] for communication. An example of the packet structure is depicted in Figure 5. The IPv4 will sit above the application and transport layer, meaning that information can be sent globally. How each region decodes the information in the IPv4 communication is down to the application, as HTTP requests can have different data structures and check sums.

This form of communication then encapsulates the data from the application and transport layer, and the link layer has been made to conform to the IPv4 protocol. It allows for large scaling of networks with large sets of data.

The data that then needs to be sent from host to host is broken down into the IPv4 packet structure. The packet structure separates the data into individual packets, and all packets are sent over the link bandwidth. If some packets fail to be received, only a specific packet needs to be resent on request.

However, in the case of real-time audio, a certain amount of missing information (dropped packets) during download is acceptable without a noticeable change in quality. Where each packet is specifically of importance, a dropped packet can be resent.

© Central University of Technology, Free State

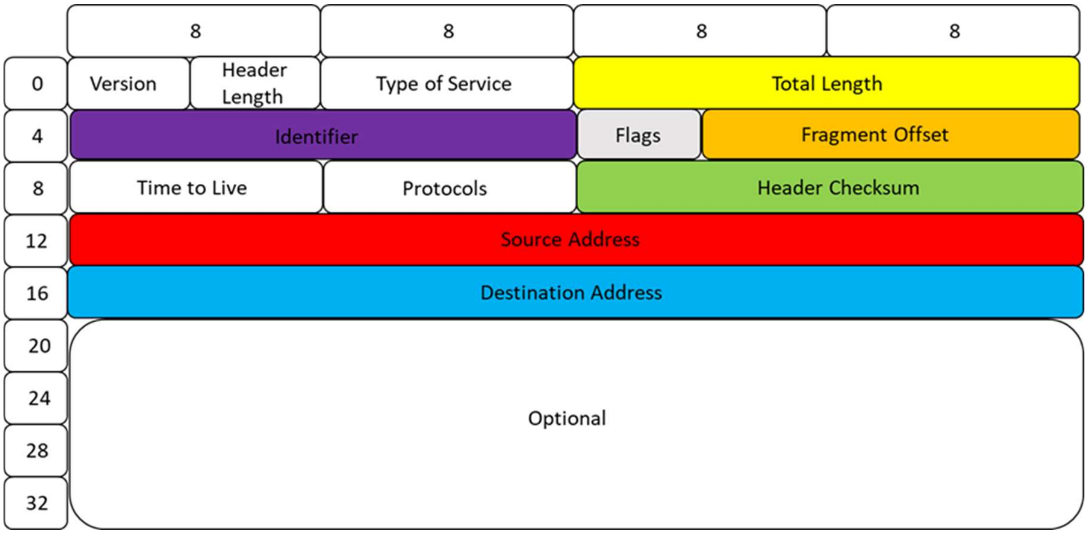

Figure 5: TCP/IP Packet Structure [52]

2.4.3.1. *Bandwidth*

The bandwidth of a link [53] is the amount of information that can be sent at any one time. Generally, routers and nodes limit the amount of bandwidth available per line to evenly distribute bandwidth across all users. An illustration showcasing the shared bandwidth among users can be seen in Figure 6. This also allows for upscaling of bandwidth on certain links if a greater speed is required.

The speed at which packets are able to be send and receive is seen as how much byte information from packets is able to arrive per second. If a faster speed is required, then a larger bandwidth can be allocated, or a faster means of transportation needs to be installed, as more data can be sent per second. A lower bandwidth can result in a lower speed, but it is not viewed as a delay towards the packet.

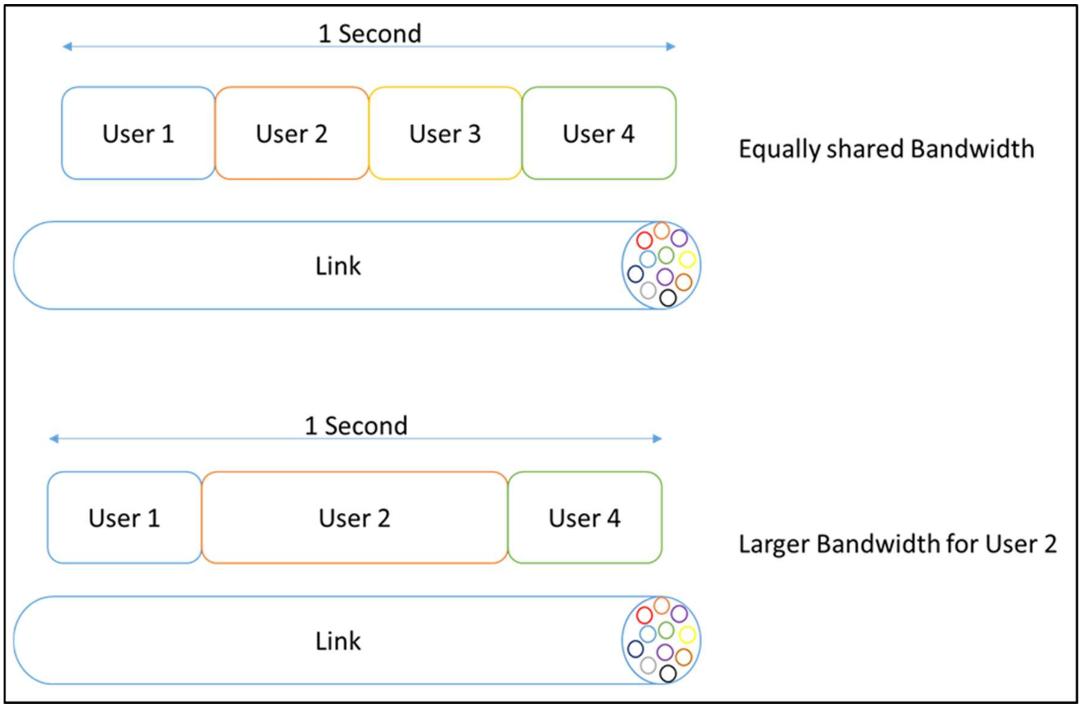

Figure 6: Communication Bandwidth

2.4.3.2. Delays in Messages

2.4.3.2.1. Latency

Before packets are transmitted they can experience a delay described as latency. Latency [54] occurs when the hardware used to send the package experiences a delay before the actual transmission of the message takes place, depicted in Figure 7 as a onetime latency occurrence. This is caused by a lack of sufficient hardware to collect/send the information, a backlog of information that is built up to be sent, the length of cable used to connect the devices, or a too large amount of information to be sent over the hardware efficiently. Latency is not effected by the physical link/connection between the devices, however such a delay is known as transit time delay.

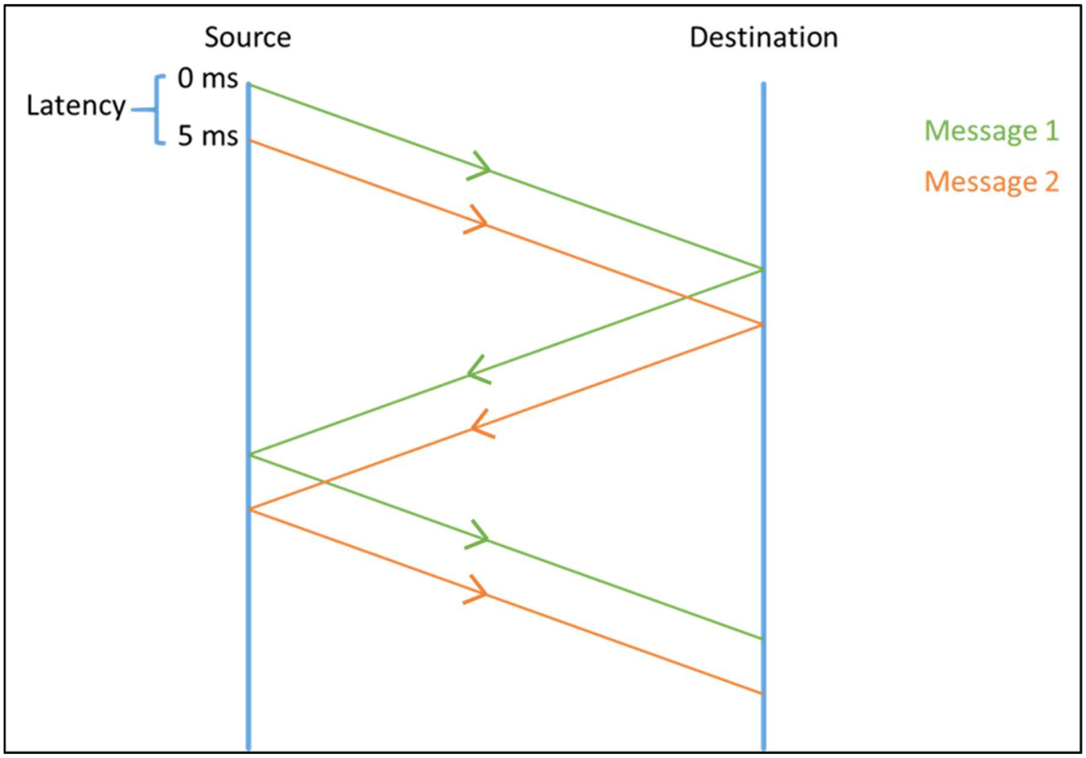

Figure 7: Latency [54]

2.4.3.2.2. Transit Time

Transit time delay [55] occurs when a message takes progressively longer to arrive to the destination, as seen in Figure 8. Since this is not a constant start delay, the message will take longer to arrive due to the speed it travels at. The speed can be influenced by the type of connection used to link the two devices, the protocol used to send information, additional nodes/hardware in the transmission line, or faults occurring in the link or message itself.

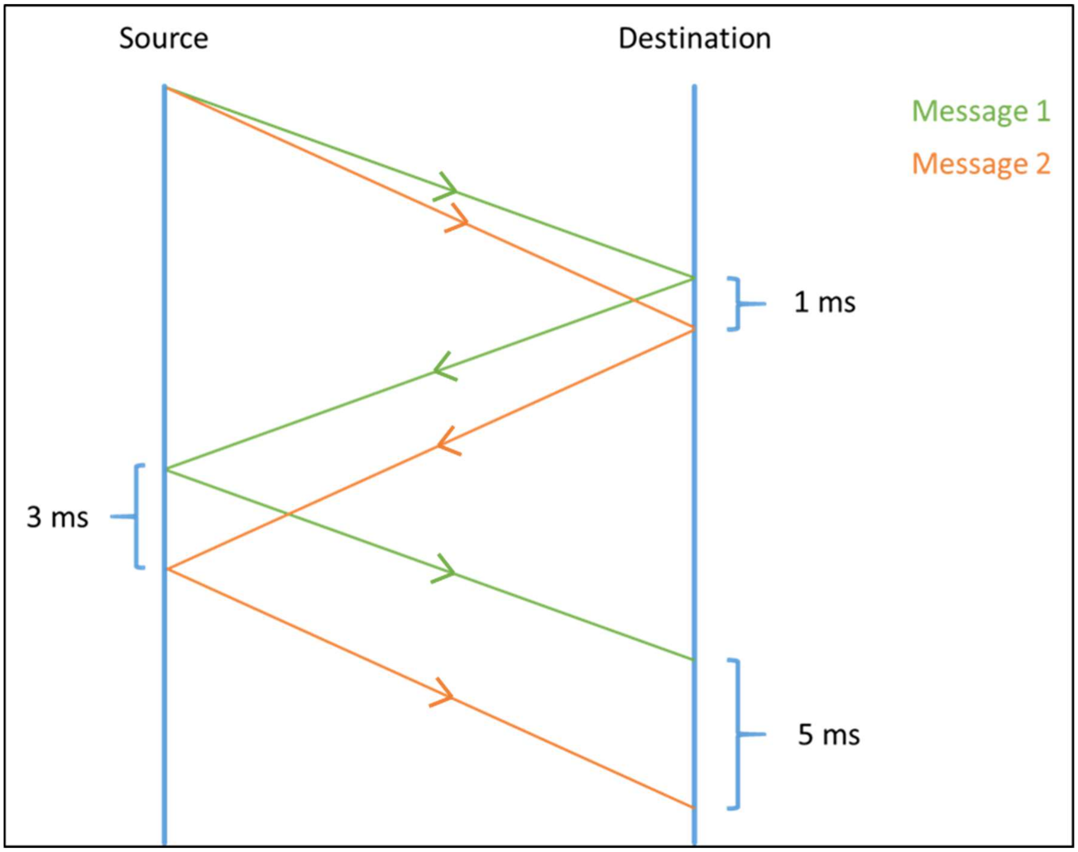

Figure 8: Transit Time Delay [55]

2.4.3.3. *Link Speed Testing*

The most common use of link speed testing [56] is known as a ping test. A ping test is one of the lightest packets able to transmit. The role of the ping test packet is to contain no information except for the source and destination address and the time started to transmit. The ping test packet will then travel from source to destination with the time stamp equipped, after which the destination host can decode the time stamp and calculate the transmission time of the packet.

Such transmission time can then be related back into the speed and bandwidth of the connection link. Since the ping test packet contains little to no information, it is the closest representation of an ideal packet to be sent to evaluate the link speed.

2.4.4. Communication in SMART Manufacturing

Factory machines can use a wide variety of communication protocols to transmit data between machines and to central servers. The communication protocols include, but are not limited to, Profinet [57], Open Platform Communications United Architecture (OPC-UA) [58], HTTP [59] and WebSocket [59]. In the manufacturing industry it is to a company's advantage to create a homogenous environment of machines.

The homogenous factory approach often leads to a vast majority of production and communications to exists, as it is not possible to integrate other manufacturer's equipment with another. However, some companies, such as OPC-UA, allows for multi-communication protocol integration to be used across multiple devices.

There are still multiple layers of communication where each communication protocol may exist. These layers are identified as cloud-to-server communication, server-to-machine communication, and machine-to-machine communication.

Ultimately this study hopes to create a new decentralised communication protocol, and to investigate its performance, with a major emphasis placed on information flow to facilitate intelligent decision making and reduced latency. A brief summary of this is shown in Figure 9.

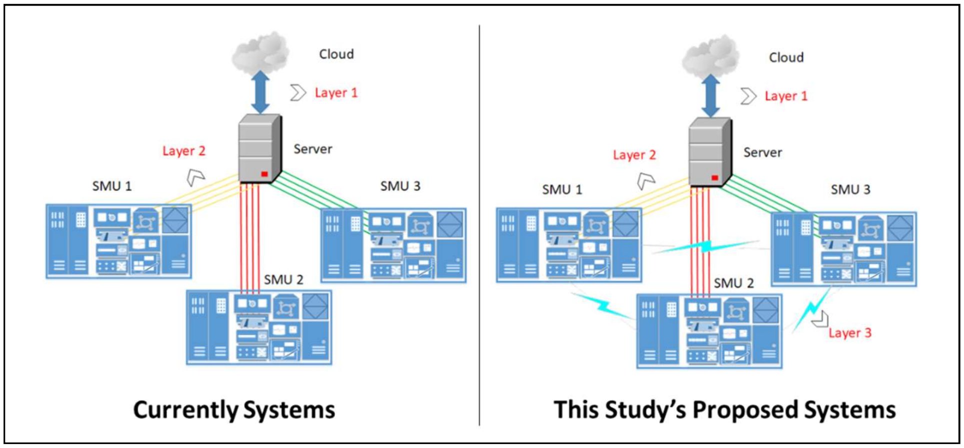

Figure 9: Study Proposal

2.4.4.1. OPC-UA

Open Platform Communication Unified Architecture (OPC-UA) [60] is the most common communication protocol in use for central servers. The reason for its popularity is its robustness to incorporate a variety of communication protocols for vendor-specific communication. It allows a central server to collect information from PLCs, and to store such information in the central server.

This can be done multiple times and from different PLC brands. PLCs can also then read information from the OPC-UA server. It allows different PLC models to communicate indirectly with one another. This is possible due to the OPC-UA server's ability to neutralise the information from a PLC into an information message type, and not a communication-specific message.

The neutralised information message can then be read from any other PLC on the OPC. OPC-UA also allows communication from other equipment such as personal computers, other central servers or web-servers. However, with the OPC-UA server acting as a central server, request queuing may build up, leading to a higher latency call per request, and a larger delay in response.

2.5. Limitation and Conclusion of Current Research

With the current communication protocol used in factories, there are still significant sacrifices that users need to consider when choosing the appropriate communication protocol. These sacrifices are prominent in real-time communication and where there is a lack of decentralisation.

As seen in Figure 10, the overall latency of a communication protocol limits the users in terms of real-time communication (chosen at 120ms). OPC and OPC-UA do offer the closest means of decentralised communication, but at the highest latency. These drawbacks of latency during decentralisation require more research to be done in offering real-time decentralisation in order for Industrial 4.0 to realise its full potential. [61]

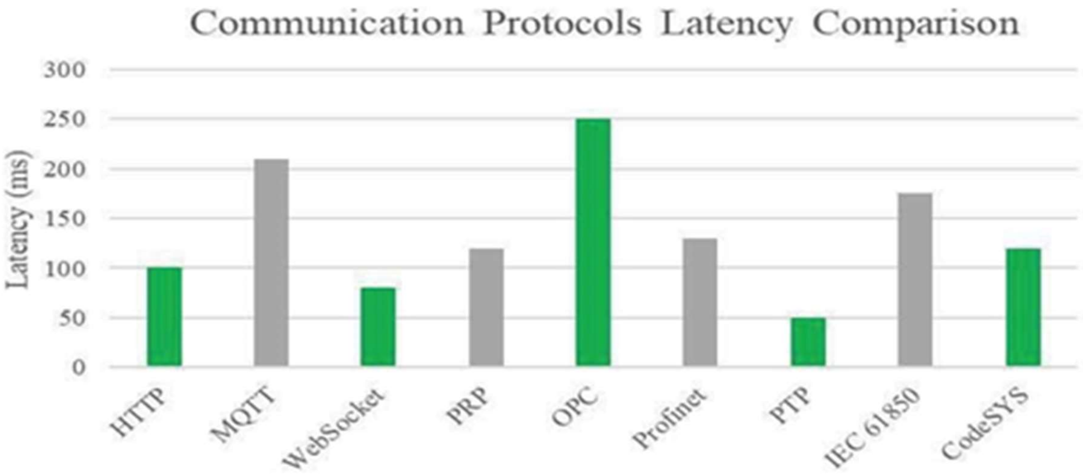

Figure 10: Communication Protocol Latency [61]

Chapter 3:

Methodology and Creation of Decentralised Communication Protocol

---

3.1. Introduction to Manufacturing Units

Manufacturing Units in a factory are utilised for automated responses of completing production tasks in a factory. These tasks are typically monitored at a central server on completion of major milestones. More intricate and precise monitoring is done on individual machines. This is done for two reasons; the first being that larger overseeing completed tasks are usually stored in memory.

With central servers equipped with large-scale hardware resources, the ability to store and compute tasks is completed by central servers with ease. Whereas small tasks require faster and more integrated components to monitor, these tasks are more suited to be monitored by individual machines, where less but faster hardware resources are required with no need for long-term memory storage.

With the evolution of Industry 4.0 becoming more seamless in integration, and a ‘swap and replace’ mentality coming in line with the factory floor, machines will need greater monitoring and information in order to properly fit in the Industry 4.0 needs.

3.2. Manufacturing Process

With Industry 4.0 offering high variety in product customisation, customers are able to specify parameters to make them uniquely distinguishable from other products and competitors. However, this also adds a greater amount of user input data and the need to accommodate such data. With conventional websites offering “pick-and-choose” style product websites, a new method of user input is needed to offer greater customisation options.

One method of achieving this is with users being able to upload special files for design or manufacturing requirements. With reference to Figure 11, customers should be able to specify unique product designs. In the case of the water bottling plant, users are able to upload a stereolithography (.STL) file that allows for unique bottle designs to be three-dimensionally (3D) printed, filled, bottled and packaged to order.

With the unique design pending from customers, it renders the manufacturing process to be flexible and intelligent during production, due to factors such as bottle height, width and volume, to name a few. These factors can change the manufacturing process per order in terms of the amount of water needed to be filled, the height position which needs to be capped, and gripper mechanics. These factors will therefore need to be determined and communicated either before production starts, at the beginning of production, or at every step of the production.

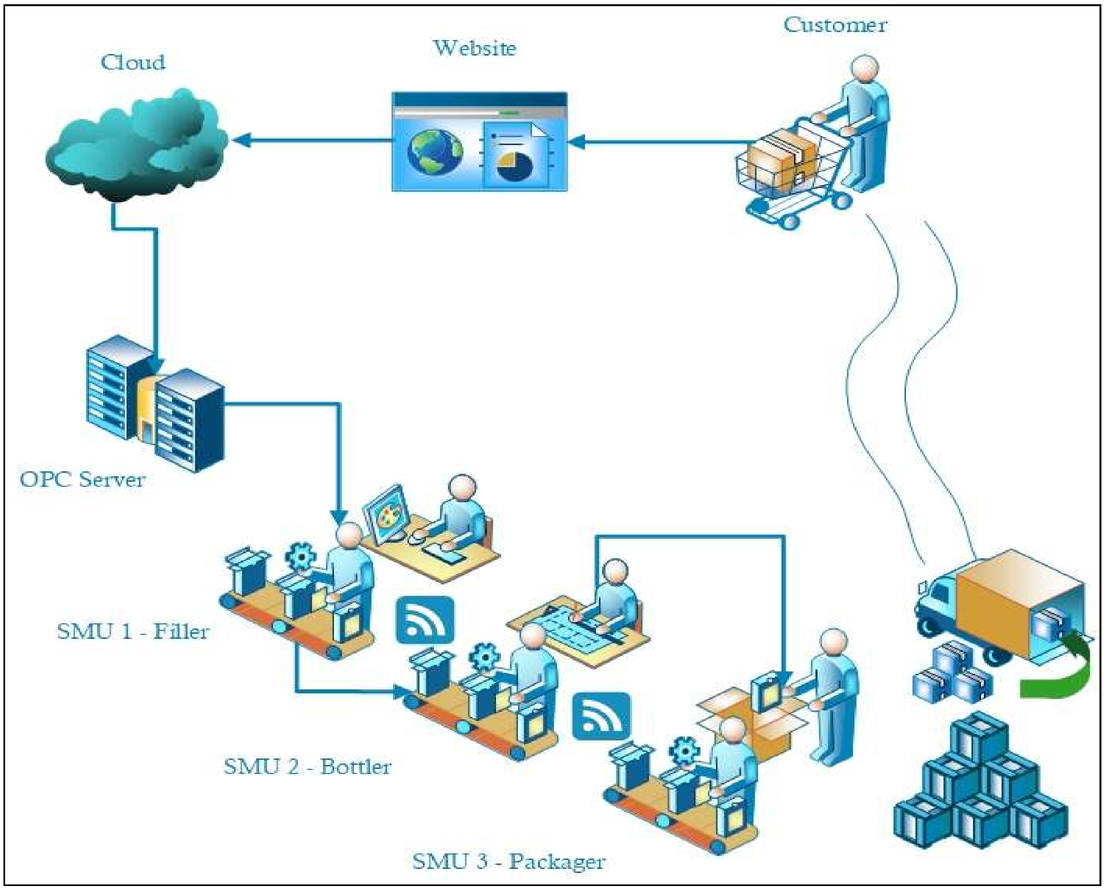

Figure 11: Traditional Manufacturing Process Encapsulated

3.3. Factory Layout

For this project, three SMART Manufacturing Units (SMUs) were used to create a water bottling plant. SMU 1 is tasked with filling the water bottles, SMU 2 with capping them, and SMU 3 with packaging the water bottles per customer orders. An overview of the production line is depicted in Figure 12.

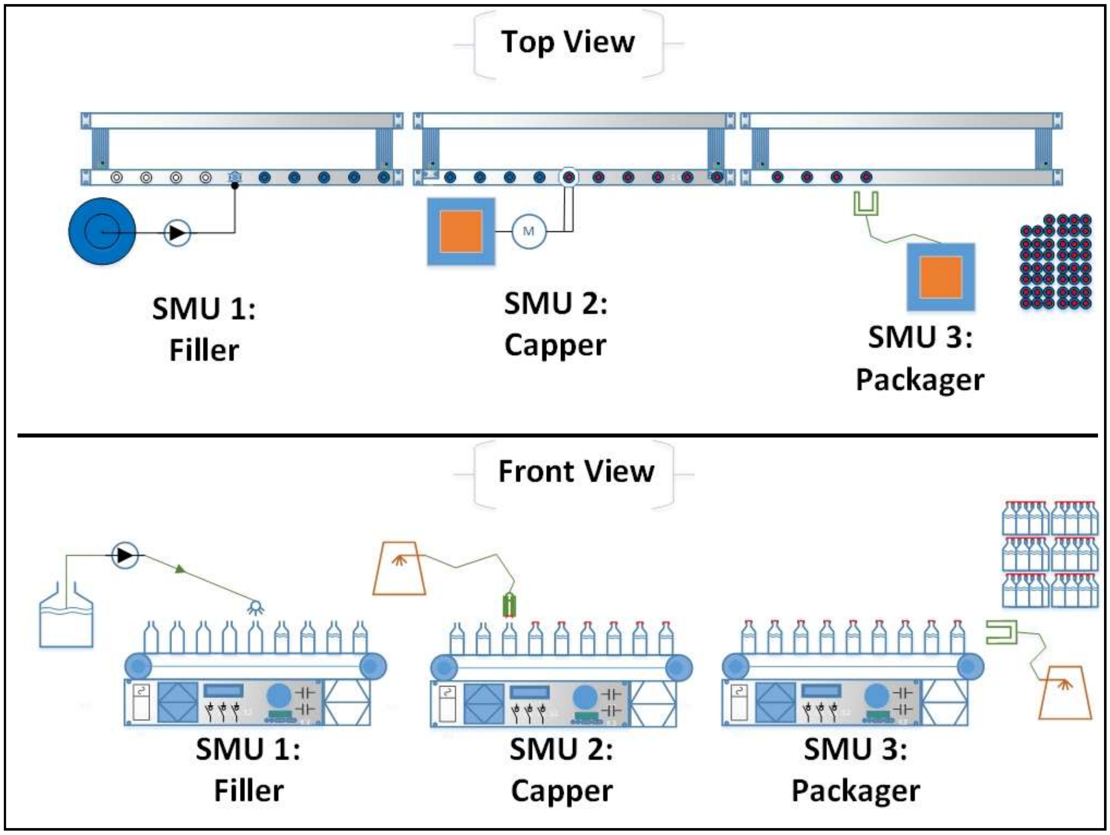

Figure 12: Factory Layout

3.3.1. SMU 1

SMU 1 is tasked with filling water bottles. With users being able to request specific designs and volumes of bottles, the conundrum of determining the correct bottle to fill can be solved in two ways. The orders can be virtually deduced to be communicated to the machines by the central server, or the bottles can be inspected by SMU 1, and can be communicated down the production line. Once the bottle size to be filled is determined, SMU 1 is able to select the correct bottle from storage and to fill it to order.

In this project, the central server will send a production order to fill the bottle that is communicated to each machine and passed through from machine to machine with the newly created decentralized communication protocol designated as the SMART Manufacturing Protocol (SMP). An additional fixture can also be added on top of SMU 1 that allows for unique bottles of different shapes and colours to be added in the production line, filled and positioned to be capped as determined by SMU 1 and communicated down the line.

3.3.2. SMU 2

SMU 2 is tasked with capping water bottles. With the production order being handled by the central server and being communicated to the machines, the height at which the bottle caps are to be attached can be communicated to SMU 2. SMU 2 can then query the central server on bottle heights, and can communicate to SMU 1 to determine the type of bottle and when the bottle is in transmission between SMU 1 and SMU 2. SMU 2 uses a KUKA robotic arm with an attached bottle cap fixture to cap water bottles with homogenous caps.

3.3.3. SMU 3

SMU 3 will package the water bottles to order. Attached to every water bottle is a unique RFID tag that is tracked with production information by each machine. This allows SMU 1 to fill water bottles by order and tag the water bottles. It further allows SMU 3 to read the RFID and stack water bottles by order. In the case that water bottles are filled first by type and not by customer order, SMU 3 will be able to set aside orders and package them as they are received.

3.4. Centralised Manufacturing Units

Traditional manufacturing lines use timely events as a substitute for communication. These timely events cost resources and time, dragging down the overall production efficiency of the manufacturing line. This is due to a duplication of sensors where, traditionally, a larger server is set up to handle sensor inputs and communicate all information to machines where possible.

With today's machines becoming more advanced, the advantages of supplying each machine with its own unique sensors have become more popular. With the introduction of Industry 4.0, each machine should be able to fabricate intelligent decisions based on its sensor data. The machines therefore become their own server, and this allows machines to utilise communication over sensor duplication. Overall this decentralisation differs from traditional centralised manufacturing factories, recaptured in Figure 13.

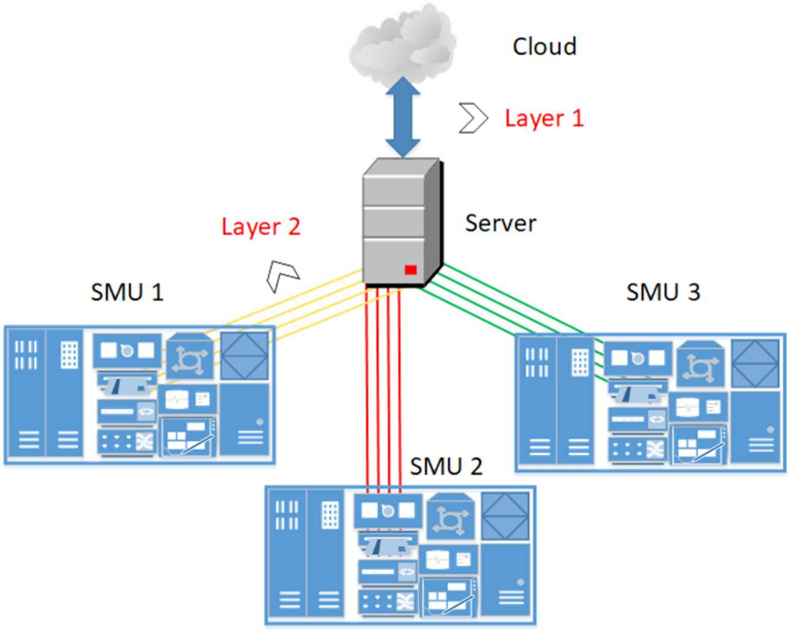

Figure 13: Centralised Manufacturing

3.4.1. Timely Events

Timely events include a variety of sensors such as inductive sensors, load cells, magnetic sensors and proximity sensors to trigger events which are hard programmed into a machines code. These timely events allow for very little flexibility and error to occur in a production line or even single machines. Often at times machines have very basic programming built within them, with only inputs driving outputs with nothing in-between. However, Industry 4.0 demands the use of intelligent decision making at every level. Therefore, simple input data cannot drive output data if product specifications change or obstruction in production occur. A real-life example of a water bottling plant driven by timely events can be seen in Figure 14.

Figure 14: Typical Water Bottling Plant

3.5. Decentralised Manufacturing Units

Decentralised manufacturing units are able to communicate to other machines without the need of facilitation through a central server. However, the type of communication differs, not only pertaining to the method, but to the type of information to be sent. Decentralised communication, in manufacturing, allows machines to communicate with other machines for specific and crucial information. These machines are able to reduce the time to decision making by reducing latency and communication time. There are several topologies that can be considered. Smaller manufacturing lines opt for direct link by the use of redundancy but availability of safety, as in Figure 15.

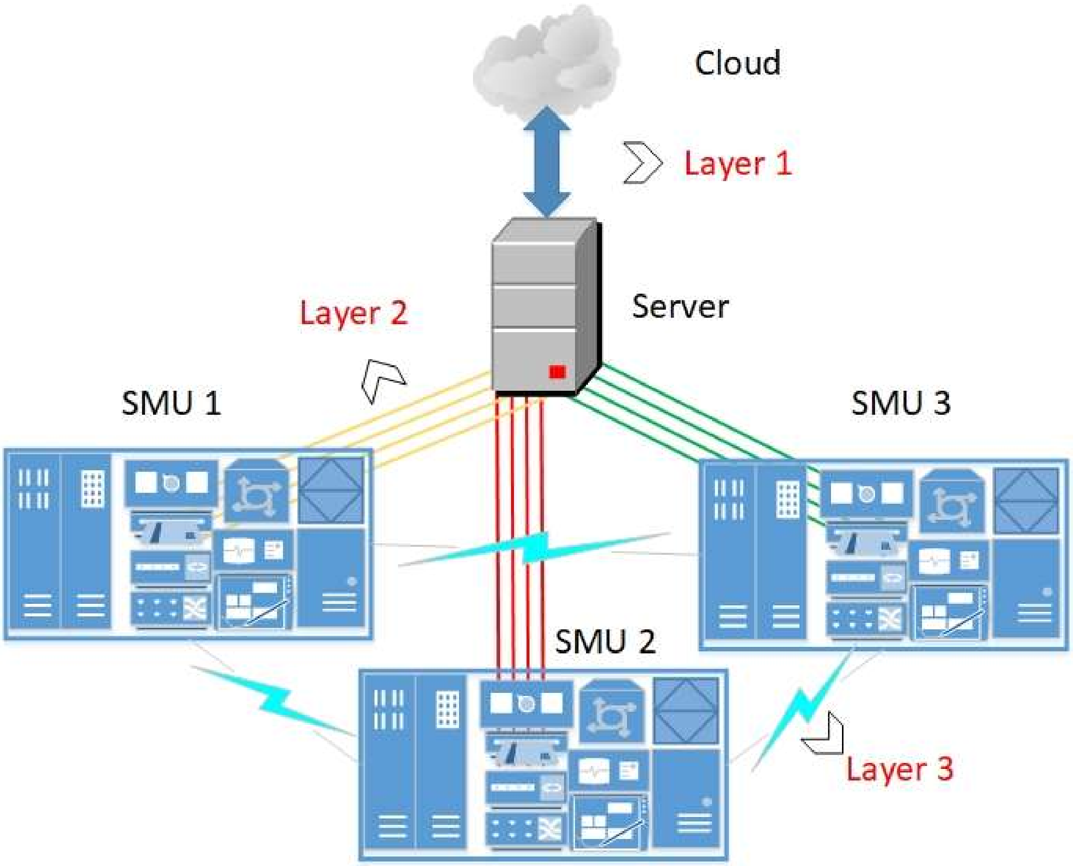

Figure 15: Decentralised Manufacturing

3.6. Decentralised Communication Protocol

A communication protocol is defined as using a throughput of information depending on the type of communication – how to send data, addressing - where to send data, and message fielding - what data to send. All these factors create a communications protocol outlined in the TCP-IP layers.

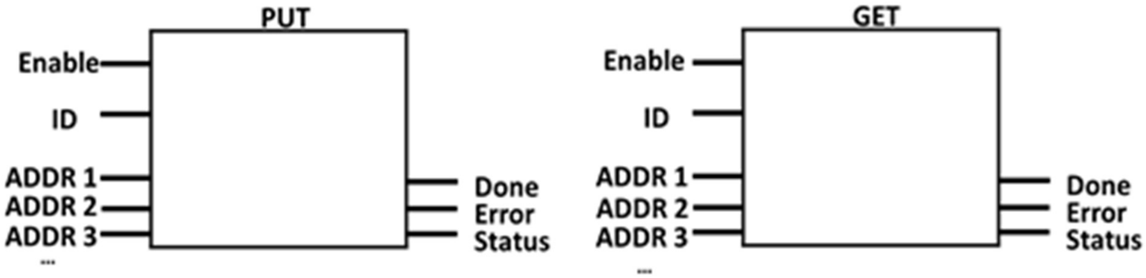

Figure 16: SMART Manufacturing Protocol – GET/PUT Method

3.6.1. Communication

The SMART Manufacturing communication protocol utilises a GET and PUT method, as seen in Figure 16, that allows each communicating component to either send a packet of information to a specific device, or retrieve a certain packet of information from a device. With most communication protocols, such as OPC-UA, a host server would have to request information to be sent, with a confirmation or request to resend if there are errors. For the same payload size, a GET/PUT method does not need a request to send or a confirmation, as seen in Figure 17. These attributes are handled by the receiving information device. Furthermore, a receiving device initiating a GET method would check if the network line is available to send information. Once a line is free to transmit, the receiving device will collect the information with the GET method and perform its own error checking with the checksum footer. These methods therefore reduce the need for the ‘request to send” and “confirmation” traffic, reducing latency, while increasing computation resources on devices.

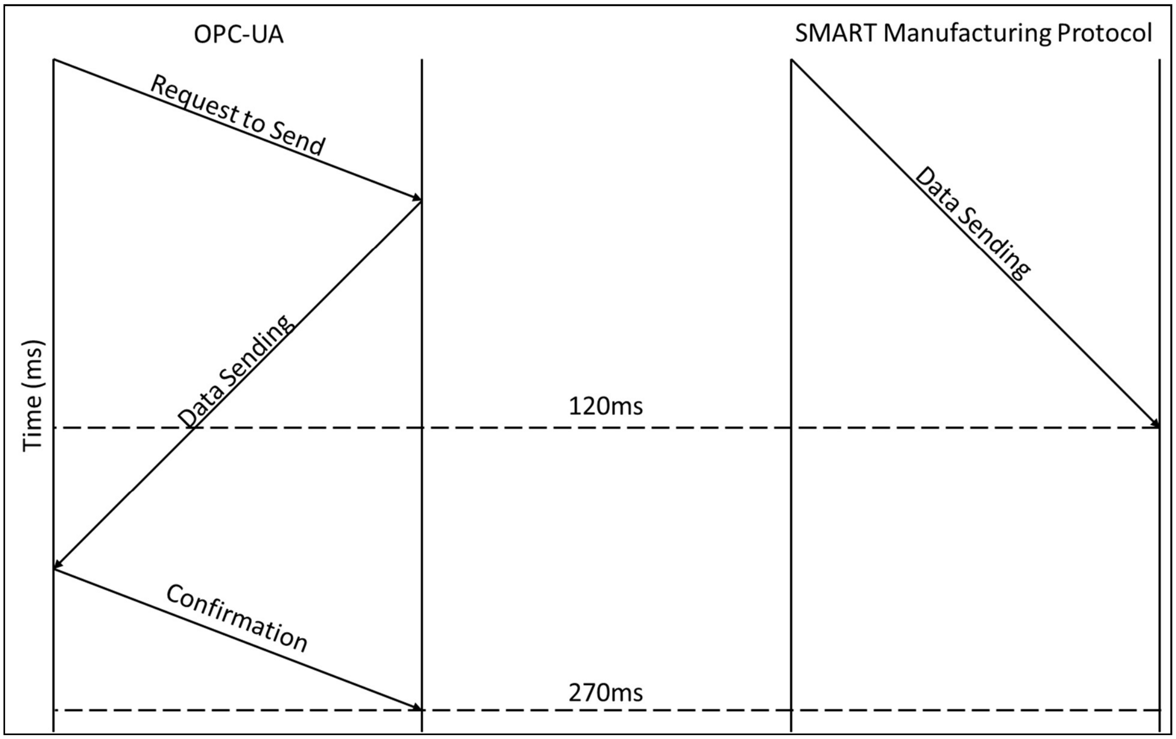

Figure 17: Communication Procedures [61]

3.6.2. Addressing

Addressing is concerned with the destination of sending and receiving data. Additionally, with the GET/PUT methods, memory addressed for sensor and calculated information by devices need to be in the machine itself for all devices on the network. This does align well with the needs of Industry 4.0, where machines would need more computational power to create intelligent decisions, as in Figure 18. These machines would not need access to the greater amount of CPU power all the time, and can therefore be used for network traffic information. Furthermore, a machine's ability to create intelligent decisions in real-time will be much easily realised when the machines have access to data stored locally in memory, and when they do not need to go and fetch the information.

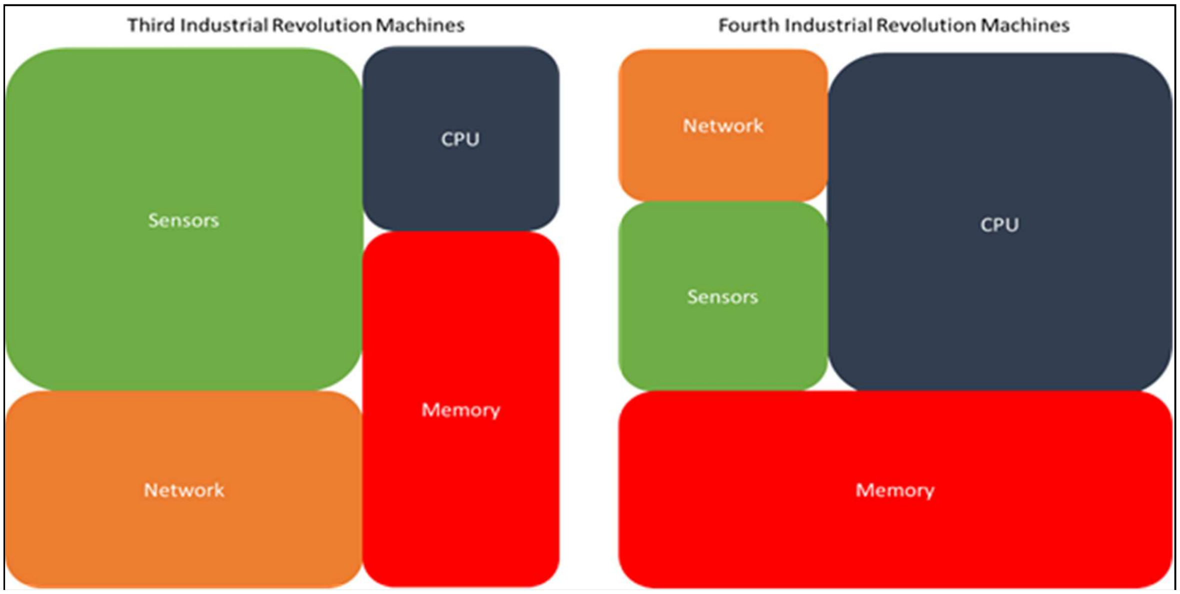

Figure 18: Machine Resource Distribution

3.6.2.1. Topology

Communication through a network exists between transmitters, facilitators and receivers of information. This is important in order to determine the scalability of a network and the transmission costs of messages. A network can be scalable through the addition of switches and routers. A router is able to assign a communication device an address.

Communicating devices then send the communicating methods to the router. The router decodes and forwards the request and packet of information to the relevant device. A switch alternatively can be seen as a multiple access point where addresses are set up on the devices themselves, and information can only be forwarded to the relevant address. A switch can therefore be seen as a multiple plugin device to expand a network with minimal to no latency cost. Switches and routers therefore do not send information, but only facilitate their flow.

3.6.2.2. Host Addressing

Host addressing is used to give a unique identification to all devices in a network. The IPv4 addressing scheme, which is used in part in the SMART Manufacturing Protocol (SMP), consists of four groups of three ten's numbers, e.g. 172.27.122.250. Each group of numbers is represented by an eight-set binary number, making the largest number assignable 255.

The largest assigning IPv4 address is then 255.255.255.255, and the lowest 0.0.0.0. This addressing scheme allows for over 4 trillion unique addresses. The groups are divided, as the first two groups represent the type of network, e.g. local network, or industrial network, called a net. The third group assigns a subnet mask, which further splits the network. The fourth group assigns the physical address within the subnet.

Therefore 255 devices can exist in one network and can communicate at any one time with minimal interference. For larger networks requiring more devices, subnet splitting can be used, and directed messages can be addressed with the applicable subnet mask. Addresses xxx.xxx.xxx.0 and xxx.xxx.xxx.255 are reserved, however. These reserved addresses are used for first-time entering of a network. When a new device is attached to a network, it is automatically assigned the xxx.xxx.xxx.255 address, called "this host address". The host address queries the "server host address" at xxx.xxx.xxx.0, where the host address is assigned a new address.

Upon each message sent from a machine, the message packet, seen in Table 1, is filled accordingly. For example, if SMU 1, with IP address 172.27.122.254, sends a GET method to SMU 2 at IP address 172.27.122.253, then the message packet is filled as such:

- The method is filled with the value 0001 byte representing a GET method.
- The destination is filled to represent 172.27.122.253.
- The version field is 0001 byte to represent the active version of the communication protocol.

- The header field is given a unique ID to represent the message sent from this machine for GET methods, with a value being incremented for data logging and message retrieval.

- The footer field is used in a similar way for PUT methods.

- The body contains machine-specific information.

Table 1: SMP Message

<table>
<tr><th>Byte</th><th>4</th><th>4</th><th>4</th><th>4</th><th>4</th></tr>
<tr><td>0</td><td>Method</td><td colspan="2">Destination</td><td colspan="2">Version</td></tr>
<tr><td>1</td><td colspan="2">Header Field Name</td><td>Value</td><td colspan="2"></td></tr>
<tr><td>2</td><td colspan="4" rowspan="2">Header</td><td rowspan="2"></td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td colspan="2">Footer Field Name</td><td>Value</td><td colspan="2"></td></tr>
<tr><td>5</td><td colspan="5">Blank Line</td></tr>
<tr><td>6</td><td colspan="4" rowspan="3">BODY</td><td rowspan="3"></td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
</table>

3.6.2.3. Memory Addressing

With SMART Manufacturing protocol, and with reference to Figure 18, each device has its own unique host address, as well as unique address for information. These addresses are used to index specific pieces of information of the SMART Manufacturing protocol, which are accessed by the GET/PUT methods.

© Central University of Technology, Free State

These memory addresses are accessed by communicating with the relevant host and then indexing the address. The GET method then returns a value that is updated on the initiating device. The PUT method rather updates a value of the receiving device. Since each method updates information on specific devices, multiple strings of these methods would need to be called to update information across all devices. At this point all devices have all other devices' information stored in memory, and do not have to collect information for decision making.

3.6.3. Message Field

As in Table 2, the SMART Manufacturing protocol has dedicated addresses reserved for device specifications reserved in the body of the SMP message. The body of the message contains information such as the machine type and machine status, while all other sensor data and calculated data are being communicated in a separate field called the "Data field", as seen in Table 2 below.

Table 2: Body Field Encapsulation

Byte	**8**	**8**
1	**MACHINE TYPE**	**MACHINE STATUS**
2	**DATA**	
3		

3.6.3.1. Machine Type

The machine type field is filled with an eight-byte number that is cross-referenced with a lookup table to determine what type of machine is connected on the network. This is used to identify, with the first-time setup, when a machine joins a production line with the reconfigurable property. If a new production is requested, the machines can determine themselves if the current factory layout is sufficient in order to complete the order, based on the machine type identifier.

© Central University of Technology, Free State

3.6.3.2. Machine Status

Communication to machines is done mostly through the PUT method. This is largely due to machines completing checks and production and updating all other machines about next incoming orders. It can be seen as a broadcast of information to all attached listeners, and success of communication is not priority monitored. Only when faults are experienced will a machine usually use a GET method to collect information and formulate and intelligent decision.

At that point a machine will have to be able to reach another machine successfully to gather information. For this reason, a machine is able to detect when a machine was last able to be communicated to, or if communication is failing to that machine. Such failure could be due to a break in communication channel between SMU 1 and SMU 2, as indicated in Figure 13, therefore allowing machines to be able to initiate a resend of request and time out. The status field also updates when the device is done receiving information and ready to be communicated to.

3.6.3.3. Data

This data field is entirely customisable, where users are able to send custom byte values. These can be sensor values, calculated values, trigger values and so on. Although the body field is entirely customisable, it is decided that the use of a machine type identifier and machine status field would be beneficial in order to create SMUs with SMART characteristics, enabling factors and abilities.

3.6.4. Methods

The GET and PUT methods, as described in section 3.6.1, will be implemented and tested in the method field of Table 1 in the third layer of Figure 15. Both methods' tests will be repeated, and an average taken to ensure concrete results and communication between SMUs. In both test cases, the full functionality of the factory will be conducted to test both overall communication time and latency. Thereafter, meaningful comparisons against existing communication protocols can be made in similar situations.

3.7. Conclusion of the Methodology and Creation of the Decentralised Communication Protocol

As expressed in the previous sections of this chapter, this study identifies that the best location for decentralized real-time communication to be conducted at the lowest level possible of a factory line, namely the machine level. The communication protocol created at lower level of the factory system, allows for more flexibility of information flow whilst adhering to more complicated restrictions such as finite machine type identifiers and physical network structure to name a couple. Inclusions of abilities of the communication protocol will allow for unique decentralised traits to be achieved whilst maintaining real-time communication can be evaluated.

© Central University of Technology, Free State

Chapter 4:

Decentralised Communication Protocol Performance, Results and Comparison

---

4.1. Introduction

This chapter showcases the results obtained from the tests conducted on the performance of the SMART Manufacturing Protocol. With the protocol being able to run in any factory scenario, the machine performance is not of critical concern, but is stressed to illustrate the intelligence machines could process with adequate communication. The main points that will determine the success of the communication protocol are as follows:

- the latency of communication;
- throughput of information; and
- expansion of network.

.

4.2. Decentralised Communication Protocol Performance

4.2.1. Latency

As discussed in section 3.6.1, the latency is measured in the amount of time that it takes a message to travel between one device to the next. There are several factors that can influence the latency of communication, such as decoding of the message structure, link speed, chance to communicate in a network, and total message length to be sent.

The latency test is broken into two categories: One Step Latency and Two Step Latency. One Step Latency refers to one machine communicating to another. It is the simplest form of communication and is used as a baseline for the communication protocol. Two Step Latency refers to one machine communicating to another through an intermediate machine.

This is possible due to the fact that each machine possesses a copy of information sent from one machine to another. An intermediate server can pass along the message information when programmed, which will ensure information interchange, in the event of a link breakage. In both cases, an individual test of the GET and PUT method was performed in quadruple to determine the average latency.

4.2.1.1. *One Step Latency*

During the One Step Latency test, a machine sent a standard payload size of 64kbs on the sequence seen in Table 3. The test results of the One Step Latency are covered in detail in the proceeding sections of the GET and PUT methods.

Table 3: One Step Latency Results

One Step Latency Results			
Test Number	Method	Sequence	Overall Communication Time
1	GET	1⟶2	115ms
2	GET	1⟶2	114ms
3	GET	2⟶1	115ms
4	GET	2⟶1	114ms
5	PUT	1⟶2	58ms
6	PUT	1⟶2	57ms
7	PUT	2⟶1	59ms
8	PUT	2⟶1	58ms

4.2.1.1.1. GET Method

As seen in Figure 19: One Step Latency GET Method 19, SMU 2 queries the information from SMU 1. What follows is a response from SMU 1 with a reply. This method can be seen as a double PUT method, as SMU 2 queries SMU 2 with the needed information in PUT form, and SMU 1 replies with the information with a PUT method on SMU 2. Due to this fact the GET method has an increased latency due to the double queries with the added latency of computational time. The computational time of the SMU is the programme cycles required to fetch and move the information out of stored memory.

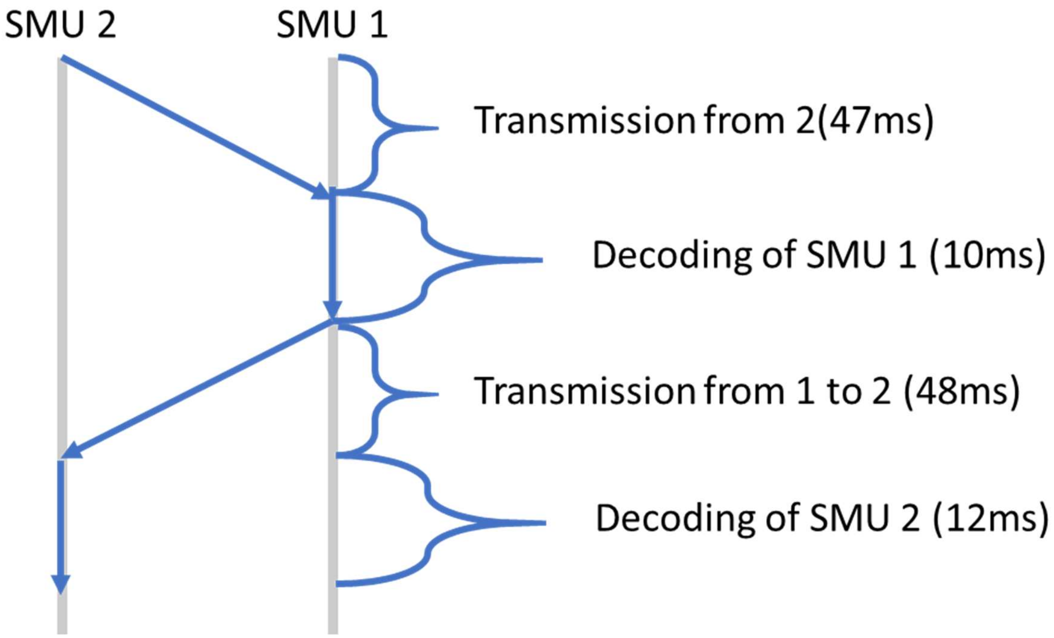

Figure 19: One Step Latency GET Method

4.2.1.1.2. PUT Method

As seen in Figure 20, SMU 2 queries SMU 1 using the PUT method. This method is used to update information for specific events when crucial information is calculated at one machine and should be updated throughout the network. When compared to the GET method in the One Step Latency, a greater reduction in latency is perceived, since SMU 1 is creating new information and responsible for passing such information to the other machines. As a result of the reduced latency structure it is advised that a PUT method be used as the primary means of communication. The use of this method would also ensure congruency of information between machines when new events take place.

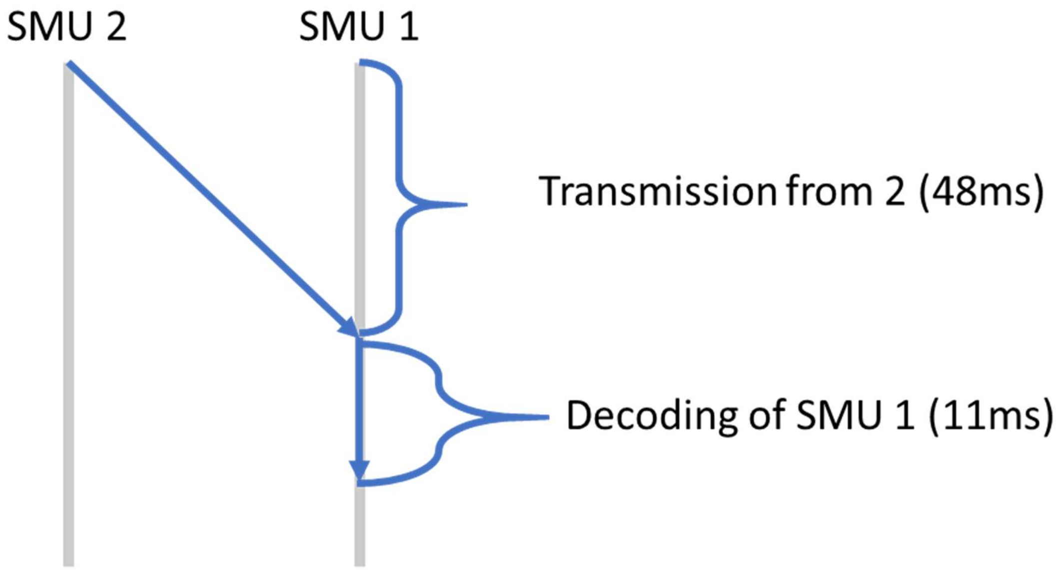

Figure 20 One Step Latency PUT Method

4.2.1.2. Two Step Latency

During the Two Step Latency test, a machine sends a standard payload size of 64kbs on the sequence seen in Table 4. The test results of the Two Step Latency are covered in detail in the proceeding sections, namely 0 and 4.2.1.2.24.2.1.1.1 of the GET and PUT method. These methods where measured using network analytical tools that defined lower power confirmation flag checks. These flag checks would be sent after computational program cycles where completed and when deemed safe to do so, thus a slight difference in "Decoding Time" could exist, but maintains to be negligible and not considered for the purpose if defining the communication protocol's latency.

Table 4: Two Step Latency Results

Two Step Latency Results			
Test Number	Method	Sequence	Overall Communication Time
1	GET	1→3→2 2→3→1	227ms
2	GET	1→3→2 2→3→1	222ms
3	GET	2→1→3 3→1→2	225ms
4	GET	2→1→3 3→1→2	222ms
5	PUT	1→2→3 3→2→1	119ms
6	PUT	1→2→3 3→2→1	115ms
7	PUT	2→1→3 3→1→2	117ms
8	PUT	2→1→3 3→1→2	115ms

4.2.1.2.1. GET Method

Depicted in Figure 21 is the illustration of the Two Step Latency GET method. The Two Step Latency method is applied when one machine is unable to communicate with another. In this case, and the next, it was purposely done by separating the link between two machines. In this specific case SMU 1 can communicate with SMU 3, and SMU 3 can communicate with SMU 2.

Therefore, the only way SMU 1 can communicate with SMU 2 is through SMU 3, with SMU 3 acting as an intermediate and independent communication module. It allows SMU 1 to query SMU 3 to gather information from SMU 2 and return it to SMU 1. This can be seen as a successful information return, albeit with an increased latency as the inclusion of an extra step to complete the communication cycle.

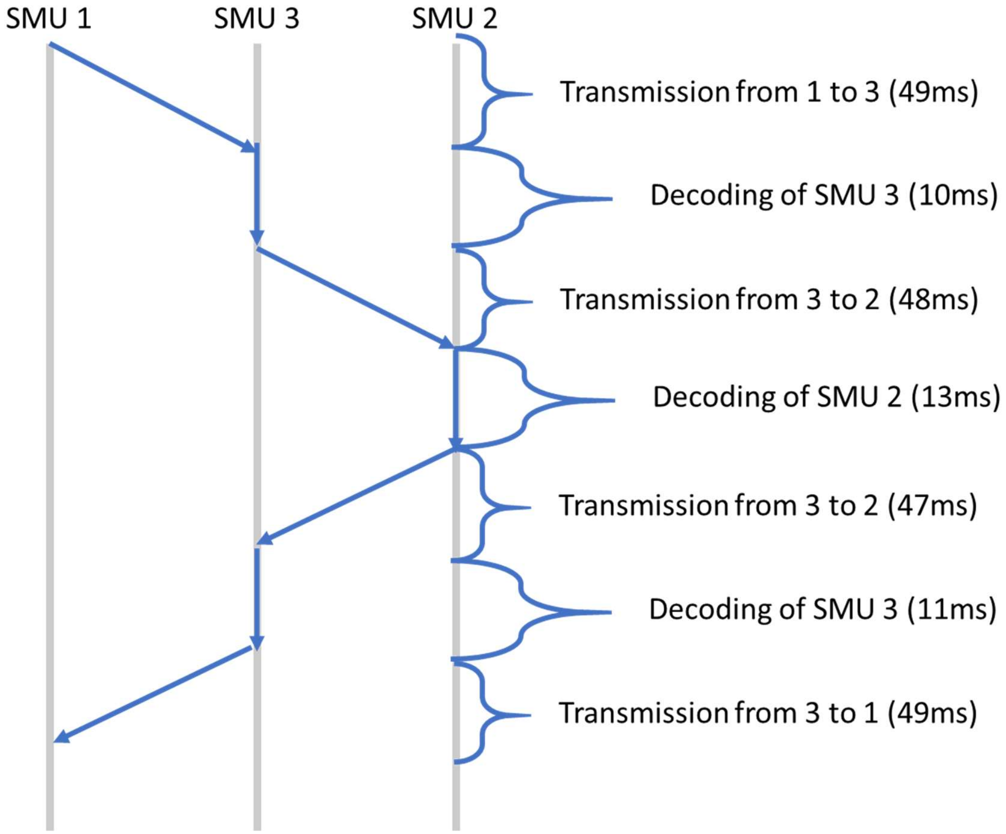

Figure 21: Two Step Latency GET Method

4.2.1.2.2. PUT Method

As seen in Figure 22, SMU 1 needs to update its information to SMU 2. Again, the link is broken, and SMU 3 can be used as an intermediate communicator. However, in this test the latency is reduced because the flow of information does not need to be returned. Instead, the information is confirmed with the use of the checksum at the received device.

It is also noted that SMU 1 already attempted to communicate to SMU 2 directly, but failed due to the lack of a connection link. In some cases, this could be perceived to increase the overall latency. However, as previously stated, latency is the measure of time taken from when a message is sent from one device to another across the medium on a successful transfer.

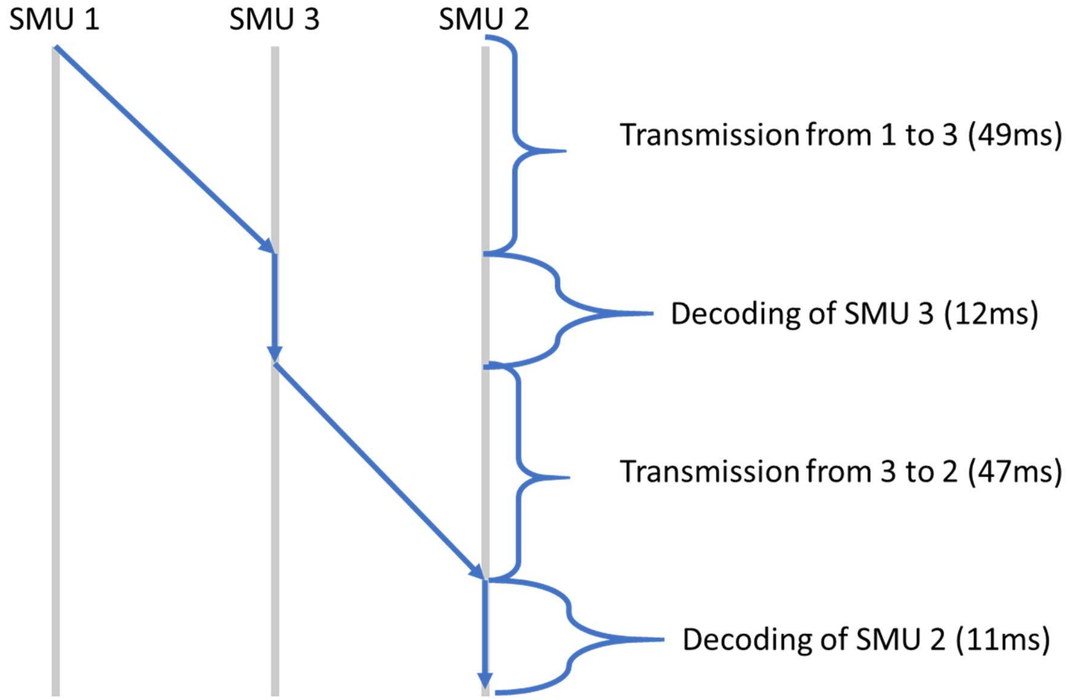

Figure 22: Two Step Latency PUT Method

4.2.1.3. Summary

In Figures 23 to 26, the conducted tests are categorised, and average latency is calculated. The perceived latency of the GET methods are greater than doubled, due to the return of information. Even with this occurrence, across all instances the communication protocol was still within real-time communication limits, namely at 250ms; Two Step Latency GET method averaging 224ms; Two Step Latency PUT method averaging 117ms; One Step Latency GET method averaging 115ms; and One Step Latency PUT method averaging 58ms.

Furthermore, all of these communication methods outperformed that of the OPC and OPC-UA protocols, which averaged 270ms. The reason for the reduced latency is the use of simple information that does not need to be repackaged for multi-vendor support, lightweight payload and reduced handshaking.

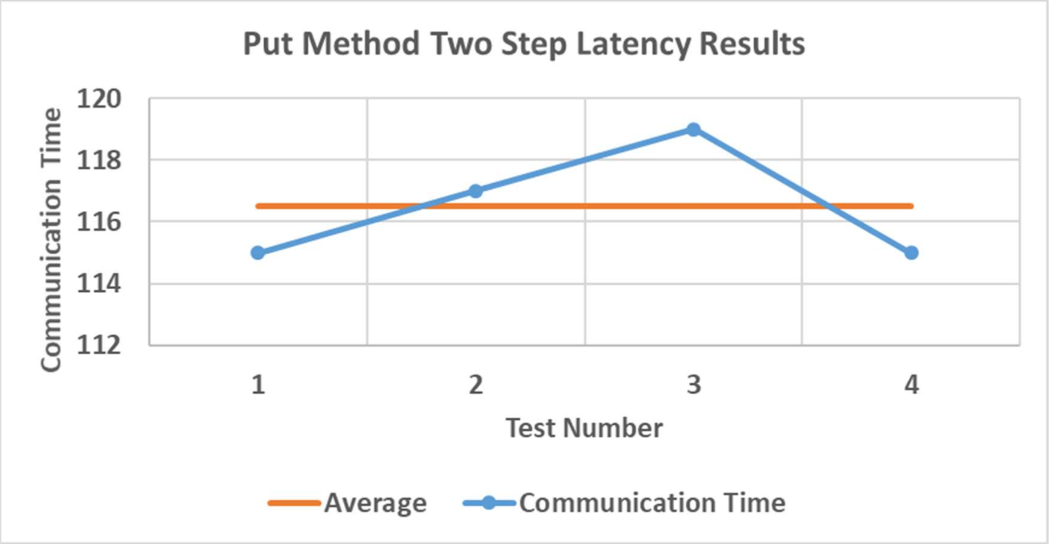

Figure 23: Two Step Latency Put Method

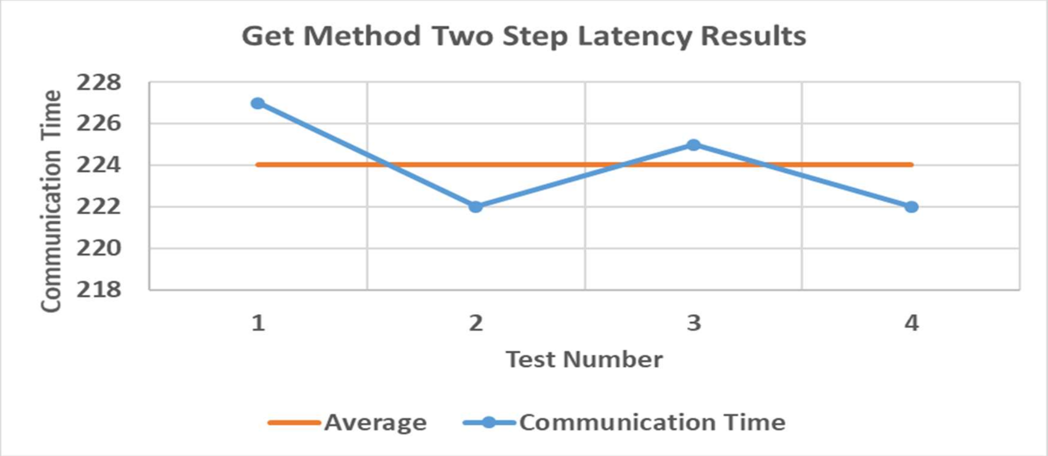

Figure 24: Two Step Latency Get Method

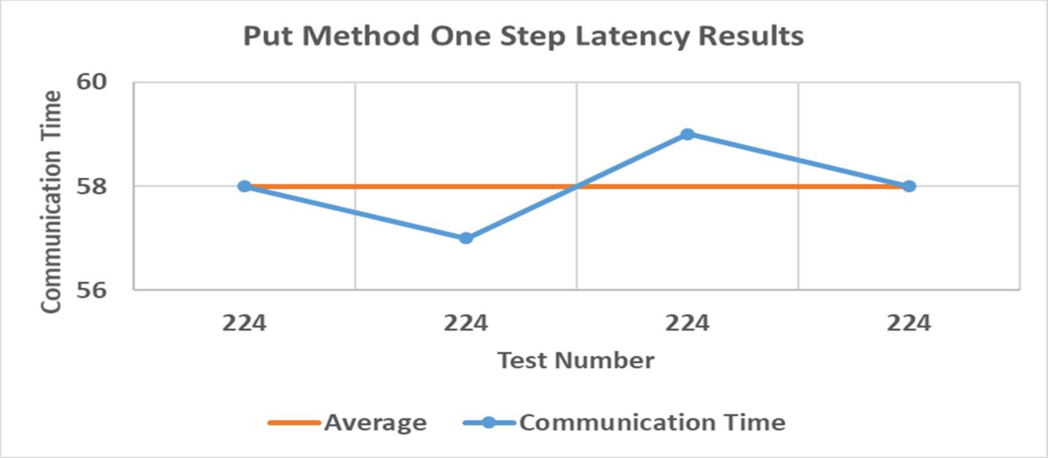

Figure 25: One Step Latency Put Method

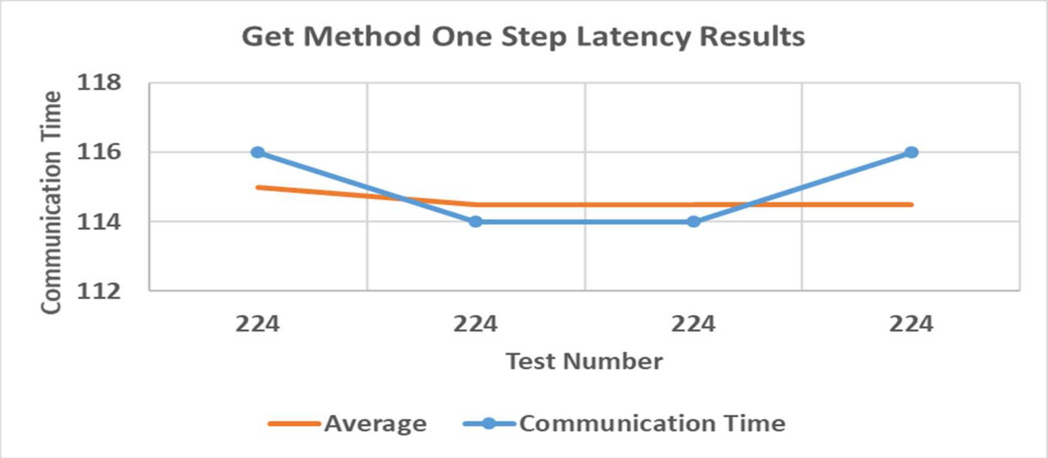

Figure 26: One Step Latency Get Method

© Central University of Technology, Free State

4.2.2. Throughput of Information

As mentioned in section 4.2.1, the message in both the GET and PUT methods is sent using a dedicated space separate package. This packet contains vital and non-vital information to be filled in before sending. Vital information that needs to be sent in a message is covered in the method, destination and version field. Only the body field contains non-vital information that can be left blank to send a message.

As a result, each packet that is sent is able to have a minimum packet size of 120 bytes with 0 memory, address lines, and a maximum packet size of 64 kilobytes with 2600 addressable memory locations per device. It is regarded as sufficient space in both the packet and the machines to store data on each machine across the network, without overloading any aspects. The majority of the packet payload is stored in the body field, which is user customisable data used during transition of information. This body field can be seen in Table 5, which is again displayed below for convenience.

Table 5: Communication Protocol Fields

Byte	**4**	**4**	**4**	**4**	**4**
0	**Method**	**Destination**		**Version**	
1	**Header Field Name**		**Value**		
2	**Header**				
3					
4	**Footer Field Name**		**Value**		
5	**Blank Line**				
6	**BODY**				
7					
8					

© Central University of Technology, Free State

4.2.3. Expansion of Network

With the communication protocol being designed and situated in one of the lowest tiers of any network layer, it should be encompassing of all other layers and flexible yet efficient in its duties. For this reason, it is important to mention the use of this protocol when a network is expanded to an N number of devices.

4.2.4. Connection Method

As seen in Table 6, the number of connections needed when compared to n number of devices follows a triangular number pattern with a slight modification in the calculation of the second difference formula. The traditional triangular number sequence being:

$$Xn = \frac{n(n+1)}{2} \tag{1}$$

Where:
Xn = the number of devices
n = the number of connections

The formula used to calculate the number of connections in this case is:

$$Xn = \frac{(n-1)[(n-1)+1]}{2} \tag{2}$$

Where:
Xn = the number of devices
n = the number of connections

© Central University of Technology, Free State

Using this formula, it is possible to determine the number of connections needed when expanding the network in order to keep reduced latency across the network and keep a decentralised approach.

Table 6: Connection Table

Connection per Devices	
Number of Devices (Xn)	Number of Connections (n)
1	0
2	1
3	3
4	6
5	10
6	15

4.2.5. Redundant Connections

The length of the connection barely varies with the overall latency of a communication. Due to the link speed, the number of hops across an intermediate device increase the perceived latency of a protocol. Although the redundancy of the network expansion does become exponentially drastic on resources and financials, it is still recommended to follow this procedure whenever possible to preserve the decentralised aspect of each device. A fully decentralised network consisting of six nodes is recreated in Figure 27.

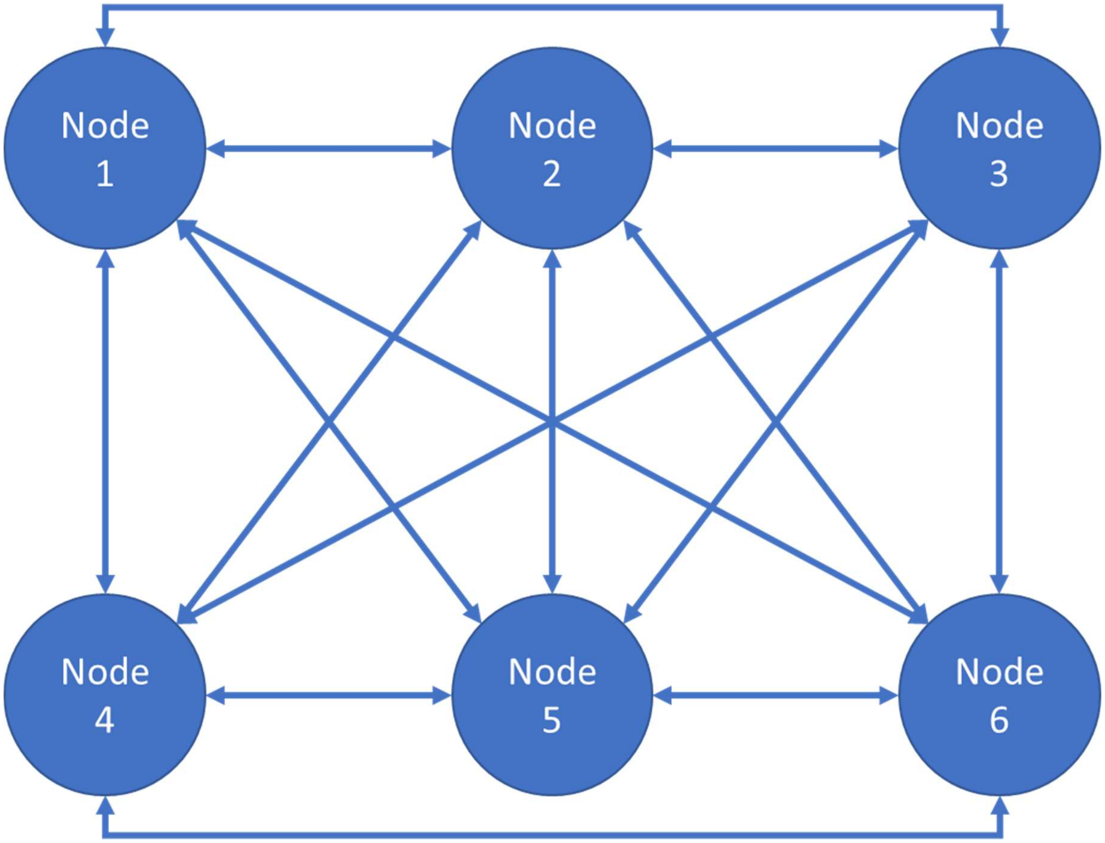

Figure 27: Full Decentralised Network

These decentralised network structures are often referred to as mesh structures, where each node is connected to every other node in the network. However, as machines and factories become more spread out and separated into spaced assembly lines, a refined and hierarchical butterfly/Grid connection, seen in Figure 28, could suffice while reducing the overall latency into real-time communication binds.

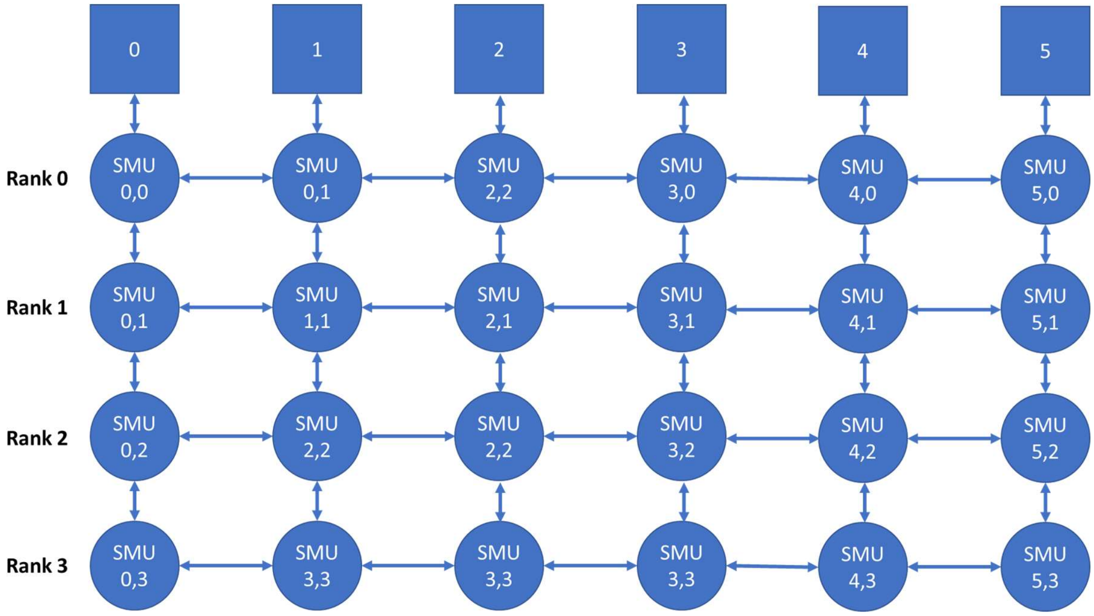

Figure 28: Proposed Grid Network

4.3. Decentralisation Protocol Summary

4.3.1. Latency

As expressed in section 4.2.1, the overall transit time of the communication protocol when expressed under maximum load conditions is less than that of the real-time communication limits and the OPC/OPC-UA communication time. Furthermore, the overall latency of the communication protocol is also lessened when compared to that of OPC/OPC-UA and other protocols. In effect it is with this measurement that the communication protocol has succeeded through this dissertation in one-third parts.

In the application case of the water bottling plant, it is important to note that the communication protocol was not used to alert the next machine in the manufacturing line of the time of arriving product. Although this could now be possible across all manufacturing plants in real time, the requirements of Industry 4.0 and SMART Manufacturing in chapter 2 indicate the need of attached sensors. The simplicity, verification and ease of including sensors on machines to be alerted of the product arrival on the machine will almost never be rivalled by communication between machines. Therefore, it is important to designate the type of information that should be sent across layers. In this case each machine rather specifies the next product's information that will be arriving. Specifically to the water bottling plant, this information would be the type of bottle (500ml or 300ml) and the calculated capping height. Passing the information along will allow SMU 2 to cap the bottle without calculating the height of the bottle again, therefore not reducing sensors, but fixtures to machines and computational time.

4.3.2. Centralised Manufacturing Units Vs. Decentralised Manufacturing Units

As the communication protocol is enabled with redundant connections, each machine in the network may operate independent of each other and without the need of a central server. It adds to the second of the third measurement of success of the communication protocol within this dissertation, namely the decentralised ability. Without the need of central servers, it is also seen as financially advantageous, as communication takes place in a decentralised manner.

4.3.3. Intelligence

In the application case of the communication protocol working in the water bottling plant, certain intelligence can be programmed into the machines that are facilitated with the correct and crucial information being sent. It is observed during the manufacturing of the water bottling plant that each machine can verify and check the current bottle order production. This allowed each machine to gauge its current state in the order that it is executing.

© Central University of Technology, Free State

Specific to the water bottling plant, it allowed a stack programme to be created in each machine. When SMU 2 calculates the average time between the arrival of the bottles to be capped and verification information from the previous machine, it can place missing bottles on a stack and classify them as "missing".

It was accomplished by physically removing a bottle off the production line between SMU 1 and SMU 2. Therefore, SMU 1 had already filled five bottles, while SMU 2 had only capped four. At the end of the 10th bottle order, SMU 1 had completed its order, whilst SMU 2 had not. This allowed SMU 2 to request another bottle to be filled with the stack information. As a result, although SMU 1 had filled 11 bottles in total, SMU 2 and SMU 3 capped and packaged the complete order of 10 bottles.

4.3.4. Decentralisation

In 4.2.1.2.24.2.1.2.2, the results obtained in the Two Step Latency PUT displayed an interesting characteristic available through decentralisation. During the tests SMU 1 was able to receive and deliver information to an ultimate host address of SMU 2. This information was passed along from SMU 1 to SMU 3 and ultimately SMU 2.

Undesirably, a link test was programmed into SMU 1 to determine a possible connection path to SMU 2. However, it is still possible to automate this process with further work into the communication protocol. It did in fact prove the applications of a decentralised network in the Fourth Industrial Revolution. If machines can communicate in a decentralised manner and in real time, constrictions in traditional manufacturing are broken down at a machine level.

4.3.5. Information Organisation

In the network describe, it is observed through experimentation of this chapter that information organisation became more defined in Industry 4.0. As with sectors in Industry 4.0 coming to fruition, specifically Digital Twins, calculations and memory storage could become redundant, and similar problems in the Third Industrial Revolution could be experienced such as latency, redundancy and dependency.

In the instance of the water bottling plant, a digital twin was designed and installed. This digital twin allowed for the measuring of machine supplies, such as water and caps, as well as the overall production time of a specific order. The digital twin was also able to collect orders from a website and assign tasks to the relevant machines through the central information.

This allowed for the digital twin to become an overseer of production, communicating with the machines at the same hierarchy level to set production speeds in case of insufficient stock. The digital twin was also tasked with calculating the overall production time, as if each machine would have to calculate this based on the machine data in memory of the communication protocol, meaning that the calculation would be replicated three times across three different machines, introducing redundancy.

In the above case, the machines should rather be tasked with the communication of the product being currently produced. As described in section 4.3.3, SMU 1 communicates the height of the bottle to SMU 2 where it is placed on a stack. If a second product information packet arrives while the previous has not been processed, then SMU 2 places the newer packet above the previous.

This stack can then be communicated back to SMU 1 to complete the order. The product information is desirable to be communicated from machine to machine at lower latency and only when crucially needed. Hence, the need for overall production time should be calculated elsewhere.

Finally, in the above-mentioned case, each machine should calculate its own product arrival time. It would imply that SMU 2 should be able to detect, with the use of sensors, when a bottle has arrived for capping and verify with the stack the product information, or recalculate the information itself.

This negates the need for SMU 1 to calculate and inform SMU 2 when exactly the product will arrive at the station. The information of the product arrival time can be estimated through each machine, with previous product arrivals. It will again reduce the data redundancy that each machine should send over the communication protocol, to allow for real-time communication of crucial information.

Figure 29 illustrates the type of information that should be calculated at each system part, as well as who is capable of directly and easily viewing the information. When the communication protocol is used in conjunction with this information organisation, it frees up intended resources for crucial information to be sent by the machines and reduces redundant calculations from machines and servers, such as order completion time.

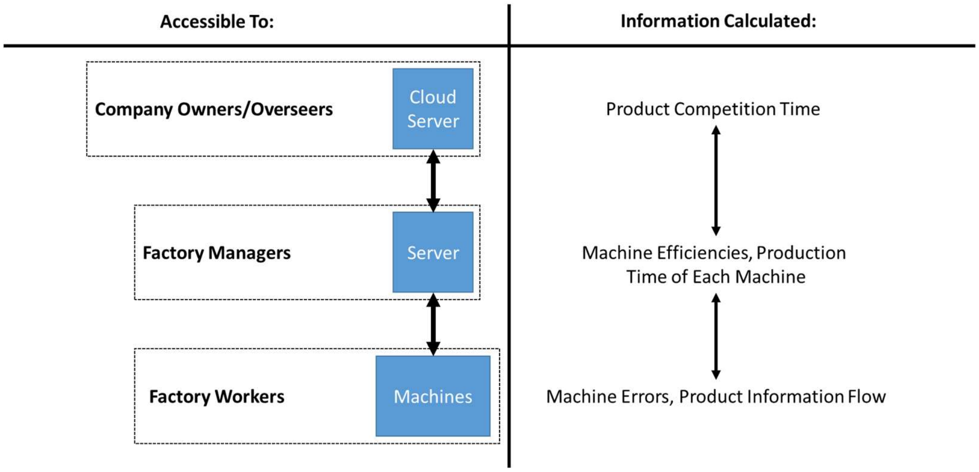

Figure 29: Information Organisation

4.3.6. Congruency of Information

In the case of the Two Step Latency GET results in section 4.2.1.2.1, a higher latency is experienced than in the One Step Latency GET method. Yet, the Two Step Latency GET method should ultimately not occur if all machines along the network require all the information to be congruent. If SMU 1 needed to collect information from SMU 2, but SMU 3 also required the information from SMU 2, SMU 1 could then have requested the information from SMU 3 via the One Step Latency Get Method. This is only expected to be possible if SMU 2, and all machines on the network for that fact, updated their information across the network via the PUT method once an update was calculated. It essentially eliminates the requirement of the Two Step Latency GET method, if information is congruent across the network.

4.3.7. Expansion of Network

As detailed in section 4.2.30, the expansion of the network is determined through the number of devices in the network being calculated from the number of connections needed. It allows one to determine the physical hardware costs of the network.

4.3.8. Redundant Connections

Similarly, in section 4.2.5, Figure 27 detailed the total expansion in the network in order to become fully decentralised. This dissertation has outlined that the physical cost becoming exponential with the increase in number of devices becomes unsustainable. It is for this reason that future proposed networks be evaluated, in order to determine a minimal latency cost with key decentralised communicators existing in every sub network.

An example can be seen in Figure 30 in order to preserve a 50% average increase in latency and 30% decentralised connections.

© Central University of Technology, Free State

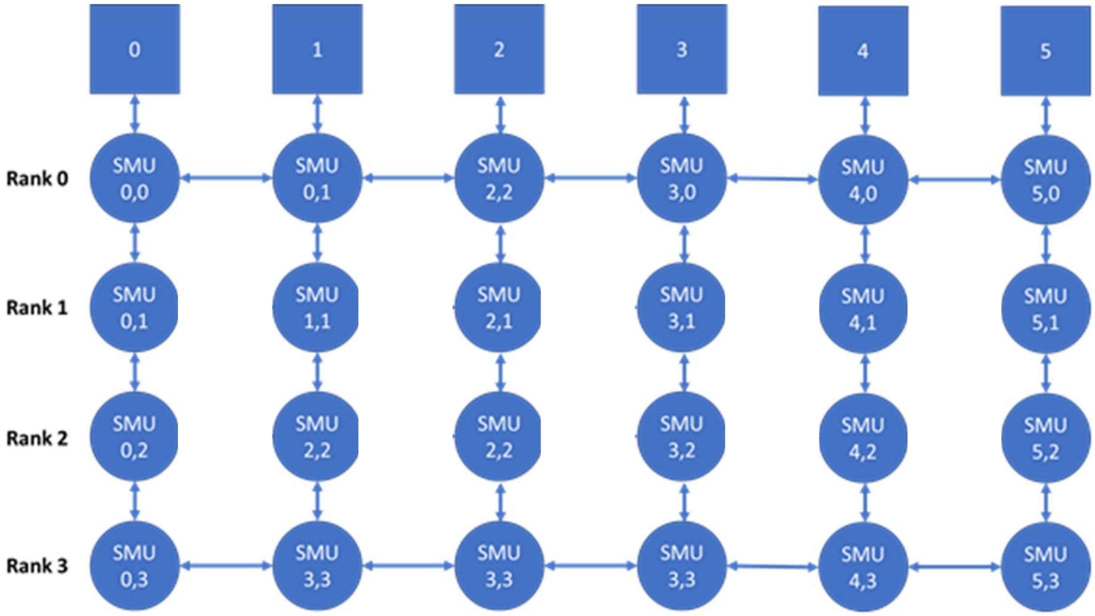

Figure 30: Proposed Minimal Decentralised Network

4.4. Communication Protocol Field Highlights

This part discusses a few notable highlights of the communication protocol to give a clearer insight in the construction of the communication protocol.

4.4.1. Machine Status

A key attribute in SMART Manufacturing is that of machine intelligence. As expressed in section 2.2.3 of this dissertation, a machine can gather intelligence through relevant information transfer between machines. It not only optimises decision making on machines, but allows machines to become contextually aware of their surroundings.

One important aspect of information transfer accommodated with the SMP is to allow machines to be assigned a unique machine type identifier and status. This allows surrounding machines to read the information and calculate if current manufacturing can be completed with the attached machines, and if the machines are online. This is seen in the Header Field Name and Value.

4.4.2. Destination

Table 1 outlined specific information being transferred by the SMP. It is clear to see a Destination Field but no Source Field. The source information, predominantly the IP address of the devices, is not needed, since it is not relevant for a machine to know where the transmitting machine lies within the network, but rather what type of machine sent the information.

Machine identification comes from the type of machine attached in the network. As SMUs are able to switch and change out of manufacturing lines, blind approaches of identification allow a receiving machine to formulate decisions based on surrounding and attached machines' information, as well as the machines' types and statuses. This already being covered from the header values, the machines are able to determine uniquely which machine was in communication by identifying changed memory block locations.

As the memory block locations are reserved per device on the network, it allows machines to identify which information has been changed, by which machines and specifically the machine involved, clearing up the communication received previously shrouded by only IP address sources. This unique way of source identification therefore does not require a specific source field to be sent during each message. The exclusion of the source field can thus be seen in Table 7, which is given below for convenience purposes.

© Central University of Technology, Free State

Table 7: Decentralised Communication Protocol Fields

<table>
<tr><th>Byte</th><th>4</th><th>4</th><th>4</th><th>4</th><th>4</th></tr>
<tr><td>0</td><td>Method</td><td colspan="2">Destination</td><td colspan="2">Version</td></tr>
<tr><td>1</td><td colspan="2">Header Field Name</td><td>Value</td><td colspan="2"></td></tr>
<tr><td>2</td><td colspan="4" rowspan="2">Header</td><td rowspan="2"></td></tr>
<tr><td>3</td></tr>
<tr><td>4</td><td colspan="2">Footer Field Name</td><td>Value</td><td colspan="2"></td></tr>
<tr><td>5</td><td colspan="5">Blank Line</td></tr>
<tr><td>6</td><td colspan="4" rowspan="3">BODY</td><td rowspan="3"></td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
</table>

4.5. Overall Communication Protocol Comparison

As seen in Figure 31, the decentralised communication protocol, or SMART Manufacturing Protocol (SMP) as previously mentioned, compared significantly lower than most other communication protocols, and even against its main competitor, OPC and OPC-UA. Albeit the communication protocol did not outperform every communication protocol, with PTP posting only slightly faster communication times, the performance of the protocol should also be expressed in functionality, which is to follow. Importantly it has to be stated clearly that the GET method of the SMP is represented in Figure 31, as it is the comparable form of communication method against other protocols. Protocols such as OPC, OPC-UA and PRP only pose status update “POST” methods, while WebSocket, HTTP and MQTT all allow for this duplex form of communication. These protocols are then compared with GET method counter parts for homogenous comparisons.

© Central University of Technology, Free State

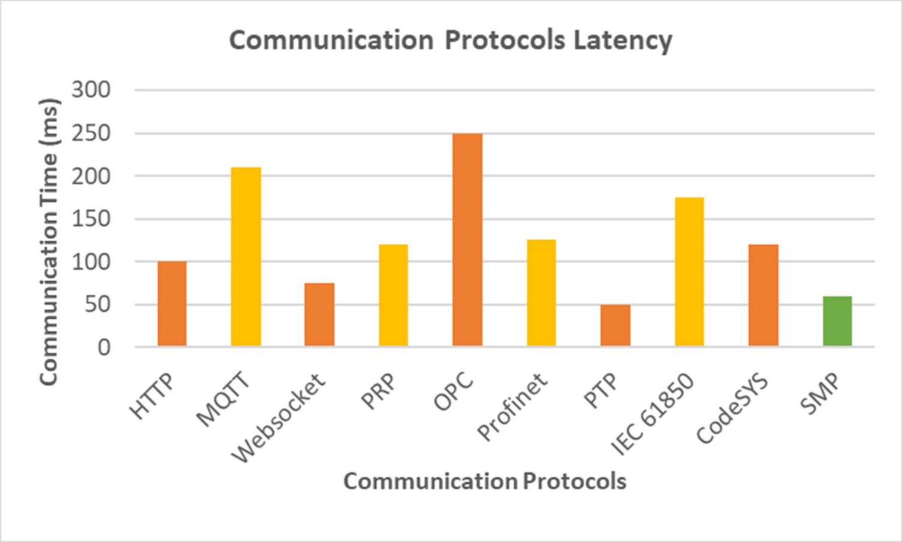

Figure 31: Communication Protocol Comparison

Table 8 summarises the communication protocols compared in Figure 31 alongside the functionality of each protocol. As the SMP had a heavy focus in use of SMART Manufacturing, it ultimately drew upon requirements and best qualities of other protocols in the surrounding field. With this it is evident that the SMP performs well in sectors such as historical message logging, decentralisation and latency. Comparatively with the other communication protocol, each protocol accounts for drawbacks in functionality, albeit each functionality is not implemented in each factory case.

Table 8: Communication Protocol Comparison and Properties

		Properties						
		Latency (ms)	**Payload Per Query (Bytes)**	**Historical Message Logging**	**Error Checking**	**Decentralised**	**Real-Time**	**Multi-Vendor Support**
Communication Protocol	**OPC**	**250**	**32**	✓	X	X	X	✓
	WebSocket	**75**	**64**	X	✓	X	✓	X
	SMART Manufacturing Protocol	**60**	**64**	✓	✓	✓	✓	X
	HTTP	**100**	**64**	✓	✓	✓	✓	X
	MQTT	**210**	**64**	X	X	✓	✓	✓
	CodeSYS	**120**	**32**	X	✓	✓	X	X
	Profinet	**125**	**64**	✓	✓	X	✓	X

© Central University of Technology, Free State

4.6. Conclusion

In this chapter the performance of the communication protocol was reviewed in its raw form. It allowed the results to reflect the true load of latency, under full payload, maximum throughput of information and the ability to communicate in a network. Following is the substantiation of abilities of the protocol due to the desired requirement in its application.

Following the performance of the communication protocol, it is observed that the PUT method outperformed its counterpart, GET method, by reducing the overall latency to near half. For this reason, it is evident that the PUT method should be used as the primary communication method between individual machines in a network. However, in the case of the Two Step Latency results, it is practical to allow machines to communicate through a network, without a direct link between each machine. This reduces the redundancy and need for greater resources; however, it creates the challenge of centralisation and dependency of other machines in the network to communicate.

With this newly created communication protocol, machines are able to possess contextual awareness by identifying attached machines in the network through unique machine identifiers. Machines are also able to communicate indirectly with one another through memory access and pass-along effects, while having active memory on hand for faster decision.

Most notably, however, is the ability of the SMUs to operate in a decentralised manner alongside enabling factors of the interoperability characteristic. Several of the key functionality of communication protocols, such as latency, payload, real-time operation and multi-vendor support also define the veracity of the communication protocol. The limitation of the decentralised communication protocol is that, currently, it does not offer multi-vendor support. However, there are only two other communication protocols that currently allow for multi-vendor support, OPC and MQTT. OPC does not support decentralised operation, while MQTT does not allow for error checking or the storage of historical messages, and has latency that is three times greater than the decentralised protocol with the same payload.

Considering all of the above aspects, it is regarded as the decentralised communication protocol overall improving machine intelligence to enable characteristics of the Fourth Industrial Revolution for machines, and improving performance in conjunction.

© Central University of Technology, Free State

Chapter 5:

Discussion and Conclusion of Communication Protocol

5.1. Introduction

This chapter aims to highlight the contributions in this study to the existing body of knowledge. It is done by summarising the achieved results, revisiting the research goals and objectives of this study, drawing a conclusion to the research, identifying the research contributions of the study, and finally providing a future scope of study.

5.2. Dissertation Summary

The first chapter of this dissertation introduced the research project as well as briefly outlining the gaps in current research and technology in SMART Manufacturing. The direction of approach, research hypothesis and research goals are then defined, on which the literature review and research methodology were based.

The second chapter provided a comprehensive literature review with a particular emphasis on Industry 4.0, SMART Manufacturing and the communication in them. This chapter also included the first elucidation of the communication latency and transit time of existing communication protocols.

The third chapter focused on the research methodology and creation of the decentralised communication protocol. Alongside this is a brief look on the creation of the water bottling plant used as a case study, and the network distribution of this case study during the centralised and decentralised operation. The communication protocol was developed as a new programme block that allowed for communication in Siemens S7-1200 modules, through Simatic Step 7 Version 13.

The fourth chapter covered in detail the decentralised communication protocol transit time and latency, as well as unique characteristics that allowed for a variety of SMUs' characteristics in the interoperability designation. A comparison was then made with existing protocols in similar lights, summarising their benefits together with their performance.

5.3. Research Goals

The major research objective of this study was to permit real-time decentralised communication between SMUs in an Industry 4.0 setting. It was done by creating a decentralised communication protocol which could achieve real-time communication (below 120ms).

With this study objective clearly demarcated, a well-defined literature review delving into Industry 4.0 and SMART Manufacturing ensued. While also giving sufficient coverage to centralised manufacturing scenes, the literature review convened with covering communication in these sectors.

What followed was development of a physical manufacturing line that acted as a test bench. This manufacturing line was chosen to be that of a water bottling plant. The water bottling plant served a threefold objective. Firstly, it allowed the use of the manufacturing line to be tested as a centralised manufacturing line with SMART Manufacturing Units communicating through OPC-UA. This allowed for a current benchmark to be obtained, with which to compare the communication protocol to. Secondly, the installation of the communication protocol between the SMUs and obtaining results during production to ensure real-time decentralised communication could be implemented. Finally, a network strategy was developed and implemented that involved a combination of communication protocols, namely OPC-UA and the newly created decentralised communication protocol that allowed for network information splitting for greater production efficiency, as well as the inclusion of a unique Industry 4.0 characteristic, Digital Twins.

© Central University of Technology, Free State

This finally led to the completion of the study's final research goal of analysing and comparing the latency and transit time of the communication protocols, as well as analysing the decentralisation of the SMUs.

5.4. Research Contributions

In this dissertation lies the development of a newly created decentralised communication protocol, falling upon newer trends and technology in order to achieve real-time communication. The following contributions from the study are considered to be novel.

5.4.1. Contributions to Existing Knowledge

Contributing to this dissertation is a complementing review paper on existing communication protocols that can be used in Industry 4.0 that looks into specific aspects of the communication protocols in terms of latency and communication time, as well as the general ability of each. The paper titled "Communication Protocol Review for SMART Manufacturing Units in a Cloud Manufacturing Environment" laid the foundation for this dissertation, and highlighted the lack of relevant research in the field.

To date of this dissertation there are a total of five publications, broken down to two conference papers and three journal articles, published between 2019 and 2020.

5.4.2. Development of a decentralised Communication Protocol

With the approach of Industry 4.0 onto existing systems, the requirement from factories and assembly lines brings about new challenges and opportunities. These new challenges should not be met with old approaches, and nor should the new opportunities be unmet with outdated technology. Therefore, the development of the new decentralised communication protocol, aligned with the future needs and trends of factories and machines alike, offer an encompassing approach to fully maximise the opportunities of Industry 4.0.

This decentralised communication protocol enables machines to fathom intelligent decisions without sacrificing on communication speeds or centralisations, further adding to the intelligence of machines.

5.4.3. Contributions to Industry 4.0

Encompassed in this study was the case study of the water bottling plant. The case study not only allowed for the study to compare and test communication protocols, but was also the backdrop for three more studies. These studies included:

The creation of a digital twin of each machine and housed in the central server. This study that produced a journal article titled “Design of Digital Twins for Optimization of a Water Bottling Plant” not only showed the construction and modelling of a digital twin in MATLAB, but also showed the beginning of the network splitting strategy, as the digital twin is attached to the central server and should therefore simulate the entire production whilst still becoming detached from the system.

The water bottling plant also provided a habitation for the study of SMART Product tracking that allowed machines to communicate with each other through products. This has the added advantage of detached communication channels and better tracking of the product throughout its lifetime, with uniquely identifiable and writable RFID tags, proving that multiple methods of communication between machines exist for Industry 4.0, leading to a final year Engineering student's project and a conference paper being produced. However, with the attached hardware and communication barriers, it did not prove to solve the problems specific to this dissertation, but it did provide valuable information on new possibilities and inspirations.

5.5. Future Work

As of the date of this dissertation, the communication protocol is only able to work through and be optimised for Siemens S7 units. It requires a homogenous mixture of machines through the network. Ideally this would be expanded into all or at least most manufacturers' equipment, as some machines are better at certain tasks than other. However, the lack of support from vendors to provide cross-platform communication halters this expansion.

It is well within the right of manufactures to only allow optimal communication on their platforms. However, with the expansion of Industry 4.0, it will have to be revisited in order to fully incorporate the attributes and applications of Industry 4.0.

5.6. Scientific Outcomes

In the timeline of this study, several research articles were written and published, covering unique and specific aspects. A list of the publications follows below, with a brief explanation of each article attached in Appendix B.

Gericke, G.A. and Luwes, N. 2017. Proposing the need for a protocol standard for Internet of Things: Internet of Things water heater case study. In *2017* Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech) (pp. 128-132). IEEE.

Gericke, G.A., Vermaak, H. and Kurakose, R.B. 2019. Communication Protocol Review for SMART Manufacturing Units in a Cloud Manufacturing Environment. In 2019 International Conference on Fourth Industrial Revolution (ICFIR) (pp. 1-6). IEEE.

Gericke, G.A., Kuriakose, R.B., Vermaak, H.J. and Madsen, O. 2020. Machine to Machine Communication Protocol for SMART Manufacturing Units. In *Journal of Physics: Conference Series* (Vol. 1577, No. 1, p. 012047). IOP Publishing.

Gericke, G.A., Kuriakose, R.B., Vermaak, H.J. and Mardsen, O. 2019. Design of Digital Twins for Optimization of a Water Bottling Plant. In IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society (Vol. 1, pp. 5204-5210). IEEE.

Gericke, G.A., Vermaak, H.J., Kuriakose, R.B. and Madsen, O. 2020. The Impact of Communication Protocols in SMART Manufacturing and Their Benefits. *International Journal of Simulation Systems, Science & Technology*, 21(22.1-22.6).

Jardine, N., Gericke, G.A., Kuriakose, R.R. and Vermaak, H.J. 2019. Wireless SMART Product Tracking using Radio Frequency Identification. In 2019 IEEE 2nd Wireless Africa Conference (WAC) (pp. 1-6). IEEE.

References:

[1] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H., 1999. Reconfigurable manufacturing systems. CIRP annals, 48(2), pp.527-540.

[2] Edmondson, N.F. and Redord, A.H. 2002. Generic Flexible Assembly System Design. *Assembly Automation* 22:139-152.

[3] Wang, W. and Koren, Y. 2013. Design Principles of Scalable Reconfigurable Manufacturing Systems. *IFAC Proceedings Volumes* 46:1411-1416.

[4] Mehrabi, M.G. Ulsoy, A.G. and Koren, Y. 2000. Reconfigurable Manufacturing Systems: Key to Future Manufacturing. *Journal of Intelligent Manufacturing* 11:403-419.

[5] Anderson, L. Rösiö, C., Bruch, J. and Jackson, M. 2016. Reconfigurable Manufacturing - An Enabler for a Production System Portfolio Approach. *Procedia CIRP* 52:139-144.

[6] Goyal, K.K., Jain, P.K. and Jain, M. 2013. A Novel Methodology to Measure the Responsiveness of RMTs in Reconfigurable Manufacturing System. *Journal of Manufacturing Systems* 32:724-730.

[7] Koren, Y. 2006. General RMS Characteristics. Comparison with Dedicated and Flexible Systems. In *Reconfigurable Manufacturing Systems and Transformable Factories*. Berlin, Heidelberg: Springer, pp. 27-45.

[8] Koren, Y. 2010. *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems* (vol. 80). Hoboken: John Wiley & Sons.

[9] Lien, T. and Rasch, F. 2001. Hybrid Automatic-Manual Assembly Systems. *CIRP Annals-Manufacturing Technology* 50:21-24.

[10] Tenhiälä, A. and Helkiö, P., 2015. Performance effects of using an ERP system for manufacturing planning and control under dynamic market requirements. Journal of Operations Management, 36, pp.147-164.

[11] Krüger, J., Lien, T.K. and Verl, A. 2009. Cooperation of Human and Machines in Assembly Lines. *CIRP Annals - Manufacturing Technology* 58:628- 646.

[12] Gyulai, D., Vén, Z., Pfeiffer, A., Váncza, J. and Monostori, L. Matching Demand and System Structure in Reconfigurable Assembly Systems. *Procedia CIRP* 3:579-584.

[13] Takata, S. and Hirano, T. 2011. Human and Robot Allocation Method for Hybrid Assembly Systems. *CIRP Annals - Manufacturing Technology* 60:9-12.

[14] Lotter, B. and Wiendahl, H.P., 2009. Changeable and reconfigurable assembly systems. In Changeable and reconfigurable manufacturing systems (pp. 127-142). Springer, London.

[15] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G, Ulsoy, G. *et al*. 1999. Reconfigurable Manufacturing Systems. *CIRP Annals-Manufacturing Technology* 48:527-540.

[16] Minnich, D. and Maier, F. 2006. Supply Chain Responsiveness and Efficiency: Complementing or Contradicting Each Other?. Bruchsal: School of Business Administration.

[17] Setchi, R.M. and Lagos, N. 2004. Reconfigurability and Reconfigurable Manufacturing Systems - State of the Art Review," in 2nd IEEE International Conference of Industrial Informatics: Collaborative automation - One Key for Industrial Environments, Berlin.

[18] Wiendahl, H.P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.H., Duffie, N. *et al*. 2007.Changeable Manufacturing-Classification, Design and Operation. *CIRP Annals-Manufacturing Technology* 56:783-809.

[19] Rooker, M.N., Sunder, C., Zoiltl, A., Hummer, O, and Ebenhofer, G. 2007. Zero Downtime Reconfiguration of Distributed Automation Systems. 2007. In Proceeding of the 3rd International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Regensburg, 2007, pp. 326-337.

[20] ElMaraghy, H. 2006. Flexible and Reconfigurable Manufacturing Systems Paradigms. *International Journal of Flexible Manufacturing Systems* 17:1-13.

[21] Koren, Y. and Shpitalni, M. 2010. Design of Reconfigurable Manufacturing Systems. *Journal of Manufacturing Systems* 29:130-141.

[22] Katz, R. 2007. Design Principles of Reconfigurable Machines. *The International Journal of Advanced Manufacturing Technology* 34:430-439.

[23] Hentea, M. 2008. Improving Security for SCADA Control Systems. *Interdisciplinary Journal of Information, Knowledge, and Management* 3:73-86.

[24] Stouffer, K., Falco, J. and Scarfone, K. 2011. "Guide to Industrial Control Systems (ICS) Security. *NIST Special Publication*, pp. 800-882.

[25] Daneels, A. and Salter, W. 1999. What is SCADA? In International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste, Italy, pp. 339-343.

[26] Van der Knijff, R.M. 2014. Control Systems/SCADA Forensics, what's the Difference? *Digital Investigation* 11:160-174.

[27] Kumar, R., 2010. Recent Advances in SCADA alarm System. International Journal of Smart Home, 4(4), pp.1-10.

[28] Shalini, S.K., Teshome, B., Muluneh, S. and Aragaw, B. 2013. Working Phases of SCADA System for Power Distribution Networks. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 2:2037-2043.

[29] Zhang, P. 2008. *A Handbook for Engineers and Researchers: Norwhich, New York, USA. William Andrew*.

[30] Shaw, W.T. 2006. *Cybersecurity for SCADA Systems*. Oklahoma, USA: Pennwell Books.

[31] Radvanovsky, R. and Brodsky, J. 2016. *Handbook of SCADA/Control Systems Security*. Second edition. Raton, Florida, USA: CRC Press.

[32] Hamed, B. 2012. Implementation of Fully Automated Electricity for Large Building Using SCADA Tool like LabVIEW. *Current Trends in Technology and Sciences* 1: (1-7)

[33] Karnouskos, S. and Colombo, A.W. 2011. Architecting the Next Generation of Service-Based SCADA/DCS System of Systems. In IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society, pp. 359-364.

[34] Alcaraz, C., Roman, R., Najera, P. and Lopez, J. 2013. Security of Industrial Sensor Network-Based Remote Substations in the Context of the Internet of Things. *Ad Hoc Networks* 11:1091-1104.

[35] Roman, R., Najera, P. and Lopez, J. 2011. Securing the Internet of Things. *IEEE Computer* 44:51-58.

[36] Ivert, L.K. 2009. Advanced Planning and Scheduling Systems in Manufacturing Planning Processes. Department of Technology Management and Economics, Chalmers University of Technology, Sweden.

[37] Stadtler, H. and Kilger, C. 2005. *Supply Chain Management and Advanced Planning - Concepts, Models, Software and Case Studies*. Third edition. Berlin, Germany. Springer.

[38] Stadtler, H. 2005. Supply Chain Management and Advanced Planning—Basics, Overview and Challenges. *European Journal of Operational Research* 163:575-588.

[39] Ivert, L.K. 2012. Use of Advanced Planning and Scheduling (APS) Systems to Support Manufacturing Planning and Control Processes, Chalmers University of Technology.

[40] Tenhiälä, A. and Helkiö, P., 2015. Performance effects of using an ERP system for manufacturing planning and control under dynamic market requirements. Journal of Operations Management, 36, pp.147-164.

[41] Li, K., Zhang, X., Leung, J. Y.-T and Yang, S.-L. 2016. Parallel Machine Scheduling Problems in Green Manufacturing Industry. *Journal of Manufacturing Systems* 38:98-106.

[42] Pandhi, N., Singh, K, and Singh, S. 2015. A Contrast: VSM, JIT, and MRP-II. *International Journal of Current Engineering and Technology* 5:385-396.

[43] Tenhiälä, A. and Helkiö, P., 2015. Performance effects of using an ERP system for manufacturing planning and control under dynamic market requirements. Journal of Operations Management, 36, pp.147-164.

[44] Hexatec. 2010. *How to Design a Good HMI Display* [White Paper]. Available: http://www.hexatec.co.uk/Consultancy/hmi_display_design_guidelines.aspx (Retrieved on 04/05/2020)

[45] Gersztenkorn, D. and Lee, A.G. 2015. Palinopsia Revamped: a Systematic Review of the Literature. *Survey of Ophthalmology* 60:1-35.

[46] Deodhar, S., Agrawal, P. and Helekar, A. 2014. Effective Use of Colors in HMI Design. *International Journal of Engineering Research and Applications* 4:384-387.

[47] Roessler, D. and Garrison, L. 2013. *Five Practical Elements of Effective SCADA Graphics* [White paper]. Available: https://pgjonline.com/2013/02/19/five-practical-elements-of-effective-scada-graphics/ (Retrieved on 01/03/2020)

[48] OPTO 22. 2014. *Building an HMI that Works: New Best Practices for Operator Interface Design* [White paper]. Available: http://www.opto22.com/site/documents/doc_drilldown.aspx?aid=4351 (Retrieved on 06/06/2020)

[49] Gruhn, P. 2011. Human Machine Interface (HMI) Design: The Good the Bad and the Ugly (and what makes them so). 66th Annual Instrumentation Symposium for the Process Industries, 27-29 Jan 2011.

[50] Hollifield, B. 2013. High Performance HMI–Proof Testing in Real-World Trials. In 2013 ISA Water/Wastewater and Automatic Controls Symposium, Orlando, Florida, USA, 2013.

[51] Hollifield, B., Oliver, D., Nimmo, I. and Habibi, E. 2008. *The High Performance HMI Handbook: A Comprehensive Guide to Designing, Implementing and Maintaining Effective HMIs for Industrial Plant Operations*. First edition. Manama, Bahrain. Plant Automation Services.

[52] Mughal, A., Ali, L., Aziz, N., Mehmood, K. and Afzal, N. 2013. Colour Vision Deficiency (CVD) in Medical Students. *Pak J Physiol* 9:1-9.

[53] Hollifield, B. 2012. The High Performance HMI: Better Graphics for Operations Effectiveness. In 2012 Water/Wastewater and Automation Controls Symposium, Orlando, Florida, USA, 2012.

[54] Krajewski, J. 2014. Situational Awareness - *The Next Leap in Industrial Human Machine Interface Design*. White paper, Invensys Systems, Houston, USA, 2014.

[55] Pokharel, B. 2013. Machine Vision and Object Sorting: PLC Communication with LabVIEW using OPC. (Unpublished thesis.) Department in Automation Engineering, HAMK University of Applied Sciences.

[56] Travis, J. and Kring, J. 2006. *LabVIEW for Everyone: Graphical Programming Made Easy and Fun*. Third edition. New Jersey: Prentice Hall.

[57] Bishop, R.H. 2007. *LabVIEW 8 Student Edition*. Austin, Texas: Pearson Prentice Hall.

[58] Sabu, H.M., Aravind, V.B., Sullerey, A. and Binson, V.A., 2015. Online Monitoring of PLC Based Pressure Control System. International Journal of Research and Innovations in Science and Technology, 2(2), pp.47-50.

[59] Muftah, M.A., Albagul, A.M. and Faraj, A.M., 2014. Automatic paint mixing process using LabView. Mathematics and Computers in Science and Industry, pp.233-238.

[60] Tomić, J., Kušljević, M. Vidaković, M. and Rajs, V. 2014. Smart SCADA System for Urban Air Pollution Monitoring. *Measurement* 58:138-146.

[61] Niemann, J. 2013. Development of a reconfigurable assembly system with enhanced control capabilities and virtual commissioning. (Unpublished Master's dissertation.) Bloemfontein: Central University of Technology, Free State, Faculty of Engineering and Information Technology.