# A MAC LAYER PROTOCOL FOR SMART INDOOR INVENTORY MANAGEMENT SYSTEM

By

Xiaofeng Li

Bachelor of Science in Engineering –University of Saskatchewan, Saskatoon, Canada, June 2017

A thesis presented to Ryerson University

in partial fulfillment of the requirements for the degree of

Master of Applied Science

in the program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2020

## AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

# ABSTRACT

Xiaofeng Li
Master of Applied Science
Electrical and Computer Engineering
Ryerson University, Toronto, Canada, 2020

The indoor inventory system is gaining more research attention and commercial value with the development of IoT. In this thesis, we presented the design of a MAC protocol that allows synchronized transmission of location and sensing data in a wireless positioning and sensor network for an indoor inventory system. The network supports real-life industrial applications and provides a highly specific positioning method.

In the network, mobile sensing tags are connected to smart readers that performs localization of tags and gathers sensing data from the tags. The readers are connected to the back-end cloud. The proposed MAC serves multiple classes of mobile tags with different priorities and latency requirements. These tags transmit critical, position and sensing data with different QoS requirements. The proposed MAC is a hybrid MAC that offers contention-based period for tag discovery and scheduled period for the transmission of sensing data with guaranteed latency. We conducted simulation to evaluate the performance of different methods of discovery process and their impact on latency assurance. We also developed a queuing model to analyze the relationship between parameters, acquiring parameters through experiment, and calculation of boundary values.

Simulation using MatLab™ software suggests that the joining period in design can increase the transmission success rate of high priority messages at the cost of a slight increment in the delay of low priority messages. Preliminary analysis suggests that by adaptively allocating the channel resources of the network to three types of tags, service efficiency can be improved. This result also guides the direction for further improvement.

We explored the performance of two options considered currently, which is selecting the discovery process according to modulo result of unique 16-bit tag ID and random select of an available discovery process. In the current environment where each tag does not have any information about other tags inside the network, the two methods have the same effect on avoiding collisions that could happen in a single discovery cycle.

The proposed MAC layer protocol can provide the best service when the available discovery process in the discovery cycle is for initialization and resetting. For an emergency, the joining period designs can still ensure a success rate for critical messages to be over 90%. Hence, the simulation results indicate the joining period method is able to improve MAC-layer performance.

ACKNOWLEDGMENTS

The author would like to take this opportunity to express my deep gratitude to my supervisors Prof. Xavier Fernando and Prof. Muhammad Jaseemuddin, who have offered me constructive guidance, precious advice, and encouragement during this thesis work. All accomplishments and research results impossible without them.

I would like to acknowledge the members of PeyTec, Mr. Peyman Moeini, Mr. Akbar Ahmari, and Mr. Anand Sinha for their valuable discussions, technical input, and useful suggestions. Special thanks go to Mr. Akbar Ahmari, who provided the most experimental parameters of this thesis.

I would like to thank other graduate students in our lab, Ahasanun Nessa, Ryan Murari, and Manpreet Singh, for offering various advice and helps during this thesis work.

Finally, I would like to thank my family for all the love and support though the past years.

# Table of contents

**Abstract** ........ iii

**List of tables** ........ vii

**List of figures** ........ viii

**Abbreviations** ........ ix

Chapter 1 Introduction ........ 1
- 1.1 IoT based inventory management system ........ 1
- 1.2 Related research about IoT ........ 2
- 1.3 Research challenges ........ 4
- 1.4 Thesis outline ........ 5

Chapter 2 Backgrounds ........ 6
- 2.1 Network introduction ........ 6
- 2.2 Positioning method ........ 8
  - 2.2.1 Two way ranging method using three messages ........ 8
  - 2.2.2 Indirect positioning method used in the project ........ 9
- 2.3 Related medium access control protocol ........ 10
  - 2.3.1 IEEE 802.15.4 MAC layer—CSMA/CA protocol ........ 10
- 2.4 Literature review ........ 13
  - 2.4.1: IEEE 802.15.4 Low Latency Deterministic Network (LLDN) ........ 13
  - 2.4.2: Related works ........ 14
- 2.5 Chapter summary ........ 17

Chapter 3 Analysis with queuing model ........ 18
- 3.1 Resources Requirements Analysis ........ 18
- 3.2 Modelling on the System ........ 19
- 3.3 Resources Allocation Analysis in the Queuing Model ........ 20
- 3.4 Simplification of the Queuing model ........ 22
- 3.5 Chapter summary ........ 25

Chapter 4 Design of MAC layer ........ 26
- 4.1 Overview of the system network ........ 26
  - 4.1.1 Network devices ........ 26
  - 4.1.2 Messages in the network ........ 28
  - 4.1.3 Network cycles: discovery and positioning ........ 29
- 4.2 Analysis of boundary values and parameters ........ 31
  - 4.2.1 Slot time calculation ........ 31
  - 4.2.2 Initial boundary value calculation: ........ 32
  - 4.2.3 Tag joining and Data arrival ........ 34

4.3 Possible improvements ........ 35
4.4 MAC Algorithm ........ 38
4.5 Chapter summary ........ 40
Chapter 5 Simulation results ........ 41
5.1 Simulation background ........ 41
5.2 Performance metrics ........ 43
5.3 Random process simulation ........ 43
5.4 Joining period in positioning cycle ........ 48
5.5 Chapter summary ........ 52
Chapter 6 Conclusions and future works ........ 53
6.1 Conclusion ........ 53
6.2 Future works ........ 53
6.2.1 Analysis Model ........ 54
6.2.2 Unexpected communication failures ........ 54
6.2.3 Retry time limits ........ 54
6.2.4 Change in design backgrounds ........ 54
Bibliography ........ 55

# List of tables

Table.3.1 Variables in the queuing model part 1 .......... 21
Table.3.2 Variables of the queuing model part 2 .......... 21
Table 4.1 Parameters in slot time calculation .......... 31
Table 4.2 Max delay tolerance for message types .......... 32
Table 4.3 Variables for initial boundary value calculation .......... 32
Table 4.4 Number of expected tags per second .......... 34
Table 4.5 Variable list of tag joining type analyze .......... 35
Table 4.6 Variable list for random initial method analyze .......... 36
Table 5.1 Parameters of simulation .......... 42

# List of figures

Figure 1.1 Internet of Things application scenario ... 1
Figure 2.1 Architecture of the Proposed Network ... 6
Figure 2.2a Two-way ranging using three messages ... 8
Figure 2.2b Indirect positioning method ... 9
Figure 2.3 BE Mode Super-Frame Structure ... 10
Figure 2.4 Flowchart of CSMA/CA ... 11
Figure 2.5 Super-frame structure of DRTLS ... 12
Figure 3.1 Data transmissions in the network and the abstract view ... 18
Figure 3.2 The main queuing model ... 19
Figure 3.3 The queuing sequence of model ... 20
Figure 3.4 QoS time vs. Service rate plot for different arrival rates ... 23
Figure 3.5 Queue space size vs service rate plot for different arrival rates ... 24
Figure 4.1 General network diagram ... 27
Figure 4.2 Structure of the discovery cycle ... 29
Figure 4.3 Structure of positioning cycle ... 30
Figure 4.4 Flow chart for tags ... 38
Figure 4.5 flow chart for master reader ... 39
Figure 5.1 (a) TSR of the critical messages in one DC, AVAP 2 to 1000 ... 44
Figure 5.1 (b) TSR of the critical messages in one DC, AVAP 2 to 100 ... 44
Figure 5.2 (a) TSR of non-critical messages in one DC, AVAP 2 to 1000 ... 45
Figure 5.2 (b) TSR of non-critical messages in one DC, AVAP 2 to 100 ... 45
Figure 5.3 (a) Average delay of critical data transmission in one DC, AVAP 2 to 1000 ... 46
Figure 5.3 (b) Average delay of critical data transmission in one DC, AVAP 2 to 100 ... 46
Figure 5.4 (a) Average delay of non-critical data transmission in one DC, AVAP 2 to 1000 ... 47
Figure 5.4 (b) Average delay of non-critical data transmission in one DC, AVAP 2 to 100 ... 47
Figure 5.5 Average success rate for CM in one pair of DC and PC, AVAP from 2 to 100 ... 48
Figure 5.6 TSR for sensor messages in one pair of DC and PC, AVAP from 2 to 1000 ... 49
Figure 5.7 (a) TSR for positioning messages in one pair of DC and PC, AVAP from 2 to 1000 ... 49
Figure 5.7 (b) TSR for positioning messages in one pair of DC and PC, AVAP from 50 to 200 50
Figure 5.8 (a) Average delay for critical messages in a pair of DC and PC, AVAP from 2 to 1000 ... 50
Figure 5.8 (b) Average delay for critical messages in a pair of DC and PC, AVAP from 2 to 100 ... 51
Figure 5.9 Average delay for positioning messages in a pair of DC and PC ... 51

# Abbreviations

| | |
|---|---|
| *ACK* | Acknowledgement signal |
| *AVAP* | Available Process |
| *BE* | Beacon-Enabled mode |
| *BI* | Beacon Interval |
| *BM* | Beacon Message |
| *BO* | Beacon Order |
| *CAP* | Contention Access Period |
| *CFP* | Contention Free Period |
| *CM* | Critical Message |
| *CSF* | Configuration Status Frame |
| *CSMA/CA* | Carrier Sensing Multiple Access with Collision Avoidance |
| *DC* | Discovery Cycle |
| *DP* | Discovery Process |
| *DRF* | Discovery Response Frame |
| *DRM* | Discovery Response Message |
| *DRTLS* | Decawave Real Time Localization System |
| *EE-MAC* | Emergency Enabled Medium Access Control |
| *FFD* | Full Function Device |
| *GACK* | Group Acknowledgement signal |
| *GPS* | Global Positioning System |
| *IoT* | The Internet of Things |
| *LLDN* | Low Latency Deterministic Network |
| *MAC* | Medium Access Control |
| *MWBM* | Maximum Weighted Bipartite Matching |
| *NBE* | Non-Beacon-Enabled mode |
| *PAN* | Personal-Area Network |
| *PC* | Positioning Cycle |
| *PM* | Positioning Message |
| *PP* | Positioning Process |
| *PriMulA* | Priority-Aware Multi-Channel Adaptive framework |
| *QoS* | Quality of Service |
| *RFD* | Reduced Function Device |
| *RFID* | Radio Frequency Identification |
| *SM* | Sensor Message |
| *SO* | Super-frame Order |
| *TDOA* | Time Difference Of Arrival |
| *TSCH* | Time Slotted Channel Hopping |
| *TSR* | Transmission Success Rate |
| *TWR* | Two-Way-Ranging |

# Chapter 1 Introduction

With the development of computer hardware, software and wireless communication technology, the Internet of Things (IoT) is advancing quickly in recent years. Various applications of the Internet of Things have also appeared. Among them, the inventory management system is gaining research attention and commercial values. An IoT based inventory management system can boost the efficiency of logistics which is one of the key factors for most companies to achieve success. This chapter introduces research challenges for IoT based inventory system and our research contributions.

## 1.1 IoT based inventory management system

An IoT based application must solve the fundamental problems of item identification, item geographical location information, item related data, sensor data, security, and other information acquisition and network transmission. Besides these, along with the application of cloud computing and artificial intelligence technology, IoT can perform smart identification, positioning, tracking or managing much quicker because the devices are driven by an on-device machine learning systems, such as Google Tensor flow Lite or backend cloud machine learning and artificial Intelligent decision service.

Figure 1.1 Internet of Things application scenario

(https://www.csoonline.com/article/3346082/what-is-shadow-iot-how-to-mitigate-the-risk.html)

An inventory management system is an indispensable part of enterprises and crucial to the decision-makers and managers of the enterprise. It can be widely applied to wholesale, retail, and production commercial enterprises, stores, stores, warehouses, etc., and integrates the purchase, sales, inventory, financial collection and payment, and customers Its main functions are: inbound management, outbound management, collection and payment management, commodity data management, user information, and customer data management, other income management, expenditure management, as well as various ledger inquiries and other functions. Applying IoT technology to inventory management and building a smart inventory management system is one of the practical applications and a favored research direction in the fields of both the Internet of Things and inventory management.

The IoT based smart inventory management system is expected to offer the following services:

- Positioning service. This service is for indoor cargo item, personnel and handling vehicle positioning;
- Sensor data service. This service is for on-spot environmental monitoring, such as data picking for temperature, humidity, luminance, etc.
- Warning message service. Warning message generating service, this is for message alarming transmission, an alarming message can be that items leave the shelf or people are too close to the forklift, smoke warning message which indicates a potential danger of fire, a high humidity warning message that may be a water leak, etc.).
- Heterogeneous multi-wireless network protocol transmission service. Wireless networks can be Bluetooth, WIFI, UWB, etc.

## 1.2 Related research about IoT

Currently, there are many related research works on the Internet of Things and inventory management systems, most of which focus on the application level. Many good applications have been considered as references to the project design.

Oguntala, G., et al. [1] carried out an inclusive survey on indoor technologies and techniques drawing attention, introducing benefits, limitations, and areas for improvement of different RF-based technologies including Wi-Fi, RFID, UWB, Wi-Fi, Bluetooth, ZigBee, and Light over other indoor technologies for reliable IoT-based applications. It also provides formulas for analysis of simple localization problems. An evaluation of performance about the indoor technologies discussed is provided with a metric of scalability, accuracy, complexity, robustness, energy-efficiency, cost, and reliability. Riad M., et al. [2] provide reviews research and development efforts in the utilization of IoT for the management of perishable inventory in recent years. An analysis of the studied research reveals the opportunities that are not yet fully realized and the challenges that need to be efficiently solved to enable their realization.

Jing, X. and P. Tang [3]first introduced characteristics and basic application of RFID technology, then analyzed data flow of intelligent inventory system from the perspective of business and function. After that, they introduced framework programs and function modules of a smart inventory management system based on IOT RFID technology to enhance understanding and prepare for the main topic. The article focuses on elaborating on the design and process of implementation of an intelligent inventory system. The system realizes full control and management of all products, faster in/out the warehouse, and dynamic inventory utilizes warehouse efficiently and improves the capacity of the warehouse by effective combining with the ERP system in an enterprise. Wang, M. et al. [4] adopts UHF radio frequency identification (RFID) based on IoT technology, in detail, and developed the software system using 902 MHz RFID tags with the UHF antenna system. Via one-dimensional, two-dimensional code and RFID technology, realizing the intelligent management and monitoring of the air materiel's entry, export, shift, and inventory of warehouse, meanwhile, reducing inventory management men's work intensity and improving inventory management level.

Zhang, L., et al. [5] proposes an inventory management system for a warehousing company. The system integrates RFID technology and a self-Adaptive distributed decision support model for inbound and outbound actives, inventory location suggestions, and incident handling. The model consists of three major components: environment recognition, knowledge merging, and decision making. Besides, a 'self-adaptive' feature is adopted for adjusting the knowledge used in the decision making procedure. An experiment is also outlined to validate the utilization of our model and the proposed system. Tejesh, B. S. S. and S. Neeraja et al. [6] presented a warehouse inventory management system using IoT and open-source framework. The warehouse inventory management system built on the architecture of the Internet of Things is developed to track the products attached to the tags with product information and their respective time stamps for further verification. The total system gives an archetype to correspond to the information flow and material flow. The web page is built in accordance with providing convenience and an interface to the user to track the products. The developed system results in a very low-cost system and works dynamically compared with the existing present warehouse inventory management systems[7].

Chao, W., et al. [8]proposes to establish a power Internet of Things intranet through wireless communication Internet of things technology with low power and wide coverage (LPWAN), combining mobile terminals and NB-IOT low power WAN. By using the active Internet of things (IoT) electronic tags and radio frequency read-write device, the asset electronic tags identified by the front end are transmitted to the IoT application network, realizing the interconnection between the sensing layer and the application layer, and verifying the accuracy and convenience of the wireless inventory of power grid assets. It has greatly reduced the amount of manpower and comprehensively supported the unified linkage of asset and equipment management. Rholam, O., et al. [7] presented a communication model based on Modbus and multiband communication using NB-IoT, this communication model allows data to be exchanged between different industrial equipment (PLCs, SCADA systems, etc.) via MODBUS RTU and ASCII, via RS485, or analog and digital inputs for sensors and actuators.

Wang, Y., et al. [9]. consider supply chains consisting of multiple suppliers, a manufacturer, and multiple distributors. The time cost of delayed transportation is integrated into previous studies to construct a new model, which is solved with an immune genetic algorithm. Unlike the genetic algorithm, the memory function and adjustment function of the immune algorithm is included in this algorithm. Different from the immune algorithm, genetic operators of the genetic algorithm are included. The immune genetic algorithm effectively overcomes the disadvantages of the genetic algorithm, improving global searchability and search efficiency. The validity and rationality of the optimized model are assessed in comparison with the previous results[10].

Although many applications have been commercialized and played a large role in real life, related research has focused on indoor positioning and the use of existing network technologies to complete data transmission, thereby achieving application integration. There are not many related kinds of research on the underlying network protocols related to applications, especially media access control protocols.

## 1.3 Research challenges

The MAC protocol design presented in this thesis was developed for the wireless positioning and sensing network (WPSN) proposed in [11]. The WPSN was a collaborative work with PeyTec. to build an industrial automation network including an indoor inventory system network. In the network, mobile sensing tags are connected to smart readers that perform localization of tags and gather sensing data from the tags. The readers are connected to the back-end cloud. The cloud server can handle all the calculations, including the scheduling, assigning roll of each reader, and the localization calculation.

The positioning method of this network does not require the tag to do any localization calculation, anchor selecting, or sending a group poll signal. The localization process is achieved by two broadcasts from the tag and one response from one of the readers.

The MAC protocol for communication between tags and readers needs to support multiple classes of mobile tags with different priorities and latency requirements. These tags transmit critical, positioning and sensing data with different QoS requirements. There is a synchronization requirement between positioning and emergency data transmission, which is not supported by an existing medium access control (MAC) protocol, including LLDN in IEEE 802.15.4e, DTRLS MAC, and other related MAC protocol for the positioning system. Hence, the IEEE 802.15.4e and its variations need modifications to fulfill the synchronization requirement.

This thesis is focused on **designing a MAC layer protocol between the readers and tags to support an indoor smart inventory management system.** It is inspired by IEEE 802.15.4e LLDN to support the following features.

1. Mobility of objects: Tags can move with a possible speed of any object inside a real-life factory environment.
2. Monitoring of the environment: With various sensors, tags should be able to monitor the environment around them. Tags may transmit sensing data periodically or critical data like an alarm signal when potential danger is detected.
3. Combination of information: in the network, multiple types of data acquired from the sensors are combined to provide details of real-life environments. One example is the combination of positioning data with critical data, which allows alert messages in a certain position.
4. The scale of operation: Operate inside the industrial environment with large scale implantation. The industrial environment means a large number of metal objects will be inside the localization area, which can move at any moment.

Communication between the readers and servers is using Wi-Fi, which can handle large size data transmission with low latency and does not involve in the main topic.

The MAC layer should be able to provide:

1. Quick joining method and hand-off mechanism to support tag mobility. Quick joining is a key concern when handling moving tags; the hand-off mechanism is for any tag that is

moving with high speed or lost communication with all readers during moving.

2. Guaranteed low latency data transmission. If contention exists for sending any data packet, period sensing data will consume more energy, and critical data may be delayed and have serious consequences. Thus contention-free transmission is necessary.
3. Handle frequent communication failure. Inside the industrial environment moving metal obstacles can block signals and cause unexpected communication failure, the MAC should be able to handle it.

## 1.4 Thesis outline

Chapter 1 contains the background of the thesis. IoT, indoor inventory system, the research challenges, and related works are discussed. The goal of the thesis, design a MAC layer protocol supporting a project from PeyTec. are included in research challenges with other details.

Chapter 2 starts with a brief introduction of the project network, followed by discussions about the positioning method, related MAC layer protocols, and related works done in the area studying the MAC layer.

Chapter 3 builds queuing models to analysis how to achieve maximum efficiency for the project network. The analysis conclusion is used to guide design in chapter 4.

Chapter 4 explains the detailed design of the MAC layer protocol of the project network. Two possible improvements of design which is random initial process selection and joining period presented.

Chapter 5 provides MatLab™ simulations for the designs explained in chapter 4. The result of simulations suggests that joining period design can improve network performance and the

Chapter 6 concludes the project and list possible directions for future works. Future works are consist of various research directions found during thesis work.

# Chapter 2 Backgrounds

This chapter provides background information. The first section of this chapter gives more detail about the proposed network and, explains the positioning method involved in the network. Then it introduces two MAC layer protocols designs based on LLDN and DRTLS. Finally, it gives a literature review of the related work done on the MAC layer.

The overall objective of the proposed network is to track hundreds of tags with a few centimetres of accuracy in real time with very low power consumption in the tags. As we will see later-on there are many challenges in achieving this objective especially when there are a large number of tags and a lot of obstructions between tags and readers.

## 2.1 Network introduction

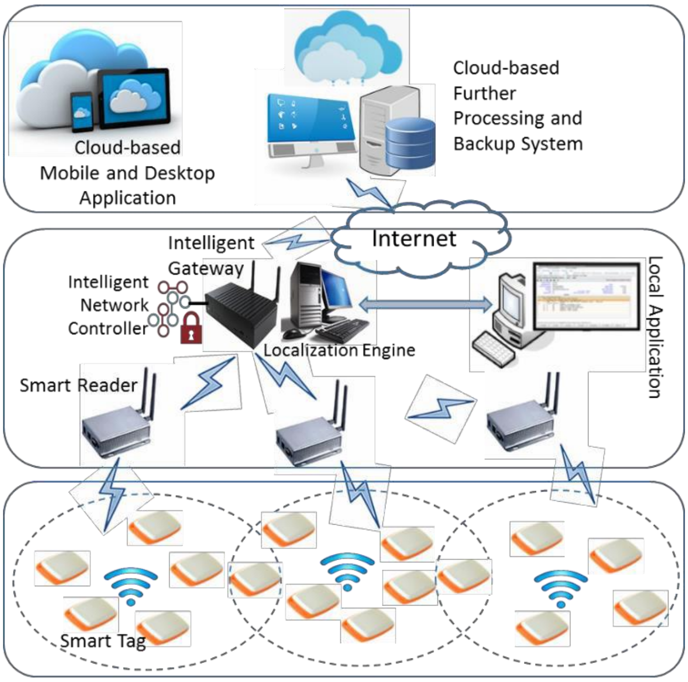



Figure 2.1 Architecture of the Proposed Network

The network under consideration is shown in Figure 2.1. It contains three types of devices: server, readers, and tags[11].

**The Server:** The server (or the Intelligent Gateway) is the main hub of the network. It is capable of more complex computation and communicates with the cloud via wired or 5G cellular wireless links for cloud-based further processing and back-up. It would be responsible for coordinating and, scheduling the events while performing localization calculations. It is assumed to have infinite computation power in this thesis. The server communicates with readers through a Wi-Fi network, which is assumed to have the capability to handle large size data packets with small transmission delays.

**The Reader:** The (Smart) readers are based on DWM 1001 devices. These readers communicate with the tags through the Ultra-Wideband channel. They also communicate with the intelligent gateway server via Wi-Fi. The readers are equipped with Decawave Ultra-Wideband (UWB) transceiver IC. This IC supports up to 6 channels and enables localization of the tags, using time difference of arrival (TDOA) of UWB signal up to a few centimetres accuracy. The readers are powered from the power grid and have no power issues

**The Tags:** The (Smart) tags are attached to objects that need to be tracked. The tracking algorithm is based on the DWM 1001 networking. Note the tags are powered by batteries and should have simpler hardware design and low power consumption. Tags will be in sleep mode most of the time to reduce energy costs. When tags wake-up from sleep, they will always enter the listening mode and wait for a beacon signal or beacon-like signal to guide their behaviour. A motion sensor is also built inside the DWM 1001, which allows it to sense the change in location and movements. Other sensors may be added to acquire more information about the environment.

In a real-life industry environment, there will be many obstacles, both mobile and stationary, made of metal, wood, plastic, and other materials between the readers and tags. Note each tag needs to communicate with three or more readers to achieve localization. The network should also be able to support the mobility of tags. Both these facts suggest that loss of connection between the readers and tags and rejoining would happen frequently.

For large scale applications, energy consumption becomes a key problem as the tags must communicate more frequently. Other major concerns, such as accuracy and latency, should also be considered. As will be discussed in detail in the positioning method section, DRTLS requires tags to be able to locate themselves including a selection of anchor nodes. For large scale applications, the basic system is not likely to be a good choice because the tags only relay on the battery as the energy source and the positioning process is energy-consuming. Therefore, a new positioning method is proposed in this thesis where the system will let the local server to do the calculations to extend the life span of tags.

## 2.2 Positioning method

### 2.2.1 Two way ranging method using three messages

Two-way ranging (TWR) method is a distance measurement method between two devices using time difference of arrival. There are three types of TWR: single-sided TWR, double-sided TWR, and TWR using three messages These are described in detail in [12].

**TWR using three messages:**

This method has three signal transmission. An example of this method between two devices, device A and device B, is shown below:

The distance between the two devices is calculated by the propagation time using the timestamp. Every device always records the time when it sends the signal using a timestamp. They also record the received signal timestamps. The *Reply time* is the time difference between the send time (using its own timestamp) and the receive time (using the received signal timestamp).

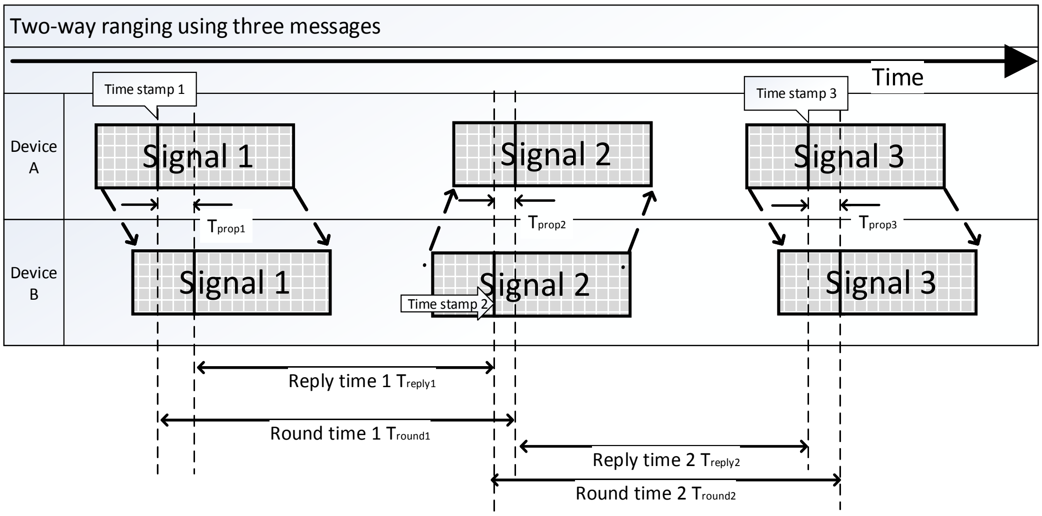



Figure 2.2a Two-way ranging using three messages

The average propagation time can be calculated by the equation:

$\hat{T}_{prop} = \frac{T_{round1} \times T_{round2} - T_{reply1} \times T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}}$, assume the desired frequency for both devices are fq.

In real life situations clock errors are expected for both device:

Real-life frequency of device A is $\mathrm{K}_a \times \mathrm{fq}$;

Real-life frequency of device B is $\mathrm{K}_b \times \mathrm{fq}$; $K_a$ and $K_b$ are close to 1;

The induced clock error of the average value of propagation time can be expressed as:

$$\text{error} = \hat{T}_{prop} \times (1 - \frac{k_a + k_b}{2})$$

## 2.2.2 Indirect positioning method used in the project

In this thesis, we consider the positioning method called indirect positioning method, first proposed by team member Mr. Akbar Ahmari. In this method, the tag does not need to do all the calculations, and the time required to perform the communication for positioning a tag is shortened. However, this method has a lower positioning accuracy as a result which is a trade-off.

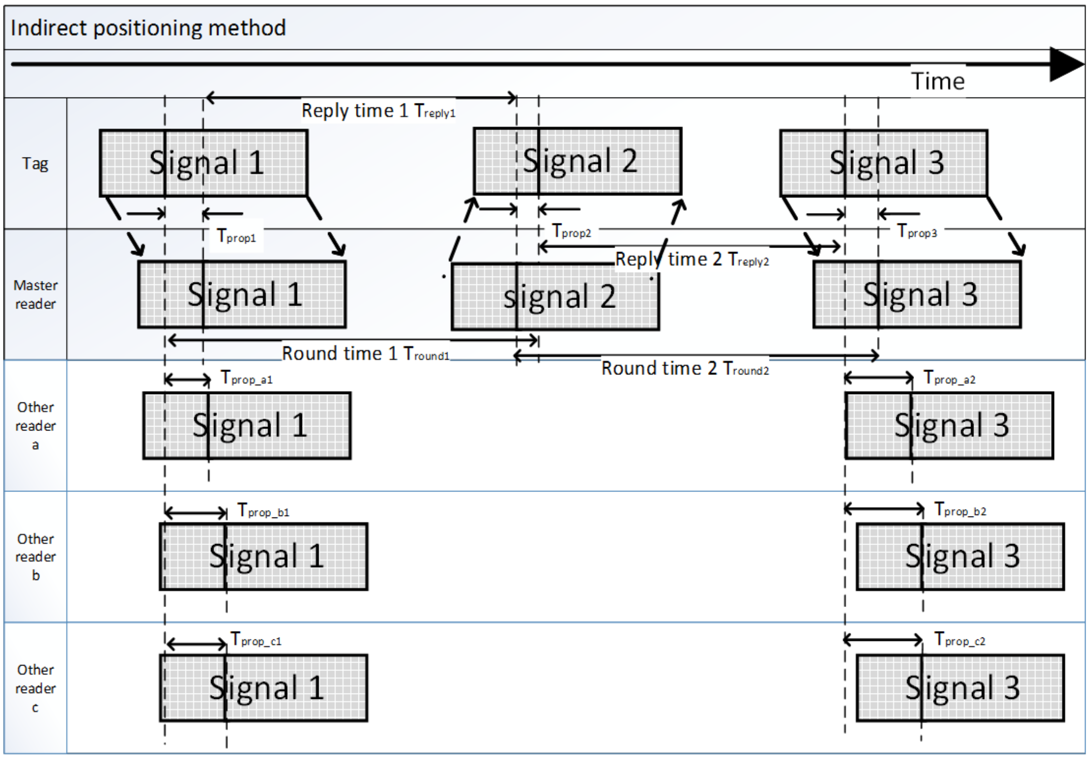



Figure 2.2b Indirect positioning method

This is how this method works. TWR using three messages is applied between the tag and the main reader. Note meanwhile, multiple other readers do overhear the two messages sent by the tag.

These overhearing readers calculate the average of the propagation time of the two overheard messages: $\hat{T}_{prop\_i} = \frac{(T_{prop\_i1}+T_{prop\_i2})}{2}$ this equation does not have any correction to deal with the induced tag clock error, and in the worst case, the distance will have the maximum error possible.

Though this method sacrifices accuracy of positioning, the propagation nature of the ultra-wide-band signal yield only a very small error. This is verified by real-life tests with Decawave devices. The accuracy was not impacted big with this approach.

## 2.3 Related medium access control protocol

### 2.3.1 IEEE 802.15.4 MAC layer—CSMA/CA protocol

IEEE 802.15.4 is a standard for low-rate, low-power, and low-cost WPAN. The design of the IEEE 802.15.4 network includes one personal area network (PAN) coordinator who acts like a 'control center' for a subset of other network nodes.

IEEE 802.15.4 supports three network topologies, which are the star (single-hop), mesh (multi-hop), and hybrid topologies include both star and mesh. This standard defines two device classes, full function device (FFD) and reduced function device (RFD). FFD devices are able to be used for any topology because FFD allows a device to talk to any other device. The full protocol is implanted in an FFD device, which also allows it to perform as a PAN coordinator. The main goal of RFD is to save power at the cost of reduced features. Thus, RFD devices only connect to FFD devices; they are unable to be PAN coordinators, have reduced protocol set, and support only simple implementations.

IEEE 802.15.4 also defines two MAC modes: beacon-enabled (BE) mode and non-beacon enabled (NBE) mode. The design is based on duty-cycles to improve power-efficiency, where the nodes enter 'sleep mode' according to a certain schedule. During the sleep mode, a node will not be able to perform any data transmission and have only a running timer to count the time before it 'wakes up' and have full functionality again.

BE mode uses a super-frame structure, which is shown below:

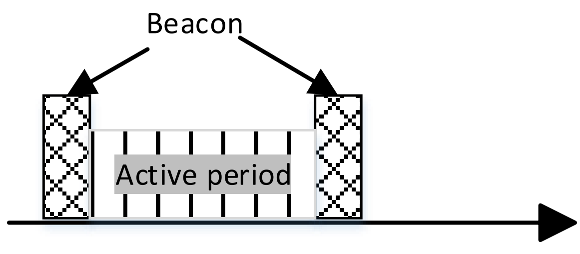



Figure 2.3.1-1 a) Super-frame structure without inactive period

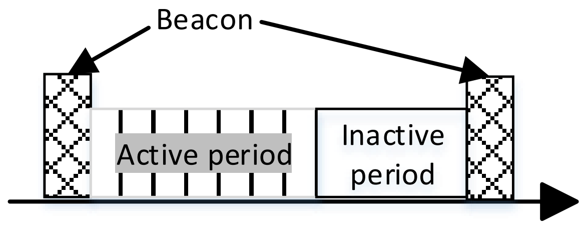



Figure 2.3.1-1 b) Super-frame structure with inactive period

Figure 2.3 BE Mode Super-Frame Structure

As shown in the figure above, each super-frame is scheduled between beacon frames, and each super-frame can have an inactive period, where no transmission will be performed. Beacon frames can be considered as special synchronizing frames, and they are periodically sent by the PAN coordinator nodes. Beacon interval (BI) is defined as the time period between two near beacon frames. The parameter that modifies BI is beacon order (BO), which is ranged from 0 to 14

(BI=15.36*2BO milliseconds). Super-frame interval (SI) is defined as the active time period of the super-frame. SI can be modified by parameter super-frame order (SO), which is between 0 and BO (SI=15.36*2SO milliseconds, 0<SO<=BO<=14) [13].

Inside each active period of super-frame, there is a contention access period (CAP) and a contention-free period (CFP). Inside CAP, the slotted carrier sensing multiple access with collision avoidance (CSMA-CA) algorithm is used. In CFP, a time division multiple access (TDMA) algorithm using a number of guaranteed time slots (GTS) is used, GTS is pre-assigned to individual nodes and have a maximum number of 7. This could be considered as a hybrid algorithm of contention bases method and pulling method [14] [15].

In NBE mode, the super-frame is not included, all the nodes are kept in active mode, and an un-slotted CSMA-CA is used.

The process of transmission of beacon-enabled mode could be shown in the figure below:

Where NB stands for the number of back-offs, CW stands for contention window size, BE stands for the current back-off exponent.

Before any transmission, the node would perform a random back-off. It then performs a clear channel assessment (CCA), which detects whether the channel is clear for data transmission. Under all situations, it will always perform at least two CCA before transmission. Energy consumption due to a large contention window is avoided by having a fixed maximum attempt for a node to try[16].

This method is able to avoid collision under the situations shown below:

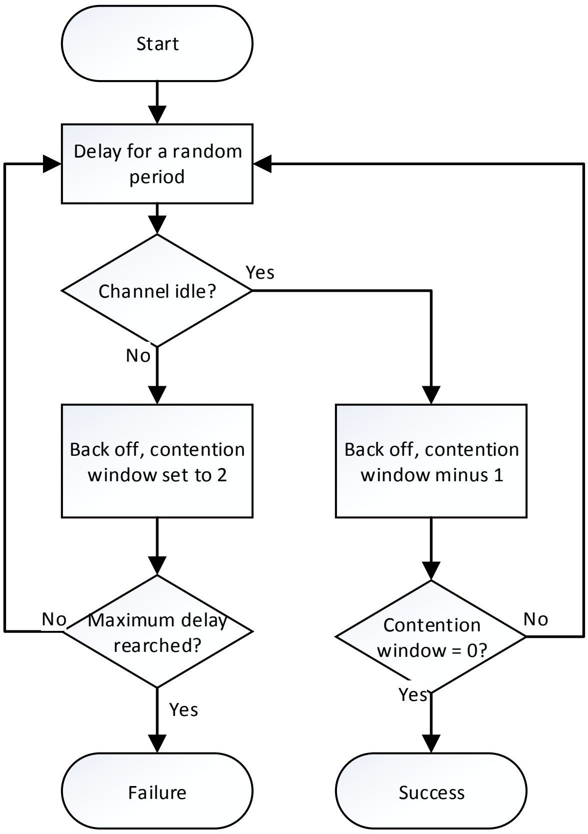



Figure 2.4 Flowchart of CSMA/CA

The figure above suggests the common situation where collision is avoided because CCA is performed during the acknowledgment frame (ACK) or data transmission.

2.3.2 DRTLS protocol

The Decawave real-time localization system (DRTLS) is designed for Decawave devices to support indoor positioning. In DRTLS, there exist anchor nodes that are fixed in location and already known its physical location from the user input. Tags that are able to select four anchor nodes, perform double-sided TWR, each one with the anchor node.

The super-frame structure of DRTLS is shown below:

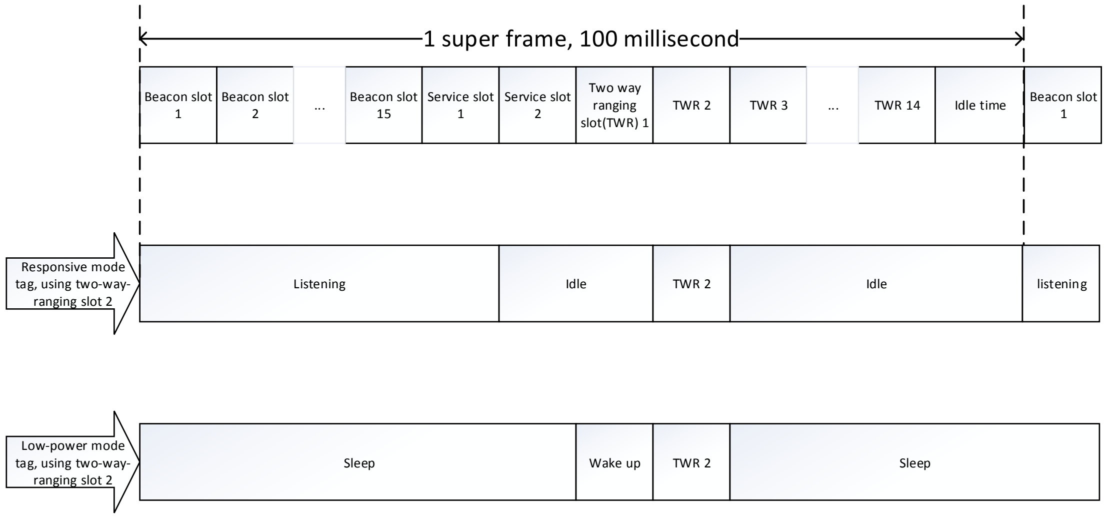



Figure 2.5 Super-frame structure of DRTLS

There can be up to 16 beacon slots, followed by two service slots that have the same function as management slots in LLDN for network management messages. Then followed by multiple TWR slots [17] [18].

This method requires tags in the network to perform anchor node evaluation, apply TWR slot, and send messages to pull the anchor nodes for the positioning process. To realize these functions, each tag is required to perform calculations, which is energy-consuming in the long term, and the max number of tags inside each network is limited. The energy consumption of each tag is also expected to be higher in the long term. Furthermore, the super-frame of the DRTLS is a constant number, 100 milliseconds, as specified for MAC layers of Decawave 1001 device based networks. Thus this MAC layer is not considered the best choice but a useful reference for the project design.

## 2.4 Literature review

### 2.4.1: IEEE 802.15.4 Low Latency Deterministic Network (LLDN)

As the development of IoT in various industrial applications grows, it has been found that though the default IEEE 802.15.4 could provide low latency transmission, it is hard to meet the demand of latency, robustness, and large-scale support. Thus, LLDN was designed with the help of TDMA concept and added to IEEE 802.15.4 to provide low latency deterministic services.

LLDN is a slotted MAC protocol that requires clock synchronization and only supports one-hop (star) topology. The time slot is a small-time period with a fixed length for each LLDN. One advantage of slotted MAC is the potential delay result from one contention has a general time unit, which is the slot duration. When multiple types of time slots form a particular sequence are being repeated periodically, the sequence is called a super-frame. A super-frame always starts with a beacon slot for beacon signal transmission. It will include management slots and data slots for management signal exchange and data communication[19].

Inside LLDN, there is one PAN coordinator response for management and most of the communications. All other members are end devices with limited features. This means that the LLDN has three states: Discovery, Configuration, and Operation.

Network applying LLDN always starts in the discovery state. During this state, the super-frame only contains a beacon slot followed by one downlink management slot and one uplink management slot. When any end devices start function, they scan possible channels for beacon signals. The end device performs synchronization and identification of the current state using a detected beacon signal. During uplink management slot, end devices willing to join the network perform CSMA/CA to send a discovery response frame (DRF) to the PAN coordinator. PAN coordinator response with an acknowledgment frame and the end device consider it has been discovered as the handshake process is completed [20].

The discovery state can end by either the PAN coordinator or no end devices that have been discovered within the limit of the discovery timeout period. The next stage is the configuration state. In this state, the super-frame structure is the same as the discovery state: beacon slot followed by one downlink management slot and one uplink management slot. End devices discovered in the discovery state perform CSMA/CA to send configuration status frame (CSF), which contains information including MAC address, required timeslot, and uplink or downlink data needed. After receiving CSF, the PAN coordinator response to a configuration request frame (CRF) with all the parameters needs altering. This state ends when the acknowledgment of CRF sent by the end device is received.

After receiving every acknowledgment frame for CRF from discovered end devices, network switches to the third state: operation. Under default situations, management slots are set to false, which means the super-frame now contains one beacon slot and multiple data slots. First, several data slots are reserved only for retransmission of failed packets during the last super-frame. For each data slot, there should be an owner end device assigned during the configuration state. Owner devices have the smallest contention window size in their slots. Group acknowledges (GACK) method is used in LLDN, which means the PAN coordinator is able to give acknowledge to multiple end devices in one slot[21] [22].

## 2.4.2: Related works

The MAC layer of the proposed system needs to support a real-time positioning for a network that contains multiple mobile tags. The tags also sense environmental parameters and are able to send alert messages.

Research done related to this project includes scheduling algorithms, time slot allocation algorithms, fast joining methods and, ranking schemes. Raza *et al.* [23] purposed a MAC protocol for time-critical and emergency communication. The proposed protocol, EE-MAC, incorporates emergency communication and allows immediate channel access for such data. Emergency Enabled MAC (EE-MAC) allows the nodes with critical/emergency information to request channel access with priority. Since the occurrence of emergency communication is asynchronous and is relatively rare, a hybrid scheme is introduced where the regular communication continues in a TDMA based super-frame. In case of emergency, a slotted request mechanism using a control channel is introduced, which allows the coordinator to halt the regular TDMA based transmission and initiate emergency communication by inserting an appropriate number of time slots in the TDMA frame. In case of multiple emergency, requests are triggered at a particular time, a queuing function is used to allocate the resources sequentially. For such cases, the communication of regular TDMA would be stopped for multiple timeslots. To ensure the collision-free transition, a halt and reinitiate sequence is defined, which informs the nodes to stop and resume communication when needed. Halt and reinitiate sequences are initiated by the coordinator to stop regular time frame communications and resume these communications, respectively. A minimum halt duration is also included in a halt sequence to improve the energy efficiency of the network. For evaluation purposes, the performance of EE-MAC is compared to IEEE 802.15.4e LLDN. The results show that the proposed protocol offers up to 92% reduction in channel access delay of emergency communication at the cost of a 5% to 15% increase in delay of noncritical and less time-sensitive data.

IEEE 802.15.4e defines a time-slotted channel hopping (TSCH) mechanism that includes a slot-frame structure. A slot-frame is composed of several time slots. The time slot scheduling is planned based on the routing topology. Each sensor node knows the time when the data transmission/reception will happen from the schedule. Another feature in the IEEE 802.15.4e standard is the frequency hopping, which could improve the reliability of the data transmission [24]. Since the IEEE 802.15.4e TSCH standard does not specify how to plan the time slot scheduling, Wang *et al.* [25] proposed a distributed scheduling strategy such that each sensor node knows the time to transmit packets in a distributed and simple manner.

Patti *et al.* [26] pointed out that LLDN does not provide priority support to properly deal with real-time traffic or dynamic channel configuration capabilities to cope with unreliable channels. Moreover, it offers limited scalability, as the cycle time grows linearly with the number of network nodes. Hence, they proposed the priority-aware multi-channel adaptive (PriMulA) framework, which introduces priority-aware scheduling, multi-channel communication, adaptive channel selection, and channel blacklisting in the LLDN. PriMulA supports a higher number of network nodes than the LLDN protocol while keeping short cycle times. In addition, PriMuLA avoids deadline miss and improves network reliability. It maintains the interoperability with LLDN standard nodes and can be implemented using commercial off-the-shelf devices. In PriMulA, each node maintains a queue of outgoing messages ordered by priority at the MAC level. To efficiently support priority scheduling, PriMulA introduces a novel message, called a PriMulA message, that is embedded in the payload of a standard LL-Data frame. This message consists of the message priority and payload. The PRIO field is encoded in one byte to achieve a good trade-off between the number of priorities and the overhead of each message, so up to 256 different priorities can be handled. However, a larger PRIO field would not affect the effectiveness of the proposed approach. In PriMulA, messages are periodically generated by the application, with a fixed period (P). For each message, the application defines a lifetime, called a relative deadline (D). D is the maximum time interval, measured at the application layer, within which a generated message has to be consumed. Both P and D of a message depend on the supported application. The relative deadline can be shorter than or equal to period, or greater than the period. PriMuLa adopts a fixed priority assignment, which assigns priorities to messages according to their relative deadlines, i.e., the shorter the relative deadline, the higher the priority. PriMulA improves LLDN in several respects. Comparative simulations in realistic scenarios show that the number of nodes that the PriMulA network can support without reaching saturation is increased by 56% compared to the LLDN. The introduction of message priorities not only reduces the cycle time compared to the LLDN and to the MC-LLDN, but it also provides a lower deadline miss ratio. The paper claim implementation feasibility without any hardware modification.

Bitencort *et al.* [27] pointed out that one of the limitations of LLDN regards the support of messages with different sizes and different periodicities. In their work, a slot allocation scheme called AdapTA is proposed, enabling support to heterogeneous message streams. The rationale is to compute a suitable size timeslot to communication devices, enabling adaptive control of the super-frame without changing the LLDN standard. It allocates slots based on the message lengths for the IEEE 802.15.4e LLDN networks. This proposal is motivated by the inefficiency of LLDN to support clusters with heterogeneous message streams. AdapTA overcomes this drawback by assigning different slot sizes to the sensor nodes based on the mini-slot concept. The performance evaluation showed the efficiency of the proposed AdapTA allocation scheme was better compared to LLDN. AdapTA allows the minimization of the network latency, by means of a shorter service interval. This result is achieved as the time reserved for each device is proportional to the message length. Moreover, the AdapTA allocation scheme can be implemented upon commercial devices since it is fully compatible with the LLDN standard [28].

IEEE 802.15.4e Time Slotted Channel Hopping (TSCH) standard has gained much attention within the Industrial Internet of Things research community due to its effectiveness in improving reliability and providing ultra-low power consumption for industrial applications, and in which its communication is orchestrated by a schedule. Despite its relevance, the standard leaves out of its scope in defining how the schedule is built, updated, and maintained. This open issue is one of the trending topics in the IETF 6TiSCH WG, that still needs to be addressed.

Ojo *et al.*[29] focuses on scheduling in TSCH networks in a centralized manner where the gateway makes time and frequency slot allocation. This paper formulates the scheduling problem as a throughput maximization problem and delays minimization problem. They proposed a graph-theoretical approach to solving the throughput maximization problem in a centralized way. For the throughput maximizing scheduling problem, they presented a graph-theoretic approach based on matching theory to solve for the combinatorial properties of the problem by using the Hungarian algorithm to find the maximum weight matching in O (n3). The combinatorial properties of the scheduling problem are addressed by providing an equivalent maximum weighted bipartite matching (MWBM) problem to reduce the computational complexity and also adopting the Hungarian algorithm in polynomial time. Simulation results were provided to evaluate the performance of the proposed scheme. The proposed algorithm reduces the computation complexity by mapping the original problem to a simplified problem equivalent to a maximum weighted bipartite matching problem without the loss of optimality. Simulation results show that the proposed scheme can achieve a very good throughput close to the optimal throughput. The delay minimization problem is said to be a binary integer problem and can be solved using branch and bound algorithms.

**Node Mobility**

Although the IEEE 802.15.4e standard has presented LLDN mode to fulfill the essential requirements of low latency applications, low latency applications, for example, in the automotive industry, demand the support of sensor node mobility, which in turn affects the network performance. Node mobility triggers several dissociations from the network that will increase latency and degrade node throughput. Al-Nidawi *et al.*[30] investigate the impact of node mobility over the LLDN mode while defining key factors that maximize latency and degrade throughput. Besides, an enhanced version of the LLDN model is presented and evaluated that supports node mobility while maintaining the targeted limits of low latency application requirements. The proposed mobility aware LLDN technique manages to reduce the dissociation overhead by a factor of 75%, while the packet delivery ratio has been enhanced by 30%. They also provided an analytical model that provides a snapshot of the trade-off process between different metrics in the IEEE 802.15.4e LLDN design, which must be considered before network deployment in mobile LL applications.

In the joining phase of TSCH network formation, sensor nodes have to remain awake status for a long time until they can reach synchronization. This process consumes a significant amount of energy. Duy *et al.* [31] propose a reliable, lightweight joining scheme for TSCH network formation to speed up joining operation. The scheme is present in detail through analysis models as well as implementation results in real sensor nodes.

As a single-hop network with only one network coordinator, there is no method or need to reduce the network design of LLDN further. Moreover, because the coordinator and other nodes are occupied most of the time, further improvement design of LLDN all involves adding relay nodes [32].

Relay nodes could help increase the reliability of the LLDN network by overhearing all the data of end nodes. In the design, using a relay node between the coordinator and other nodes is introduced and explored. This design could significantly increase the PDR. However, the relay node is not cheap because it will be required to overhear all the data from devices. It will also consume a considerable amount of energy in addition to the LLDN energy costs, which makes this not appear to be very practical.

Relay nodes could also expand the network coverage by acting as an intermediate node between coordinators and nodes out of the maximum coverage. This design share the same drawback, including extra energy consumption and the cost of the relay node itself.

Node mobility support of LLDN can take advantage of the bidirectional slots inside each super-frame. Therefore, a Mobility-Aware LLDN was introduced where bidirectional slots are used. The proxy coordinator could use the concept of the passive beacon during the bidirectional slot, where the new joining node will send their joining request [21].

The main idea of our scheduling algorithm is to arrange a time slot for each leaf node to transmit a packet to the root node in a distributed manner. For an internal node, it may need more than a one-time slot if it has more than one child node. Each node has two tasks. First, it will calculate which time slots it will transmit a packet to its parent node and which time slots it may receive packets from its child nodes. Second, for each assigned time slot, it will determine the channel offset. The simulation results demonstrate that the proposed approach is able to achieve high throughput, low latency, low control overhead, and low power consumption.

## 2.5 Chapter summary

This chapter starts with a brief description of the architecture of the proposed network which serves an indoor smart inventory system with wireless positioning feature. The network consists of multiple devices based on Decawave 1001 module and a cloud server to assist computation. Then two related positioning methods, the two-way ranging method and the proposed original indirect positioning method, are explained. These are followed by a detailed discussion of existing MAC layer protocols based on LLDN in IEEE 802.15.4e.

# Chapter 3 Analysis with queuing model

In the last chapter, the network architecture was described, and a detailed literature survey is provided. In this chapter, an $M/M/1$ queuing system is described. This queuing model is essential to analyze the purposed problem theoretically. This chapter gives the theoretical basis needed for MAC protocol design in the forthcoming chapters.

## 3.1 Resources Requirements Analysis

In the proposed system, there are three types of messages transmitted between readers and tags that require channel resources:

(1) Critical Message (CM): Critical messages are generated under emergency conditions and have a small delay tolerance. Few examples are: the position measurements of a tag exceed a pre-defined threshold; the temperature in the warehouse sensor exceeds 80 degrees; or an item is displaced beyond a specific range (e.g., the item is removed from the shelf, and the item leaves the electronic fence).

(2) Position Message (PM): This is the data related to the position of a target device such as, tags people, forklifts, or other equipment that require real-time position information to be known.

(3) Sensor Message (SM): This refers to the data sent by the environmental monitoring sensor installed in the network area. Contrast to critical data, this type of data transmission have relatively larger delay tolerance.

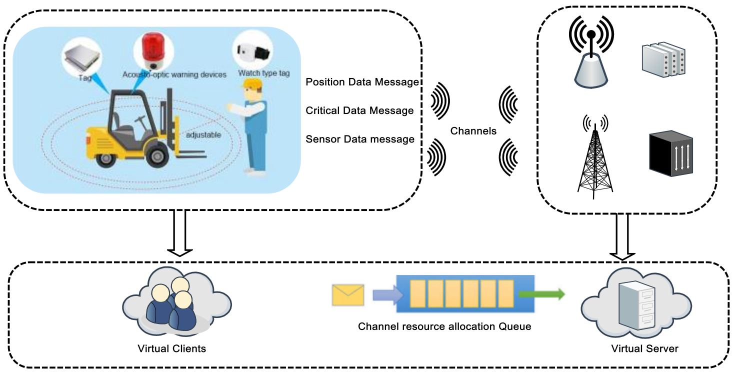



Figure 3.1 Data transmissions in the network and the abstract view

The goal is to design a MAC protocol to support the three types of message transmissions efficiently. In an abstract viewpoint, the channel resource allocation can be considered as a kind of service provided by a virtual server, while the data as three types of clients, the resource of the virtual server is allocated for each type of message transmission. Thus a queuing model can be built for analysis.

## 3.2 Modelling on the System

The model is aimed to provide an initial analysis of the network. Three types of data are assumed to handle by three independent servers: a queuing system model with three roles: virtual clients (or virtual customers), a virtual server and a queuing policy.

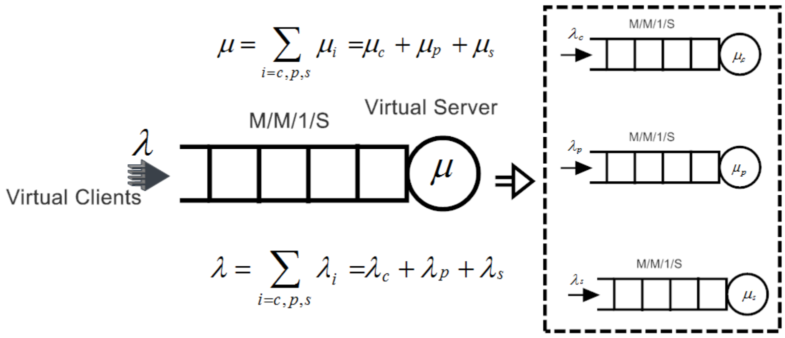



Figure 3.2 The main queuing model

**(1) <u>Virtual Clients and Arrival Rates</u>**

The arrivals stream of virtual clients is formed by three types of messages which are mentioned before: critical data messages, position data messages, and environment or device sensor messages. Arrivals of messages follow a Poisson process $p(x, \lambda_i t) = \frac{e^{-\lambda_i t}(\lambda_i t)^x}{x!}, x \in \mathbb{R}, i \in [c, s, p]$ where $\lambda$ is the average number of outcomes per unit time or region, or virtual clients arrive rate and $x$ are the actual number of clients arrived. Since there are three types of data, let us define three arrival rates:

$\lambda_c$ : Critical data message arrival rate
$\lambda_p$ : Position data message arrival rate
$\lambda_s$ : Sensor data message arrival rate

Although in a real-life situation, there might be end devices with more than one type of data, (for example, a moving temperature sensor). However, in order to design for the worst-case scenario in which the number of clients is maximum, it is assumed that one device can only generate a single type of data. Therefore, from the assumption and the characteristics of the Poisson process, the arrival rate for total virtual clients is: $\lambda = \sum_{i=c,p,s} \lambda_i = \lambda_c + \lambda_s + \lambda_p$ (3.1).

**(2) <u>Virtual Server</u>**

The service of time slot schedule or channel resources allocation is considered as a virtual server. For each client, the virtual server is able to provide service in a random independent time $T_s$. $T_s$ has an exponential distribution with parameter $\mu$, where $\mu$ is the average number of messages being served per unit time. The Probability Density Function (PDF) and the Cumulative Density Function (CDF) of the $T_s$ are given as:

$$f_{T_s}(t) = \mu \exp(-\mu t), \quad t \geq 0 \quad (3.2)$$

$$F_{T_s}(t) = 1 - \exp(-\mu t), \quad t \geq 0 (3.3)$$

The service of the virtual server can be divided into three types corresponding to the messages above. So the queuing system in Figure 3.2 can be separated into three sub queuing systems. The service rate of these queuing systems are,

$\mu_c$, average critical data message service rate

$\mu_p$, average position data message service rate

$\mu_s$, average sensor data message service rate

Similar to the data arrival rate, the total service rate can also be written as: $\mu = \sum_{i=c,p,s} \mu_i = \mu_c + \mu_s + \mu_p (3.4)$, because of the properties of each message, each type of data has different QoS (Quality of Service) priorities with service time constraints. These are,

$t_c$: Critical data message service time， has the highest-level priority

$t_p$: Position data message service time， has the second-highest priority

$t_s$: Sensor data message service time, has the lowest-level priority

This means in most situations it is expected that $t_c < t_p < t_s$

**(3) <u>Queuing policy</u>**


Figure 3.3 The queuing sequence of model

Each type of client has to enter the queue and wait for service. The queue space sizes (or buffer size) are defined as:

$S_c$: Critical data message queue space size

$S_p$: Position data message queue space size

$S_s$: Sensor data message queue space size

Assume the queuing policy follows the rule of “First Come, First Serve” (FCFS). If the queue space is full, any more arriving clients will be blocked. This is similar to a real-life situation where if the buffer of readers is full, then any more incoming messages will be dropped. According to the above assumptions, the model is called $M/M/1/S$ model.

## 3.3 Resources Allocation Analysis in the Queuing Model

Suppose in a specific scenario of a smart factory, there are three kinds of end-devices: critical, position and, sensor. Each device sends a message with service priorities according to their data type, in a certain channel or bandwidth, to meet the maximum service delay constraint of that type of message transmission. Service resources must be allocated correspondingly. This can be achieved through several methods, including optimization of the MAC layer, network layer, or application layer protocols. The scientific problem behind the scenario can be described as follows:

Consider the given $M/M/1/S$ queuing model with three types of clients, each sending critical, position, and sensor type data with different priorities. Arrival rates of clients follow the Poisson process with the rate of $\lambda_c$, $\lambda_p$, and $\lambda_s$ respectively. The queue space or buffer size for the queue is $s_c$, $s_p$ and $s_s$ to ensure the Qos their service times are $t_c$, $t_p$ and $t_s$ with drop probability $p_{bc}$, $p_{bp}$ and $p_{bs}$.

The problem is how to allocate the resource $\mu$ to meet these three different QoS requirements with service rate $\mu_c$, $\mu_p$, $\mu_s$ respectively.

Table.3.1 Variables in the queuing model part 1

| Type of MSG | Arrivals rate of MSG | Service Time | Queue Space | Service Rate |
|---|---|---|---|---|
| Critical data | $\lambda_c$ | $t_c$ | $s_c$ | $\mu_c$ |
| Position data | $\lambda_p$ | $t_p$ | $s_p$ | $\mu_p$ |
| Sensor data | $\lambda_s$ | $t_s$ | $s_s$ | $\mu_s$ |

In the $M/M/1/S$ queuing system, other variable symbols are listed in Table 3.2

Table.3.2 Variables of the queuing model part 2

| Symbol | Description | Symbol | |
|---|---|---|---|
| $\rho$ | Utilization factor, $\rho = \lambda/\mu$ | $\mu$ | QoS time of message transmission |
| $T_q$ | Queuing time | $S$ | Queuing space or buffer size |
| $T_s$ | Service time | $P_b$ | Drop or blocking probability of message |
| $T$ | Total time consumption from data generation to the processing complete $T = T_q + T_s$ | | |

From the queuing model, the service time and queuing time can be represented as below:

$$\begin{cases} T_s = \dfrac{1}{\mu-\lambda} - \dfrac{S\rho^{S+1}}{\lambda-\mu\rho^{S+1}} (3.5) \\ T_q = \dfrac{\rho}{\mu-\lambda} - \dfrac{S\rho^{S+1}}{\lambda-\mu\rho^{S+1}} (3.6) \end{cases}$$

From the above equations 3.5 and 3.6, the total time consumption of data and drop/block possibility are:

$$\begin{cases} T = T_q + T_s = \dfrac{1+\rho}{\mu-\lambda} - 2\dfrac{S\rho^{S+1}}{\lambda-\mu\rho^{S+1}} (3.7) \\ P_b = \dfrac{(1-\rho)\rho^S}{1-\rho^{S+1}} (3.8) \end{cases}$$

Theoretically, the relationship between the variables $\mu$, $\lambda$, $P_b$, $T$, and $S$ can be represented in one equation. Unfortunately, however previous attempts to numerically solve these equations either by using MatLab™ software and by using hand calculations have been not feasible. Hence, an approach is put forward a simplified $M/M/1/S$ queue model is required.

## 3.4 Simplification of the Queuing model

In the queue model $M/M/1/S$, let $S \to \infty$, then the queuing model is transformed into $M/M/1$. In the $M/M/1$ queuing model, the following relationships hold:

The average service time is $E\left[T_s\right] = \dfrac{1}{\mu}$

The probability density function (PDF) $T_q$ is:

$$f_{T_q}(\tau) = \frac{\mathrm{d}}{\mathrm{d}\tau}\left[1 - P\left(T_q > \tau\right)\right] = \rho\mu(1-\rho)\exp(-(1-\rho)\mu\tau), \quad \tau \geq 0 (3.9)$$

Average of waiting time:

$$E\left[T_q\right] = \int_0^\infty \tau f_{T_q}(\tau)\mathrm{d}\tau = \int_0^\infty \tau\rho\mu(1-\rho)\exp(-(1-\rho)\mu\tau)\mathrm{d}\tau = \frac{\rho}{\mu(1-\rho)} = \frac{\lambda}{\mu(\mu-\lambda)}, \quad \mu > \lambda (3.10)$$

The QoS of time (Service response time):

$$T = E\left[T_q + T_s\right] = E\left[T_q\right] + \frac{1}{\mu} = \frac{1}{\mu-\lambda} (3.11),$$

An expression $\mu$ can be drawn:

$$\mu = f(\lambda, T) = \lambda + \frac{1}{T} (3.12)$$

To meet the time constraint of messages $t^c, t^p, t^s$, the service rate is,

$$\mu_i = \lambda_i + \frac{1}{t_i}, i = c, p, s \quad (3.13)$$

Figure 3.4 shows the relationship between $\mu$ $\lambda$ , $t$ . It is clear that when QoS constraint time is very short, the system requires a much higher service rate. QoS in the figures stands for unit period of time.

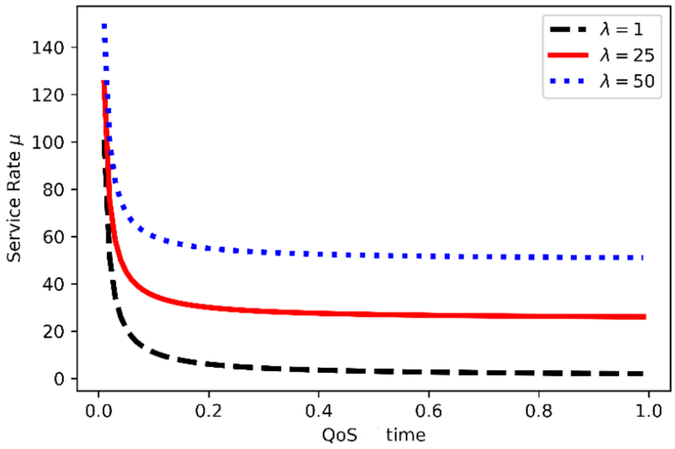



Figure 3.4 QoS time vs. Service rate plot for different arrival rates

It can be concluded that as the value of $t^c, t^p, t^s$ increases, the system's QoS requirements for service response rate decrease, which means it requires fewer service resources.

From the system view (the combination of three queues), the total service rate requirements, which means the total channel or resource captivity is:

$$\mu_c + \mu_p + \mu_s = \lambda_c + \lambda_p + \lambda_s + \left( \frac{1}{t_c} + \frac{1}{t_p} + \frac{1}{t_s} \right) \quad (3.14)$$

$$\mu_c + \mu_p + \mu_s = \lambda_c + \lambda_p + \lambda_s + \left( \frac{t_p \cdot t_s + t_c \cdot t_s + t_c \cdot t_p}{t_c \cdot t_p \cdot t_s} \right) \quad (3.15)$$

$$\sum_{i=c,p,s} \mu_i = \sum_{i=c,p,s} \lambda_i + \left( \frac{t_p \cdot t_s + t_c \cdot t_s + t_c \cdot t_p}{\prod_{i=c,p,s} t_i} \right) \quad (3.16)$$

$$\mu = \lambda + \left( \frac{t_p \cdot t_s + t_c \cdot t_s + t_c \cdot t_p}{\prod_{i=c,p,s} t_i} \right) \quad (3.17)$$

Here, $\mu$ is the total service rate of the system. It can be the total channel bandwidth or capacity.

In $M/M/1$ the queue, let $N_q$ denotes the average message in the queue:

$$N_q = \frac{\lambda^2}{\mu(\mu-\lambda)} (3.18)$$

Then, it can be shown $\mu = \lambda \cdot \left( \frac{1 \pm \sqrt{1+\frac{4}{N_q}}}{2} \right) (3.19)$,

This means that if $\lambda$ and queue space is given, the corresponding service rate can be determined.

However, using $M/M/1$ a substitute $M/M/1/S$ by letting $S \to \infty$ is impractical. For in a real queueing system, the queueing buffer $S$ cannot be infinity. Given a certain $S$, sufficient $\mu$ has to be provided to let the queueing size is small then $S$, $N_q \le S$.

In this case, let $N_q = S$, $\mu = f(\lambda, S) = \lambda \cdot \left( \frac{1 + \sqrt{1+\frac{4}{S}}}{2} \right) (3.20)$, then the relationship between $\mu$ $\lambda, S$ and can be shown in the following figure.

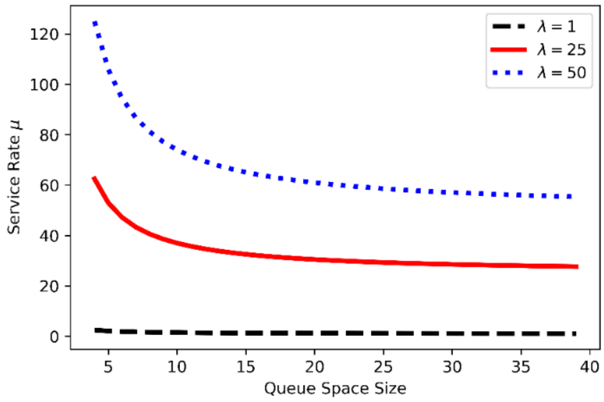



Figure 3.5 Queue space size vs service rate plot for different arrival rates

Figure 3.5 indicates that when the $\lambda$ is low, an increase in the queue space size has little influence on the service rate. As the queue space size increases, the service rate drops. When we combine the above equation, the service rate is $\mu = \max\{ f(\lambda, T), f(\lambda, S)\}$,

Where, 
$$\begin{cases} \mu = f(\lambda, T) = \lambda + \frac{1}{T} (3.21) \\ \mu = f(\lambda, S) = \lambda \cdot \left( \frac{1 + \sqrt{1+\frac{4}{S}}}{2} \right) (3.22) \end{cases}$$

This means at the MAC layer, according to the equation above, with proper allocation strategy of the time slot, the requirement of different QoS time constraints can be met. Besides, in this model, the $\mu_c, \mu_p, \mu_s$ are fixed. If the probability of a critical data message sent on the system is relatively small, the resources allocated for $\mu^c$ may be wasted most of the time, causing unnecessary delay in other messages. Hence, to improve the overall performance, unused $\mu^c$ should be reallocated for other types of data messaging transmission. Also, if critical transmission requires more resources than what is currently available, then resources originally allocated for other clients with lesser importance could be reallocated to it to improve the overall quality of service.

When the total resources are constant, which means: $\mu = \mu_c, \mu_p, \mu_s$ is constant, then by dynamically adjusting any of the $\mu^c \ or \ \mu^s \ or \ \mu^p, the$ highest efficiency can be achieved.

## 3.5 Chapter summary

This chapter presents an analytical model for resource allocation for the MAC protocol design. An M/M/1 queuing model is built with a queue length constraint mechanism to simplify the M/M/1/S queue. The relationship between service rate, acceptable QoS time delay, queuing length are buffer size are analyzed. The result suggests that by appropriately optimizing the resource allocation for each client, overall efficiency can be improved to an absolute maximum. This model can be used as a reference for further designing resources allocation in the MAC protocol.

# Chapter 4 Design of MAC layer

The chapter explains the details of the MAC layer design of the indoor positioning system. Section 4.1 gives an overview of basic components, which are devices, messages, and function cycles. Section 4.2 analyzes the basic parameters and performs an initial-boundary value calculation. Section 4.3 provides two directions to improve the communication quality of the network. After combining the improve methods, workflows of network devices are introduced in section 4.4.

## 4.1 Overview of the system network

In this section, all of the basic components of the network are introduced. Besides the server providing extra calculation assistance, most hardware in the system are devices based on Decawave 1001 modules, which have Decawave 1000 IC built inside. Messages being transmitted between the devices are based on the IEEE 802.15.4 protocol. Because only one channel is available in the project design, two function cycles are designed: the discovery cycle(DC) and the positioning cycle(PC). The last part explains a joining period in the positioning cycle designed for critical tags failed to transmit CM to retry communication.

The system is expected to be inside an environment similar to a warehouse. The environment is divided into positioning zones of approximately $50 \times 50$ meters. The MAC layer focused on the devices within each positioning zone.

As discussed in sections 1 and 2, this design is highly specified and closely related to the WPSN project in collaboration with PeyTec. and supported by NSERC CRD grant. Values to design parameters were gathered from real life scenarios through discussion with engineers in the industry specifically Peytec. Details are discussed in the specified sections.

### 4.1.1 Network devices

As shown in Figure 4.1, the system has three types of devices: server/gateway, reader, and tags.

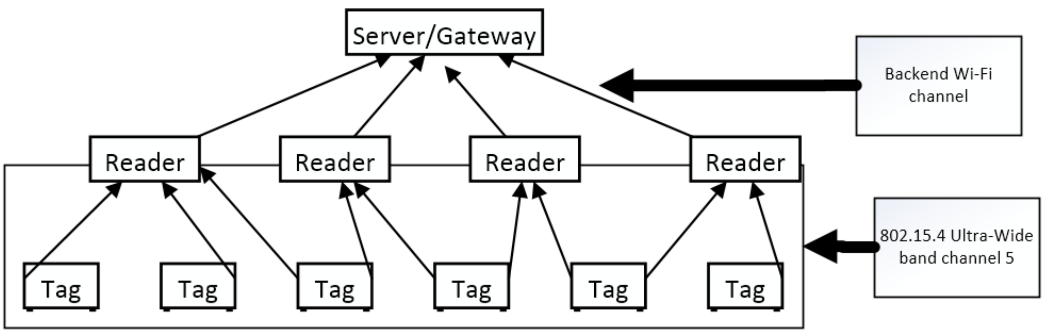


Figure 4.1 General network diagram

**The server** performs the control function. It can be either located in a cloud or in the local network serving as a gateway to the cloud. Through a Wi-Fi channel, it is connected with every reader or at least the master reader wirelessly.

The server has the following properties:

- Able to perform position calculation to improve system efficiency
- Because connected wirelessly, the communication between server and reader is not reliable

**Readers** are connected with the server through a Wi-Fi channel and connected with tags through an ultra-wideband channel. Readers are deployed in the network after planning to known locations that never change. The number of readers inside the network is assumed to be enough to cover the whole network area. From the positioning method explained in chapter 2, the minimum number of readers is 3. The maximum number of readers are expected to be 10 for a typical industrial plant.

Readers are based on Decawave 1001 devices and have the following properties:

- Able to communicate with the server through a Wi-Fi channel, which allows the server to aid calculation.
- Able to communicate with tags through an ultra-wideband channel: beacon messages, acknowledgement messages, and other messages. The critical data, positioning data, and sensing data are all exchanged through this channel.
- Perform positioning calculation or store the data required for calculation if aid from the server is not available

Master reader:

In each positioning zone, there is one **master reader**, which generates a beacon message and sends ACK to tags. The master reader is selected before the network starts its operation. For the reliability of the beacons, a node in the center of the zone is selected as the master reader such that most of the other readers receive its beacon at high signal strength.

Though the master reader is important, all readers are equally capable of performing the functions of a master reader; hence, one of those readers replace the master reader in case of its failure. Similar to Decawave 1001 default positioning method, the master reader has a time limit to provide service. One or multiple assist readers are selected to replace the master reader if the current master reader fails.

Non-master reader:

All readers inside the network zone except the master reader take parts in PP and overhears every uplink transmissions to increase the success rate.

**Tags** are sensor nodes that collect information from the environment. They are the source of data and expected to be localized, either mobile or fixed. There are three types of Tags:

- **Critical** tags send critical data, and they can be mobile.
- **Sensor** tags send sensor data, and with some exceptions, they are generally not expected to be mobile.
- **Positioning** tags are mobile and their positions are tracked.

### 4.1.2 Messages in the network

According to Decawave 1000 device user manual, network messages are transmitted through a 6.8 Mbps ultra-wideband channel. Besides the formatted overhead of each message, the payload of each message is 20 to 25 bytes. Even with the maximum payload which is 127 bytes, it takes less than 0.5 milliseconds for transmission through the 6.8Mbps channel.

The length of data generated by embedded sensors is normally several bytes. For the 6.8 Mbps ultra-wideband channel used for network communication, this means critical, or sensor frame transmission delay is less than 0.3 millisecond. Thus critical and sensor data can be included in any message from tag to the reader.

The MAC includes two types of cycles: the discovery cycle and the positioning cycle. A detailed description of the network cycles is in the next section.

Beacon message (BM):

Beacon messages are broadcast by the master reader in the network in the beginning slot of each cycle. This type of message contains:

- Indication of the current cycle to be either discovery cycle or positioning cycle
- Slot duration usually in milliseconds.
- Number of slots in the current cycle can be used to calculate the time to the next beacon message

Discovery response message (DRM):

Discovery response messages are sent from tags to all readers after receiving a beacon message in a discovery cycle. Purposes of this type of messages are:

- acknowledgement of successful reception of the beacon
- Attempt to register itself as a member of the network

Acknowledgement message (ACK):

Acknowledgement messages are sent from the Master reader to a tag in response to various messages from the tag. Purposes of this type of messages are:

- Acknowledgment of the successful reception of a message.
- Real time left for this cycle (for those tags missed beacon message in the current cycle and tags who already been assigned a slot, this value helps them to set sleeping time).

For the discovery response message, it also contains:

Slot number assigned for the tag being discovered in the next positioning cycle.

Positioning message (PM):

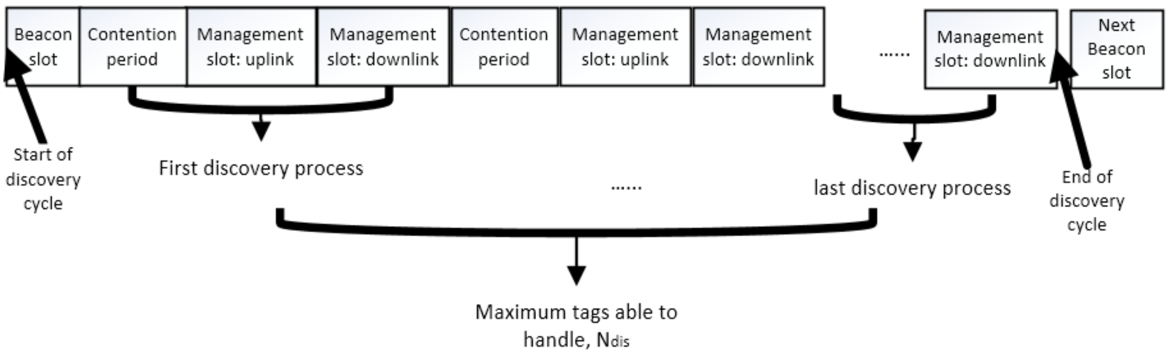
A tag broadcasts Positioning Message (PM) in its assigned positioning ess. After the broadcast of the first PM, the tag waits for ACK from the master reader. If it receives acknowledgement successfully, then it proceeds to broadcast the second DP to complete the PP. Otherwise, it can repeat the transmission of the first DP until the maximum wait time expires.

### 4.1.3 Network cycles: discovery and positioning

The proposed MAC is a scheduled MAC where its basic scheduling cycle comprises of two periods: discovery period (DC) and positioning period (PC). Each period starts with a beacon message broadcast by the master reader. The master reader requires an ACK to assign a positioning process for each joining tag, and the delay between initial tag joining and possible positioning process are required to be minimum possible, thus during functioning the network always toggles between DC and PC. Initial cycle is the discovery cycle.

**DC** is meant for tags to join the network. DC starts with one beacon slot, followed by multiple discovery processes (DPs) consists of one contention period and a pair of management slots, including one uplink slot and one downlink slot.

Many difficulties exist in the realization of the CSMA/CA mechanism due to the limitation of the available software for implantation. Currently, the network requires a minimum 1 millisecond to sense any massage or data transmission. Thus the contention period is designed to be 2 milliseconds, initial back-off for critical messages is 1 millisecond, and 2 milliseconds for non-critical




messages.

Figure 4.2 Structure of the discovery cycle

Critical tags have higher priority over non-critical tags. This cycle only handles random data transmission. Random data transmission is not scheduled and can be initiated at any time during the functioning of the network. Random data includes joining attempt of tag, critical data, and sensor data. Although sensor data is generally periodic and can be scheduled, there are cases for unexpected sensor data transmission that requires aperiodic transmission.

The Master reader assigns positioning slots in PC for each tag successfully completed join process during DC. It transmits the positioning slot number in its acknowledgment to the designated tag. The tag later transmits positioning data in the assigned in the next PC. Because of the random nature of tag joining, collisions are expected to happen in uplink slots. Collision may increase delay, decrease data throughout, and increase energy consumption. The main design objective is to reduce collision in DC

To avoid collisions, after receiving BM from the master reader, each tag randomly chooses one available uplink slot for the initial transmission attempt. If the initial transmission attempt fails, critical tag checks if there are any uplink slots left available and retry if possible; Non-critical tags skip the current DC and wait for the next BM.

The value of critical messages drops significantly as delay increases, so critical messages are prioritized by the contention mechanism. Before the uplink transmission attempt, a tag waits for a certain time period called initial back-off. Critical tags back-off (wait) for lesser time than non-critical tags. With the carrier sensing feature, non-critical tags will sense any critical data occupying the channel and skip the current process.

**The positioning cycle** is for finding real-life locations of tags in the network. It starts with a beacon slot, followed by multiple positioning processes (PPs). A PP consist of three transmission slots: first is for uplink transmission from the tag to the reader, second for downlink transmission from the reader to the tag, and the third is for uplink. As discussed in the previous section, three successful transmissions are enough to find the location of a single tag. Target tags for each PP are determined in DC prior to the current PC. The tags do not transmit random data during PC, and PPs are expected to be in series, which creates a time period preventing random communications.

In PC, **joining period**s are optional. The joining period is a time period where no scheduled communication is happening. Tags failed to join in previous DC are able to retry to join the network during a joining period in a PC. The joining period allows critical tags to fail in previous DC to retry communication with the master reader. Any critical tags are allowed to keep attempting to join the network before receiving an ACK, which indicates transmission success.

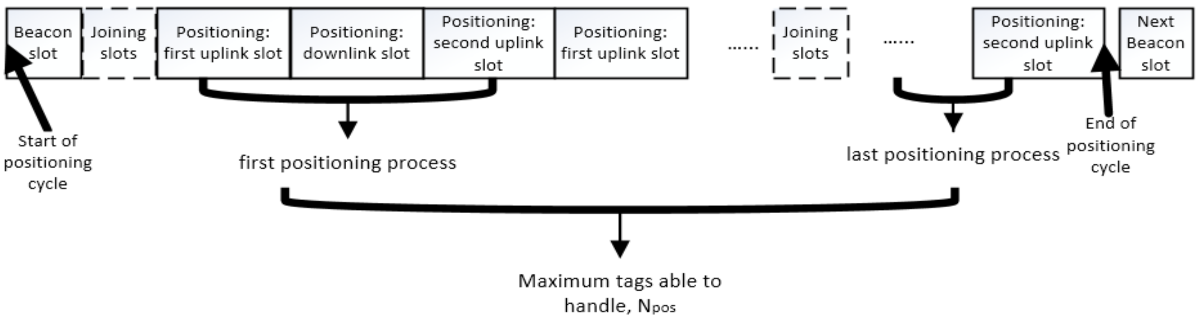



Figure 4.3 Structure of positioning cycle

Possible location to insert joining period in positioning cycles includes: between the initial beacon slot and positioning processes; between positioning processes; and during the end of the cycle. Inserting a joining period before positioning processes is expected to raise the success rate of critical message transmission; inserting joining periods between positioning processes is mean to avoid any possible starvation; joining period at the end of the cycle is used for data transmissions with high delay tolerance, in project case, it is the sensor message.

## 4.2 Analysis of boundary values and parameters

This chapter aims to analyze parameters and boundary values. Design is based on the parameters given or calculated information from customer, and guided by boundary values.

### 4.2.1 Slot time calculation

Slot time calculation are based on the parameters in Table 4.1

Table 4.1 Parameters in slot time calculation

| Variable name | Description | Symbol | Typical value |
|---|---|---|---|
| Processing time | Processing time of each message in hardware | $t_{proc}$ | Depend on different hardware using in the system |
| Message transmission/reception time | Time required for each data transmission | $T_{tx}$ | Depend on data packet length |
| Guard time | Time period in each slot to reduce the impact of unexpected time error | $t_{guard}$ | Depend on the maximum possible timer error |
| Packet size | The number of bytes in a data packet payload | $P_{size}$ | Depends on the requirements from customers |
| Data rate | Data transmission bitrate over the radio channel | $R_{data}$ | 6.8 Mbps for 802.15.4 Ultra-wideband channel 5 |
| Slot time | Time duration of a slot | $t_{slot}$ | Designed value base on transmission time, maximum timer error, and processing time |
| Discovery cycle time duration | The real-life time duration of one discovery cycle | $t_{dis}$ | Calculate by other parameters |
| Positioning cycle time duration | The real-life duration of one positioning cycle | $t_{pos}$ | Calculate by other parameters |

For the tags and readers based on Decawave 1001 devices, the data rate is 6.8 Mbps. On average data packet size is 20 bytes; the maximum size possible is 127 bytes. Thus the maximum transmission time possible for each packet is

$$T_{tx_{max}} = \frac{1}{R_{data}} \times P_{size_{max}} = \frac{1}{6.8 \times 10^6} \times 127 \times 8 \approx 0.15\, millisecond$$

The average transmission time is

$$T_{tx_{avg}} = \frac{1}{R_{data}} \times P_{size_{avg}} = \frac{1}{6.8 \times 10^6} \times 20 \times 8 \approx 0.02\, millisecond$$

Internal timer error of the system is tested to be lesser than 1 millisecond; thus the guard time is decided to be 1 millisecond

$$t_{guard} = 1\, millisecond$$

Processing time of one message, which can either be generating the message or receive and decode the message. In each slot, it is expected that both generation and reception of message happens once. To make sure each processing successful average time for processing is

$$t_{proc} \approx 1\ millisecond$$

Data transmission includes generation of data in the initial device, transmission of a message over a radio channel, reception and process message. To reduce error it should also be two guard times at the beginning and end of a slot, thus for one message transmission slot time should be:

$$\mathrm{t_{slot}} = 2 \times \mathrm{t}_{proc} + \mathrm{T_{tx}}_{max} + 2 \times \mathrm{t}_{guard} = 2 \times 1 + 0.15 + 2 \times 1 = 4.15\ milliseconds$$

Slot time resolution is 1 millisecond, so $t_{slot} = 5\ milliseconds$ This means one data transmission. The slot duration is 5 milliseconds.

The discovery of the tag includes one uplink transmission of the discovery response message and one downlink transmission of the acknowledgement message. Because the minimum guard time is 1 millisecond. The minimum initial back-off is 2 milliseconds: critical tag back-off for 1 millisecond; non-critical tag back-off for 2 milliseconds.

The minimum tag discovery time is

$$t_{dis} = 2\ transmissions + initial\ backoff = 2 \times t_{slot} + 2 \times t_{guard}$$
$$= 2 \times 5 + 2 \times 1 = 12\ milliseconds$$

As introduced in the previous section, a PP includes 2 uplink transmissions and one downlink transmission, which is used to measure the location of one tag.

The minimum tag localization time is

$$t_{pos} = 3\ transmissions = 3 \times t_{slot} = 3 \times 5 = 15\ milliseconds$$

### 4.2.2 Initial boundary value calculation:

Following constraints are acquired from customers:

Table 4.2 Max delay tolerance for message types

| Type of message | Max delay tolerance (milliseconds) |
|---|---|
| Critical | 500 (0.5 sec) |
| Sensor | 10000 (10 sec) |
| Positioning | 1000 (1 sec) |

List of variables are in in the table below:

Table 4.3 Variables for initial boundary value calculation

| Name | Symbol | Description |
|---|---|---|
| Max critical tag | $maxtag_c$ | Max number of critical tags each discovery cycle able to handle avoiding starvation |
| Max sensor tag | $maxtag_s$ | Max number of critical tags each discovery cycle able to handle avoiding starvation |
| Max positioning tag | $maxtag_p$ | Max number of critical tags each discovery cycle able to handle avoiding starvation, due to the characteristics of positioning message, the calculation is more complex |

An initial upper bound on the number of tags can be computed using the following formula:

$$\frac{\text{maximum critical delay } - \text{beacon slot duration}}{slots\ needed\ \times slot\ duration} = maximum\ \ possible\ number\ of\ tags$$

As discussed in section 4.1, DPs are able to handle random data transmission. Critical data and sensor data can be transmitted along with the request of joining the network; this means that as long as the joining is successful, data is successfully transmitted. If unlimited slots are provided for one DC and every critical tag and sensor tag keeps attempting to send DRM before receiving an ACK, it is still possible for them to exceed maximum delay tolerance. Positioning data can only be considered successfully transmitted when the time period between DP and the PP arranged by the master reader is within the maximum delay tolerance.

To avoid expire maximum delay tolerance of three kinds of tags, the available processes (AVAP)inside each DC consider only one BM is broadcasted are:

$$maxtag_s = \frac{\text{delay tolerance}-\text{beacon length}}{(backoff+one\ pair\ of\ slots)} = \frac{\text{10000 milliseconds} - \text{5 milliseconds}}{(2+5\times 2)milliseconds} = 832\ tags$$

$$maxtag_c = \frac{\text{delay tolerance}-\text{beacon length}}{(backoff+one\ pair\ of\ slots)} = \frac{\text{500 milliseconds} - \text{5 milliseconds}}{(2+5\times 2)milliseconds} = 41\ tags$$

Positioning tag: the joining order decides the positioning order to reduce the delay between the initial joining attempt and the PP in the next cycle.

$$maxtag_p = \frac{\text{delay tolerance} - \text{discovery beacon length } - \text{positioning beacon length}}{(backoff + one\ pair\ of\ slots)}$$

$$= \frac{\text{1000 milliseconds } - \text{5 milliseconds } - \text{5 milliseconds}}{(2 + 5 \times 2)milliseconds} = 82\ \ tags$$

This means each DC can arrange positioning processes up to 82 positioning tags before starvation happens.

For supporting 41 joining attempts, the time duration of one DC after the BM is $41 \text{discovery}\ process = 41 \times 12 = 492\ milliseconds$. Time left before expiration of critical delay tolerance is $500 - 497 = 8\ milliseconds$. The result suggests that between each beacon that allows random data transmission, to ensure that no message expires the minimum delay tolerance, one DC is able to support up to 41 joining attempts of all kinds of tags.

PC cannot handle any random data communication, this means that the duration of PC cannot exceed the time for delay tolerance. Considering no other transmission is allowed:

$$maxtag_p = \frac{\min(\text{delay tolerance})-\text{discovery beacon length}}{(three\ slots=one\ positioning\ process)} = \frac{\text{500 milliseconds} - \text{5 milliseconds}}{(5\times 3)milliseconds} =$$

$33\ tags$.

However, the total time duration of a PC with 33 PPs is $33\ positioning\ process + 1\ beacon\ slot\ length = 33 \times 15 + 5 = 500\ milliseconds$. This leaves no extra time for any other transmission. If the DC does not have any time period for random communication, up to 32 PPs are supported.

$$maxtag_p = 32\ tags, \text{if no time period inside positioning cycle provided}$$

The initial calculations can be used as the design guide for a relatively ideal situation. The aim is to make sure that no message exceeds maximum delay tolerance because of waiting. There exists a trade-off between avoiding long waiting time and successful transmission rate when the number of joining tags is greater than the calculated results. Because this project is heavily related to the application and is limited by real-life constraints, the estimated number of joining tags are given by the customer. A discussion of the joining tags is in the next section.

### 4.2.3 Tag joining and Data arrival

In real life environment, critical or emergency situation do not likely happen at high frequency. Regular communication is expected to be sensor data and positioning data. For design purposes, more extreme cases are considered. The situation is similar to the initial joining period, where all tags are joining the network at the same time. Other possibilities are when an emergency incident happens inside the whole network zone, which leads to all critical tags trying to send an alert message. The estimated average joining of tags are provided by the customer and are described below:

Table 4.4 Number of expected tags per second

| Tag type | Maximum joining tag | Average joining tag | Maximum delay tolerance |
|---|---|---|---|
| Critical tag | 50 | 25 | 500 milliseconds |
| Sensor tag | 100 | 50 | 10000 milliseconds |
| Position tag | 100 | 50 | 1000 milliseconds |

These values are provided by customers at the start of the project design.

Critical tags: For every second, there are expected to be an average of 25, and the maximum number of 50 critical tags attempt sending critical data. Critical tags are able to join the network and send critical data at the same time. Critical tags are allowed to have mobility. This means critical tags can utilize the random communication time for joining the network. This means the data transmission of the critical message could be considered the same as the critical tags are repeatedly joining in a network system without memory.

Sensor tags: There are expected to be an average of 50 and a maximum number of 100 sensor tags inside the positioning zone. The customer requirement shows that sensing data can be generated after every minute. After a discussion with the customer, the maximum delay tolerance for sensing data is set to be 10 seconds. Sensing tags are expected to be fixed inside the environment and do not require localization. Sensing data can be scheduled for periodic transmission.

Positioning tags: Average and maximum number of positioning tags are considered to be 50 and 100, respectively. They are mobile devices travelling inside the whole indoor environment with the walking speed, which is around 1.4 meters per second. This is one reason that the delay tolerance of DP is 1 second. There always exists a possibility for positioning tags to enter or leave the network zone. DP requires the sender tag discovered by the network first and positioned in the following DC.

The target average number of all three types of tags joining the network= $25 + 50 + 50 = 125\ tags$, which is more than three times the calculated value. This shows that if all 125 tags are allowed to join the network to avoid starvation, it will create a considerable amount of collisions that would likely reduce the transmission success rate (TSR). Thus, we need to develop a mechanism for meeting the target delays for each type of tag.

We consider three cases for joining of tag in the simulation: stable, normal, and random.

Table 4.5 Variable list of tag joining type analyze

| Name | Symbol | Description |
|---|---|---|
| Possibility for a certain number of tags joining | $p$ | This value varies with the type of joining |
| Number of joining tags | $x$ | The input value for joining possibility function |
| Estimated average joining tags | $\lambda, \mu$ | Acquire from customer |
| The standard deviation for normal distribution | $\sigma$ | Can be adjusted to meet the design goal |

**Poisson type**: The possibility of tag joining is described with Poisson distributions: $p\left(x, \lambda^{i}\right) = \frac{e^{-\lambda^{i}}\left(\lambda^{i}\right)^{x}}{x!}, x \in 1,2,3 \ldots, i = c, s, p;$ (4.1) where parameter $\boldsymbol{\lambda}$ stands for the estimated average number of joining tags. $\boldsymbol{\lambda}_c, \boldsymbol{\lambda}_s, \boldsymbol{\lambda}_p$ denotes average arrival rate of critical tag, sensor tag, and position tag.

However, for this joining type, the value of standard deviation is too small that the maximum possible number of tag joining is 0, $p\left(max^{i}, \lambda^{i}\right) = 0,\ i = c, s, p, [max^{c}, max^{s}, max^{p}] = [50,100,100]$ (4.2) Thus this joining type is discarded because the possibility for the estimated maximum number of tag joining is 0.

**Random type: The** possibility of tag joining is described by Gaussian distributions:

$$p(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad x \in \mathbb{R} \qquad (4.3) \qquad \mu\ is\ the\ average\ joining\ tag,$$

$\sigma\ is\ chosen\ to\ be \frac{1}{5}\ of\ the\ maximum\ possible\ joining\ tag$ when joining behaviour is described by Gaussian distribution, the possibility for maximum and even beyond is positive. The possibility of tags exceeds the limit is sum to the possibility of the closest maximum or minimum. This type of joining provides are consider to be the best among three choices to describe real-life joining.

**Uniform type: the** possibility of tags joining in are all equal: $p = \frac{1}{max\ tag\ possible}$ (4.4) compare to the other two types this describes tags are joining in a random fashion. This would violate the estimated average value and is also not considered.

## 4.3 Possible improvements

The main purpose of the scheduling design is to increase the data throughput of the network. This can be achieved by avoiding delay expiration for every message. There are two main reasons for the rise in delay: Collision and starvation. If DC supports fewer discovery processes, collisions become the main reason for the delay increment. However, if the DC and PC support more tags joining slots, starvation has a lesser possibility to seriously affect TSR of transmission. The estimated number of tags joining is expected to be the initialization, periodically resetting, or a worst-case for emergency situations.

Collisions normally happen in the uplink slot when two or more tags transmit. The master reader can detect a collision when it senses the channel occupied but cannot decode the frame.

The situation can be improved by either reducing the possibility of collisions or reducing the impact of collisions on delay.

To reduce collision, after receiving BM in DC, each tag chooses a random available DPs to attempt sending a discovery response message. For all joining tags, the maximum available DPs are acquired from BM. Each tag has a unique 16-bit tag ID. However, tags do not have information about each other. This problem can be described as: Given a certain number (number of tags) of unique 16-bit numbers. Without any further information, give a mechanism that could mapping all of the 16-bit numbers to a limit range (number of AVAPS). The mapping result is required to have collision-resistance.

Several methods of random selecting are considered: modulo result, totally random, and hash function result. Variable list is below:

Table 4.6 Variable list for random initial method analyze

| Name | Symbol | Description |
|---|---|---|
| Selected discovery process | $S_{ini}$ | Result of random initial process selection |
| Number of joining tags | $x$ | The input value for joining possibility function |
| Estimated average joining tags | $\lambda, \mu$ | Acquire from customer |
| The standard deviation for normal distribution | $\sigma$ | Can adjust to meet the design goal |
| Tag ID | $ID$ | Unique tag id, assume to be a 16-bit number |
| Available discovery process | $S_{ava}$ | Available discovery processes decided by the master reader. |
| Number of discovery processes | $S_{dis}$ | Total number of discovery processes in the discovery cycle |
| Number of positioning processes | $S_{pos}$ | Total number of positioning processes in positioning cycle |

Modulo result method makes use of the 16-bit unique tag ID: $(ID\ \boldsymbol{mod}\ S_{ava}) + 1 = S_{ini}$, this method maps every tag to one available DP. For an increasing number from 0 to maximum tag ID, which is 65535, this method provides a result of a periodic linear function which have a value range of 1 to available DP.

In the project environment, every tag does not have information about other tags within the same network zone. There is a chance for high collision-resistance if the ID of tags is a series of continues number in the application.

Total random method is randomly choosing one available DP for every tag: $rand(S_{ava}) = S_{ini}$, this method is randomly choosing one initial DP. The random function used here is the build-in Mat lab random function, which generates results using the real-life time when the function is running as parameters and follows distribution uniformly. This means the function does not have any relation with tag ID or any other given parameters.

Hash result method, this method takes advantage of existing hash functions.

Existing hash functions such as MD5 and SHA256 have high collision resistance. Unique tag IDs could be considered as the input, and the result is selected DP. This method is considered not a good choice for the project design case because the length of the hash function result far exceeds the available range of DP. One example is MD5 result, 32-bit hexadecimal value, which ranges from 0 to 5.4445e+39. This means for available DPs, denoted by $\mathrm{S}_{\mathrm{available}}$, the possibility for one tag id lies inside range of AVAPs is $\frac{\mathrm{S}_{\mathrm{ava}}}{5.4445\mathrm{e}+39}$. Even for 1000 AVAPs, this possibility is less than 0.01‰.

To further explore the efficiency of the modulo method and total random method, Matlab simulation are used results are described in chapter 5. For available DPs with number $\mathrm{S}_{\mathrm{available}}$, according to the pigeonhole principle, if entering tag exceeds $\mathrm{S}_{\mathrm{available}}$ there is guaranteed to be more than one collision. Due to the given tag joining estimation exceeds calculated boundary value, collisions cannot be avoided. Thus, reducing the possibility of a collision through a random initial DP choosing method. The other direction considered is to reduce the successful rate impact of collisions.

The structure of DC and PC is introduced in section 4.1.3. In every management slot, only one data transmission is allowed. For every received discovery response message, the master reader response one ACK in the following downlink slot, which can assign one PP. The possibility exists for every tag joining in requiring positioning. Thus, the system is required to prepare the same number of PP as DP, $S_{dis} = S_{pos}$.

The information of available DPs and PPs is received by every tag in DC, to avoid unexpected errors, the number of DP does not change in the following PC. If one collision happens, the master reader cannot assign any tag to one PP. 3 available slots (a message PP takes) is created in the PC. The design allows critical tags to retry, joining the network in the available slots created. This means for more than two critical tags colliding in the same discovery process, which creates a collision, 1.5 discovery processes are created in the positioning cycle for retry joining purposes. Because collisions are expected to happen, the joining period is inserted right after the beacon slot to shorten critical message delay and raise the success rate of critical messages. One result of collisions happening is fewer positioning messages received and less possibility for starvation caused by PPs. Joining period at the end of the cycle is not considered because it is not likely to raise critical message successful transmission rate.

Two directions to deal with collisions are discussed in this chapter. After combines both the random initial process selection and provide retry slots according to the number of collided DPs, discussion of workflow for tags and master reader is in the next section.

## 4.4 MAC Algorithm

This chapter introduces details of the MAC layer. The flowchart of tags and master reader access algorithms are described below.

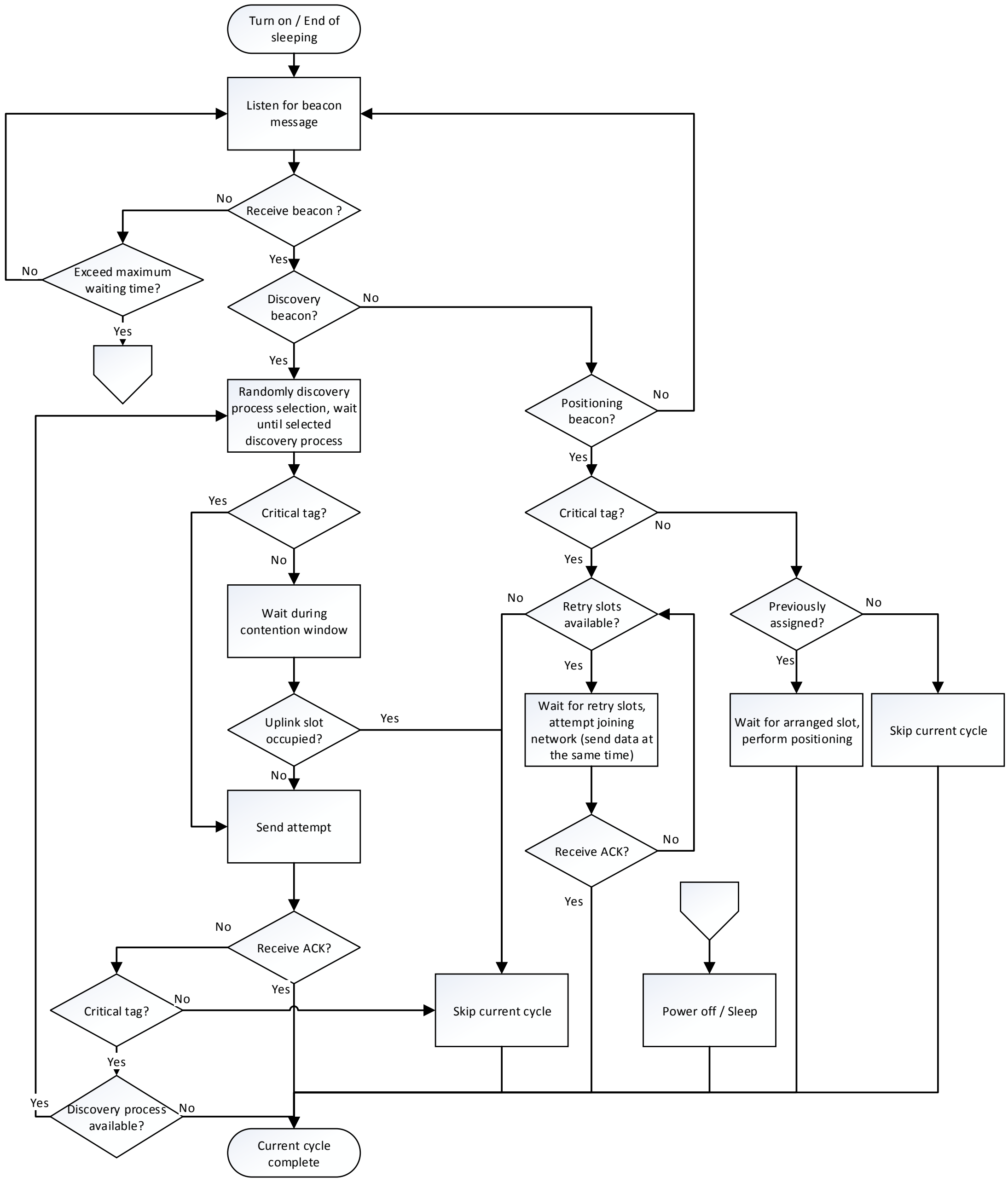



Figure 4.4 Flow chart for tags

When a tag is powered on, or it ends the sleep period, it listens for BM from the master reader. If a tag missed a BM, it keeps listening for BM until the maximum wait time expires. If it fails to receive BM then it tags powers off itself or goes to sleep mode.

If the BM indicates the beginning of DC, the tag randomly selects one initial DP to attempt joining the network. A joining tag can also send critical or sensor data along with joining message. When the selected DP starts, during the 2-millisecond contention window, critical tags send the message after waiting for 1 millisecond; non-critical tags wait for 2 milliseconds and perform carrier sensing before beginning its transmission. If the channel is occupied, the non-critical tag skips the current DC; otherwise, it transmits the message. Tags wait for ACK message after the transmission. If no ACK is received, critical tags perform another selection among the rest DPs and skip the current cycle if there is no DP available left while non-critical tags skip current DC. Upon ACK reception, the tag considers the data transmission successful and goes to sleep mode for the remaining DC time.

If the BM indicates the beginning of a PC, tags will perform localization in their assigned slots that they have received in the ACK from the master reader upon successful join in the previous DC. The non-critical tags that failed to join the network in the previous DC skip a current cycle. The critical tags that miss joining the network in the previous PC check the beacon for any available joining slots in the current PC. Upon finding such a slot, it attempts joining the network; otherwise, it skips the PC. This mechanism allows a critical tag that arrives passed the beacon of the last DC to join the network. It also allows a critical tag that has failed to join the network in the last DC due to collision can still join the network without missing its deadline.

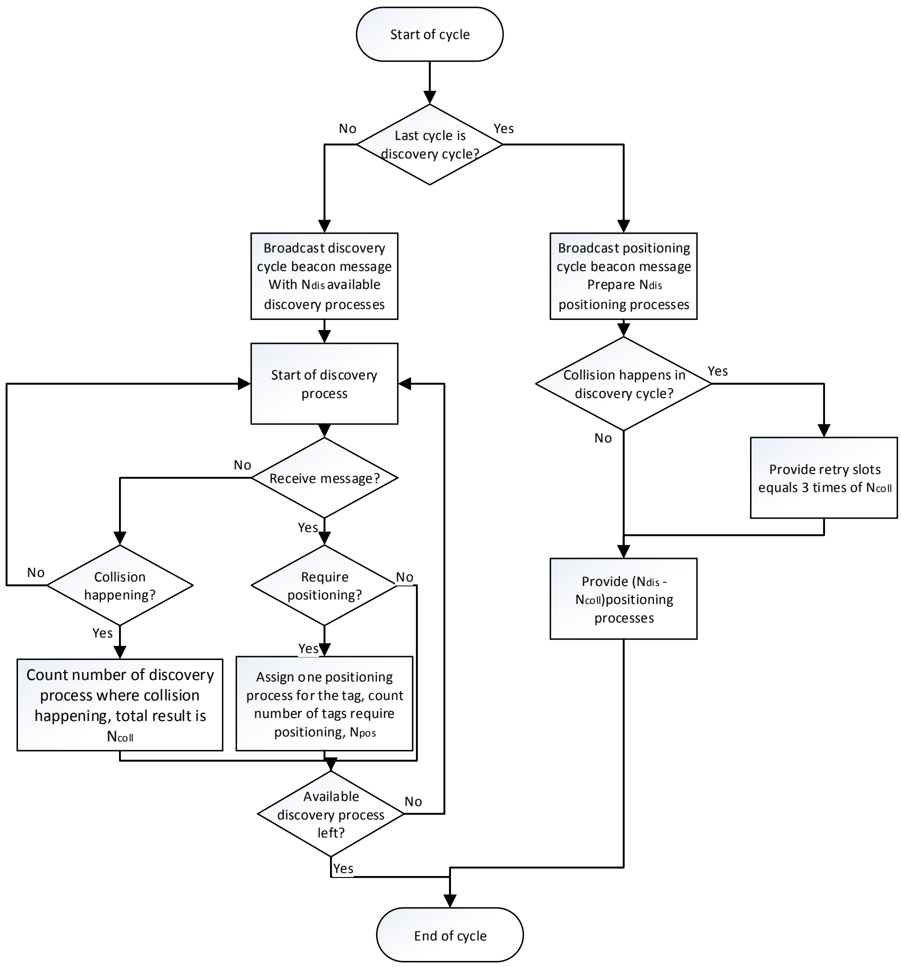



Figure 4.5 flow chart for master reader

Compared to tags, readers have an unlimited power supply. Thus readers are able to stay in active mode. At the start of each cycle, the master reader toggles the current cycle to be different from the last cycle, which could be achieved by checking the previous cycle type and broadcast a different beacon type.

If the current cycle is decided to be a discovery cycle, the master reader broadcasts a BM announcing $S_{dis}$ DPs are available in this cycle. After transmitting a beacon message, the master reader listens for uplink transmission in each DP. It responds successful uplink transmission with its ACK message to the tag. A tag that requires a slot in the positioning cycle is assigned a slot number that is sent in the ACK message. The master reader counts the total number of slots assigned in $N_{pos}$If the master reader detects a collision; it keeps count of the number of DPs with a collision in $N_{coll}$.

To reduce unexpected errors, every PC also have $S_{dis}$ PPs. At least $N_{coll}$ PPs will be empty because the maximum number of position process master reader able to assign is $S_{dis} - N_{coll}$. First, $N_{coll}$ PPs are left for critical tags failed in previous DC to retry and any new critical tags to join. They are followed by $N_{pos}$ PPs already assigned in DC. The rest $S_{dis} - N_{coll} - N_{pos}$ PCs are left for sleeping time. This period can also be used for sensor data transmission or other random message transmissions.

Other readers, called assisting the reader inside the network assists master readers with message reception, ACK message response, and positioning of tags. When any tag attempts to send an uplink message, all readers inside the network try to receive the message. If the master reader failed to receive any uplink message, assist readers are might still able to receive it and send it to the master reader.

## 4.5 Chapter summary

In the positioning network, devices are

- Server providing computational assistance
- Readers including a master reader that broadcasts beacon message and handle most messages from tags; assist readers which take part in the positioning process and improve the success rate of uplink messages from tag to reader
- Tags which have three types: critical, sensor, and positioning. Each type of tag generates the same type of data; critical data has the highest priority because of high message value and small delay tolerance; sensor data and positioning data share equal priority. Positioning data require a positioning process to achieve transmission success; sensor data has a relatively high delay tolerance.

Communication between tags and the reader shares a single channel in the project. MAC layer has two function cycles: the discovery cycle and the positioning cycle. Given the delay tolerances, the estimated joining number of tag exceeds the boundary value of discovery processes to avoid starvation.

As the estimated situation is expected to be a special condition with a short duration. The impact of starvation is considered minor compared to the delay caused by a collision. To reduce the impacts of collisions, random initial process selection and reuse of available positioning processes are discussed. Then workflow of tags, master readers, and assist readers are introduced with flow charts and explanations.

# Chapter 5 Simulation results

This chapter investigates the performance of various MAC allocation described in chapter 4. Due to the limited research done in MAC of smart inventory networks (which is the focus of this MAC design), currently, there is no proper choice to compare our result with. This chapter introduces a simulation background, evaluating metric, and simulation results. This chapter evaluates the trade-offs and the performance of random initial slot selection and the retry-period method. The tool used for simulation is MatLab, version R2018a. The main performance objective is to improve the successful transmission rate. The secondary goal is to reduce the average delay without affecting the success rate.

The result of simulations shows that without information between the joining tags, modulation random initial slot selection mechanism can provide the maximum successful transmission rate. Completely random selection provides a similar success rate. Other simulation shows adding a joining period boosts the successful transmission rate (STR) of critical tag transmission significantly while sacrificing the STR of positioning tag. Due to the length of maximum delay tolerance, STR of the sensor tag is not affected by this method.

## 5.1 Simulation background

The target area we consider in our simulation is an indoor environment similar to a warehouse: approximately $50 \times 50$ meters square space with the possibility of obstacles of various materials that affect transmission. The reasons for evaluating performance through simulations instead of conducting filed experiments are:

- The available DecaWave 1001 modules available are not enough to test the case with the average number of tags that can provide a consistent result.
- Any modification of MAC layer design may require re-programming of every tag involved in the experiment. Besides, it requires a large number of experiments to reduce the error of random unexpected situations.
- In the simulation, it is quick to modify any parameters if needed. The resources available are limited by the computational ability of the device performing simulation rather than real-life hardware.

We consider three types of devices in the network: readers, tags and a server. There exists only one server in each network. The server is assumed to have infinite computation power and energy and only communicates with the reader to provide computational assistance.

The minimum requirement for performing any positioning is 3 readers. The total number of active readers are assumed to be within the range of 3 to 10. Readers are spread in fixed positions inside the network to make sure efficient communication coverage. We consider one reader as the Master reader. Every active reader receives uplink transmissions from tags. Ideally, the reader network can maintain a stable data reception if no collision happens.

The estimated number of tags is discussed in chapter 4, section 4.2.3. All tags are in the listening mode to receive a BM after activated. Then activated tags follow the algorithm discussed in section 4.4. Joining the behavior of tags is selected to be the normal type.

We focus on the study of collision and its impact on the performance. In real-life situations, there exist unexpected failure data transmission due to the moving obstacles, the reflection of the environment, clock error, etc. This kind of situation is not considered currently in the simulation because of the randomness and complexity. In the following simulations, if one message is transmitted, it is considered to be guaranteed to reach the destination device. Thus, when multiple messages are transmitted simultaneously they are guaranteed to cause collisions. Every beacon message is received by all active tags, and all PPs are successful. In the current simulation, the locations of critical tags are assumed already known, and critical tags are fixed in location. Critical messages alone are enough for alerts.

The simulations use 1 millisecond as the basic time unit. This means all the simulation results cannot give any difference lesser than 1 millisecond. This is because though the internal clock inside DecaWave 1001 modules is lesser than 1 millisecond, the network requires 1 millisecond to detect any signal. This value is acquired by testing DecaWave 1001 module devices.

Decawave 1001 module user manual suggests that only channel 5 can meet industrial certifications. We consider only one available channel in the design and its simulation. Design for multiple channel situation is one of the further research goals.

A tag joins the network by either turning the power on, regaining stable communication after retries, or awaking from a sleep cycle. In the simulation, the delay calculation of each message starts at the first transmission attempt. Collisions mean one data transmissions fail, but the tag can still retry sending the same message when possible. A message expires when the delay exceeds the maximum delay tolerance level.

The simulation aims to explore and guide the current design. The main performance objective of the MAC design is to maximize the successful transmission. As calculated in chapter 4 the maximum slot is expected to be less than 82 to avoid starvation of critical tags and position tags. Simulation explores the performance of the network from 2 to 1000 available slots. Because the estimated situation is expected to be the initial joining period, emergency, and periodically reset the network. One discovery cycle (DC) and its corresponding positioning cycle are considered enough for most of the simulation.

Table 5.1 Parameters of simulation

| Name | Value | Description |
|---|---|---|
| Base time unit | 1 millisecond | Base time unit of simulation, which is also the minimum unit for errors and difference |
| Channels available | 1 | This is acquired from the customer before design, only ultra-wideband channel 5 is available |
| Maximum number of discovery processes | 1000 | To avoid the starvation of positioning tags, discovery processes should not exceed 82. Critical tags are able to retry in the positioning cycle and thus not considered. However, 82 may not achieve the best performance of the network with the effect of collision and retries. |
| Joining behavior of tags | Normal | Detailed discussion is in Chapter 4, section 4.2.3 |

## 5.2 Performance metrics

The following metrics are useful to consider for evaluating the performance of the MAC layer protocol.

1. The success rate of CM transmission
2. The success rate of PM transmission
3. The average delay for successful CM transmission
4. The average delay for successful PM transmission
5. The success rate of SM transmission

Some of these metrics are of higher importance than others. The successful rate of CM transmission has the highest priority because the critical message has the highest value because it indicates an emergency incident. Another reason is that critical tags are expected to join the network with the shortest delay.

PM messages are regular data transmissions in the network that is of lesser importance than CM. The retry period for critical tags is added before the positioning period. This means trade-off exists between TSR (Transmission Success Rate) of CM and PM transmission.

Delay of successful transmissions has less priority in evaluation. A lesser delay means extra time to handle any unexpected incident for successful transmission. Because SMs have a relatively large maximum delay tolerance, which is larger than the simulated period, TSR describes the ability of the network to reduce the transmission delay of SMs.

## 5.3 Random process simulation

According to the detailed design in chapter 4, random initial process selection has two options that are also considered for the simulation in this section:

- Modulo result selection: $(ID\ \boldsymbol{mod}\ S_{ava}) + 1\ = S_{ini}$
- Random selection: $rand(S_{ava}) = S_{ini}$

Because random initial process selection affects the DC cycle only, and the estimated situation is expected to be not continuous, one initial DC is simulated to evaluate the performance of the random initial process section mechanism.

For comparison purposes, another random method is also simulated

- The random number generated by Poisson distribution with rate parameter lambda equals to half of the available slot $S_{ini} = poissrnd(\lambda), \lambda\ = \frac{S_{ava}}{2}\ \ in\ Matlab$

This initial process selection method has a high possibility of choosing the process in the middle of the discovery cycle. This is expected to create a large number of collisions in the second half of DC and boost TSR in the first half of DC. Because of the estimated number of tags joining is large, this method should have less value than the other two methods. Delays of non-CMs are counted as a whole because only one DC is considered, which means there is no difference between SMs and PMs.

The plot for simulation results is shown below. The number of available processes (AVAPs) ranges from 2 to 1000 in Figure 5.1 (a). In order to better understand the transition in successful rate we plotted the case for AVAPs from 2 to 100 in Figure 5.1(b).

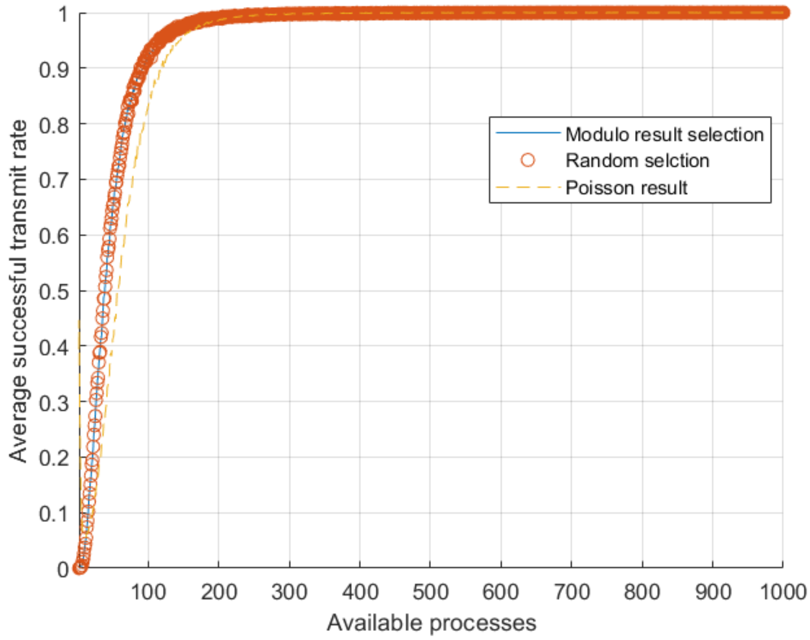



Figure 5.1 (a) TSR of the critical messages in one DC, AVAP 2 to 1000

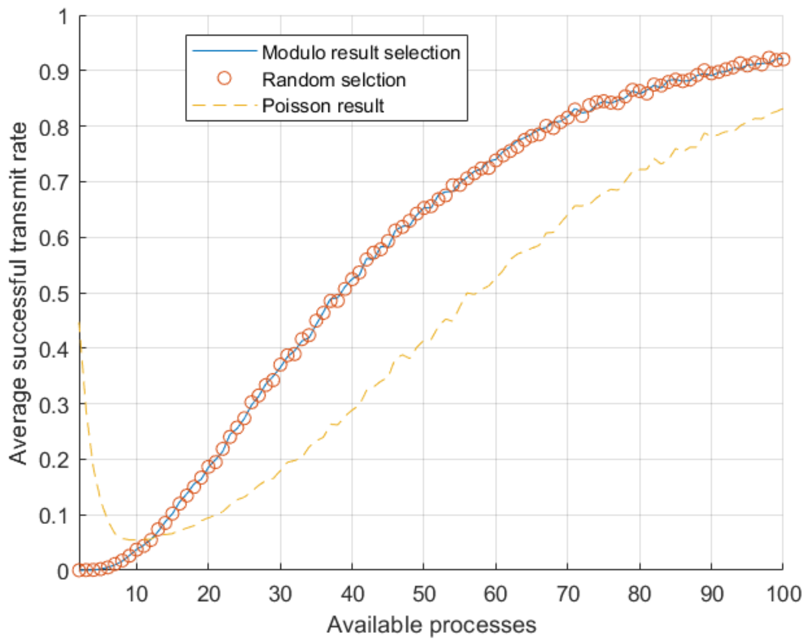



Figure 5.1 (b) TSR of the critical messages in one DC, AVAP 2 to 100

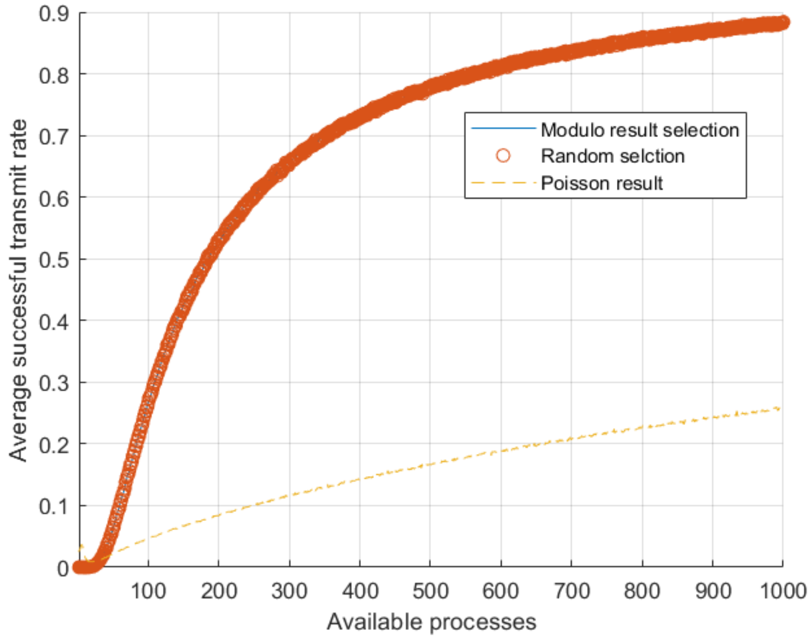



Figure 5.2 (a) TSR of non-critical messages in one DC, AVAP 2 to 1000

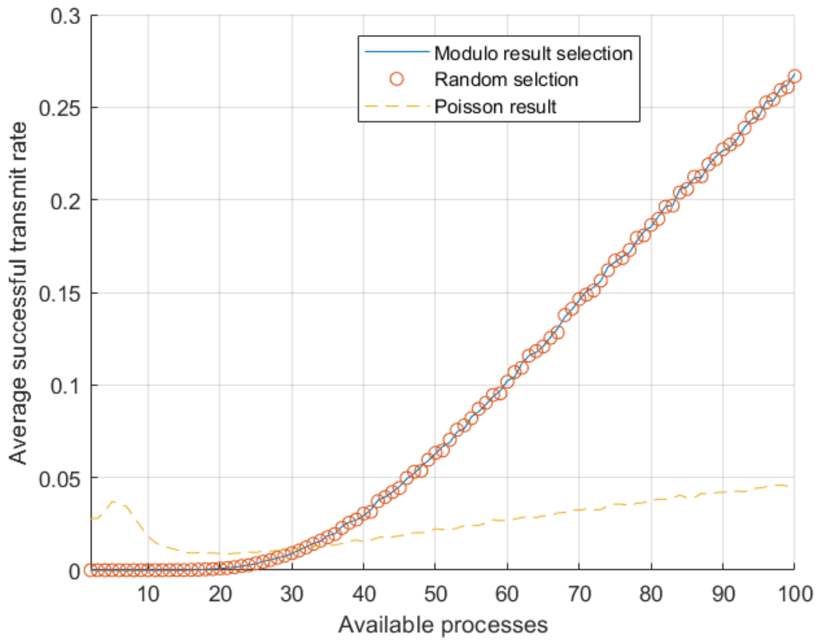



Figure 5.2 (b) TSR of non-critical messages in one DC, AVAP 2 to 100

For the current background of design where tags have no information about any other tags in one DC. Compared to Poisson selection with $\lambda$ equals the mean value of available processes number, the modulo selection and random selection increase the success rate of all communications with a small influence on the delay of successful communications. Modulo and random selection methods have similar effects on the success rate and average delay for all three transmission types.

TSR for both critical and non-critical increases with more AVAPs. Poisson clusters selection of initial processes at half of the maximum number of AVAPs. Thus it can increase the success rate when too few processes are provided because most collision happens in the same range of processes. The Poisson selection may be an option suitable for the situation where AVAPs are too few.

For 90 AVAPs, TSR for critical messages is 89.5%, while TSR for non-critical messages is 22.7%. This indicates that for a pair of DC and PC, TSR for non-critical messages is expected to be lower than 22.7% when the number of AVAP is 90.

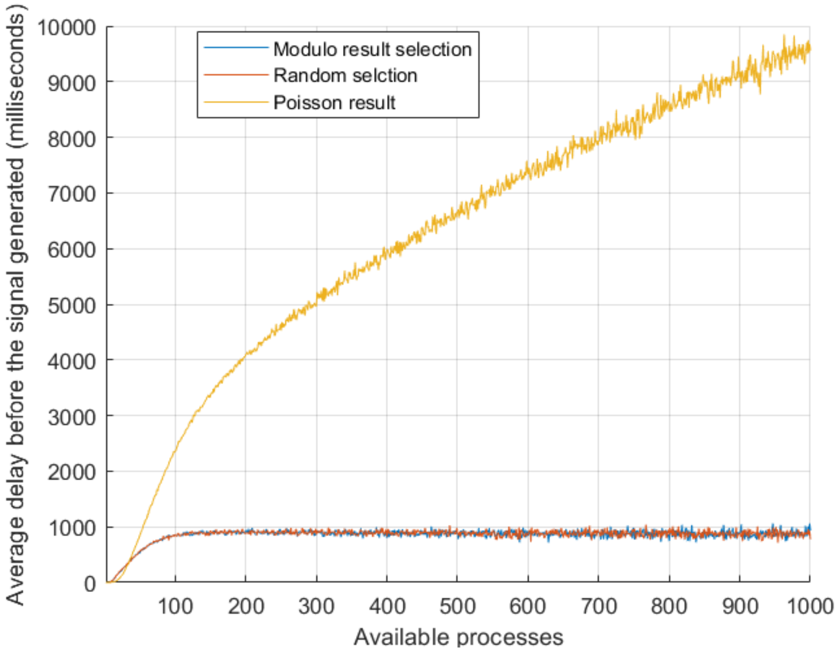



Figure 5.3 (a) Average delay of critical data transmission in one DC, AVAP 2 to 1000

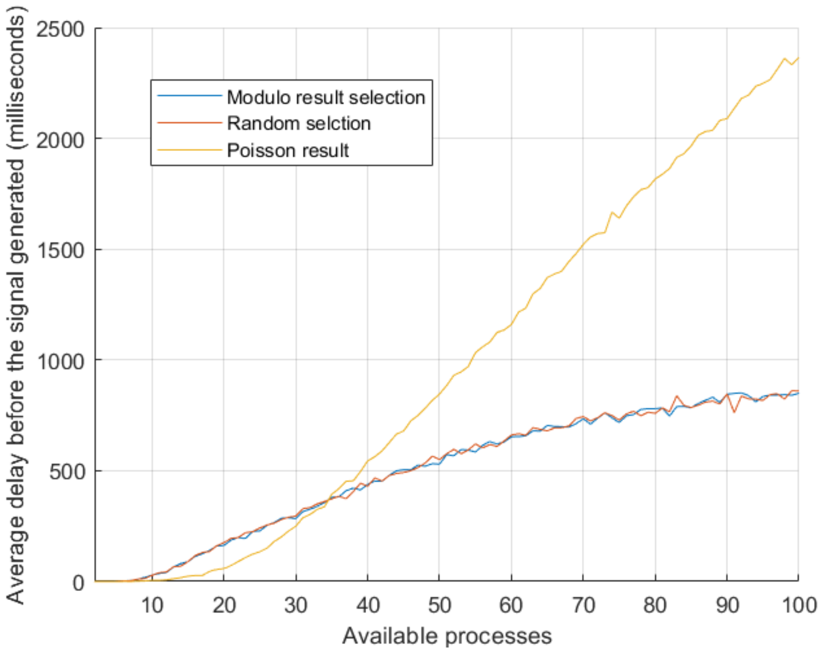



Figure 5.3 (b) Average delay of critical data transmission in one DC, AVAP 2 to 100

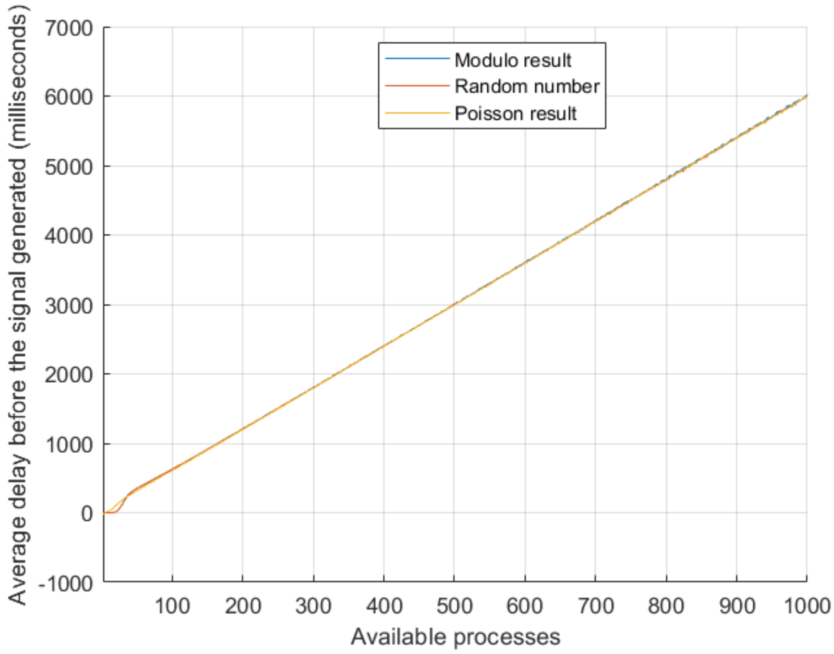



Figure 5.4 (a) Average delay of non-critical data transmission in one DC, AVAP 2 to 1000

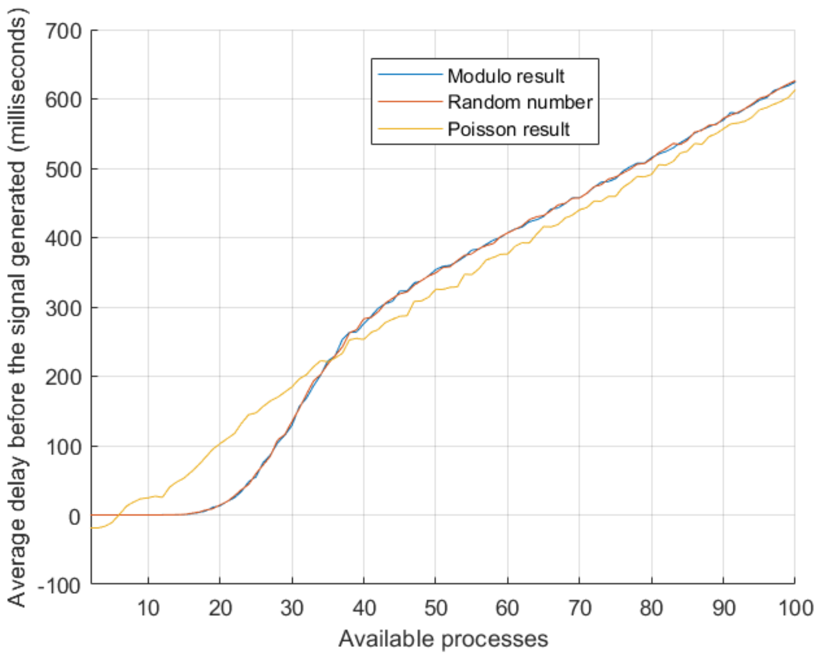



Figure 5.4 (b) Average delay of non-critical data transmission in one DC, AVAP 2 to 100

For small numbers of AVAPs, the Poisson selection provides a shorter delay for CM transmissions. However, this compromises with a significant drop in TSR. At 30 AVAPs, the average delay for CM is 253.9 milliseconds and TSR is 17.9% for the Poisson selection, while modulo and random selection methods show an average delay of 309.1 milliseconds and 36.7% TSR. As the number of AVAPs increases, the Poisson selection experiences collisions causing average delay to increase, while the modulo and random selection methods can achieve a stable delay around 1 second. For non-critical messages, transmission delay has a positive linear relationship with the number of AVAPs.

## 5.4 Joining period in positioning cycle

Details of adding joining period in PC where critical tags can retry CM transmission are explained in chapter 4. The relationship between AVAPs and network performance is investigated by simulating one DC and PC for a given joining behavior of tags.

Three designs are simulated and compared in this section:

- Original design: It does not have a joining period in PC. Its primary purpose is for comparison.
- One attempt rejoin design: Joining period decided by the number of collisions in DC. Critical tags are allowed to have one joining attempt in a joining period.
- Infinite attempt rejoin design: With joining period in PC, critical tags can retry infinite times rejoining the network.

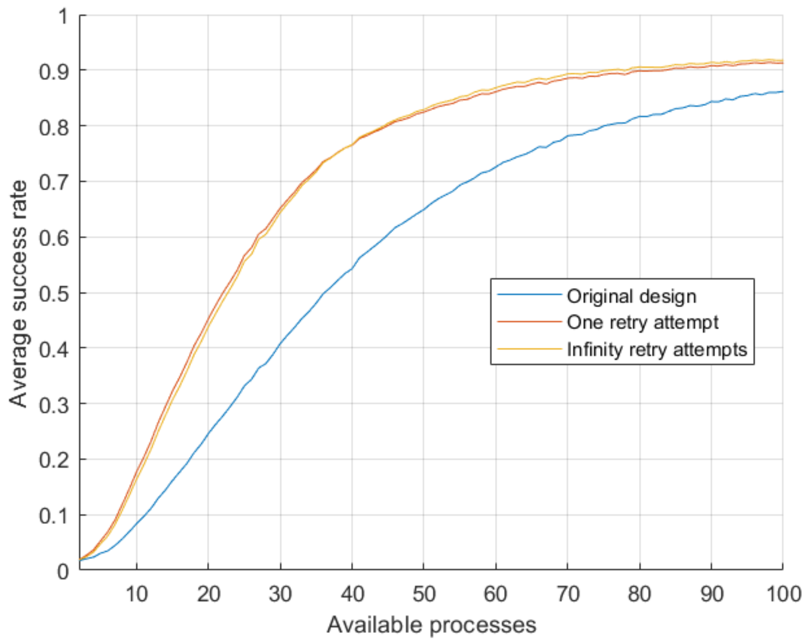



Figure 5.5 Average success rate for CM in one pair of DC and PC, AVAP from 2 to 100

The simulation result in Figure 5.5 shows that in one pair of DC and PC with tag joining as estimated situation explained in chapter 4, the joining period improves the success rate for CMs. With joining period, if critical tags using random initial process selection method, for AVAPs from 2 to 39, allowing critical tags to have one joining attempt provides the best TSR for CM; for AVAPs from 40 to 100, allowing critical tags retry infinite times provides best TSR for CM. The joining period gives 8% to around 20% increase in TSR when the number of AVAPs ranges from 10 to 100. If the tags join the network at an unexpected time, the TSR of CM is 90.5% for AVAPs to be 82, which is the maximum number of available processes to avoid CM starvation during positioning cycle,. Joining period design can provide more than 80% TSR for AVAPs over 45.

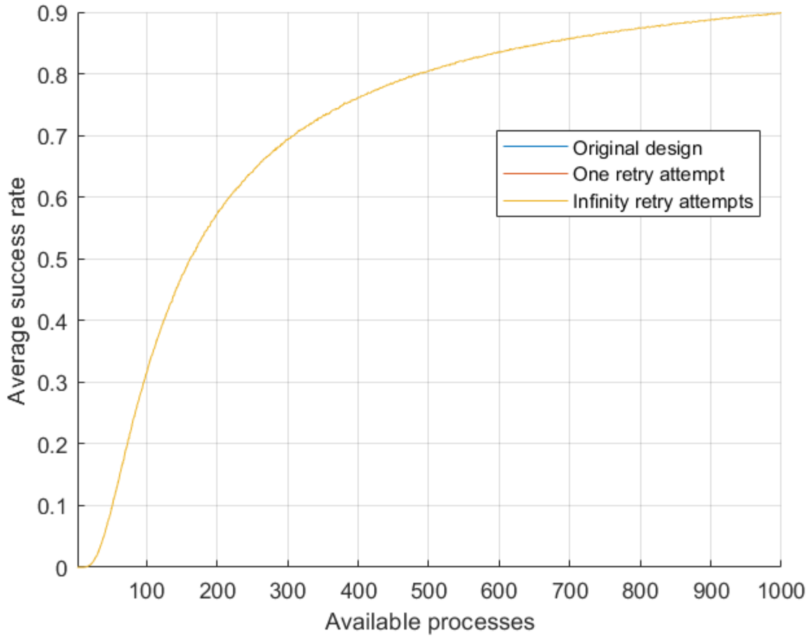



Figure 5.6 TSR for sensor messages in one pair of DC and PC, AVAP from 2 to 1000

Due to the high maximum delay tolerance for sensor messages, introducing a joining period in PC does not affect TSR for sensor messages for AVAPs ranging from 2 to 1000.

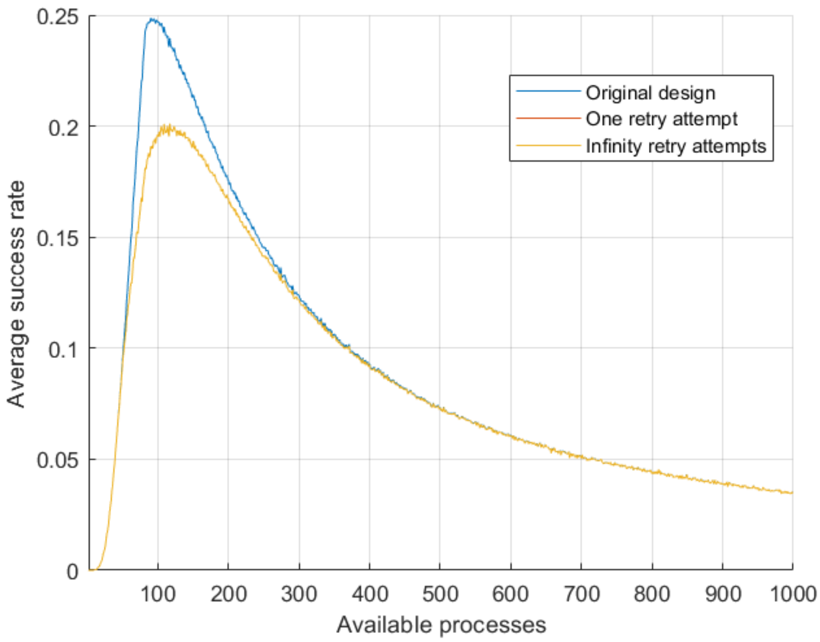



Figure 5.7 (a) TSR for positioning messages in one pair of DC and PC, AVAP from 2 to 1000

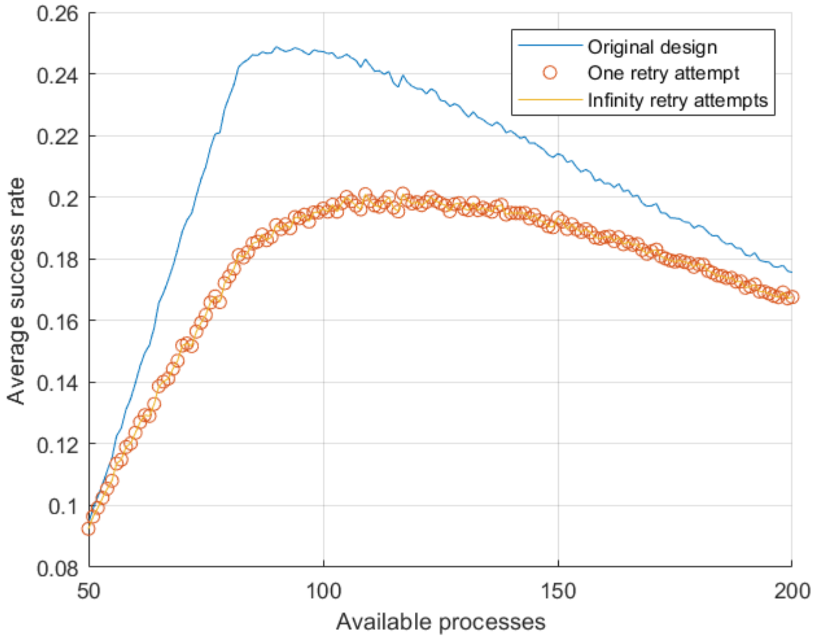



Figure 5.7 (b) TSR for positioning messages in one pair of DC and PC, AVAP from 50 to 200

Figure 5.7 (a) and (b) show the joining period in PC trades off TSR for positioning tags. Compared with the original design, joining tag reduces up to 5% success rate for positioning messages. The maximum TSR of PM, which is approximately 20.1%, can be achieved when AVAPs is 109. For maximum estimated joining, 20 tags out of the total 100 positioning tags, require positioning processes. This is less than 32, which is the maximum number of PPs to avoid starvation during positioning.

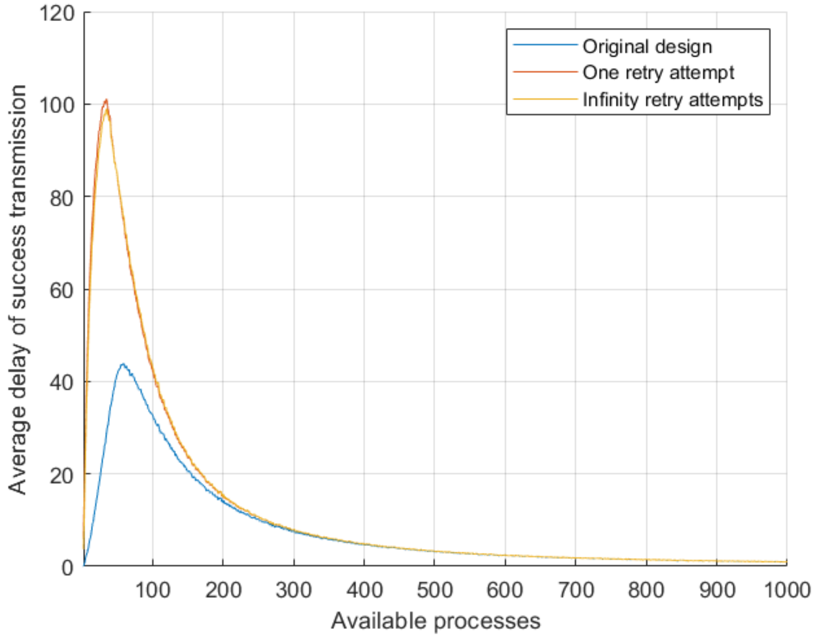



Figure 5.8 (a) Average delay for critical messages in a pair of DC and PC, AVAP from 2 to 1000

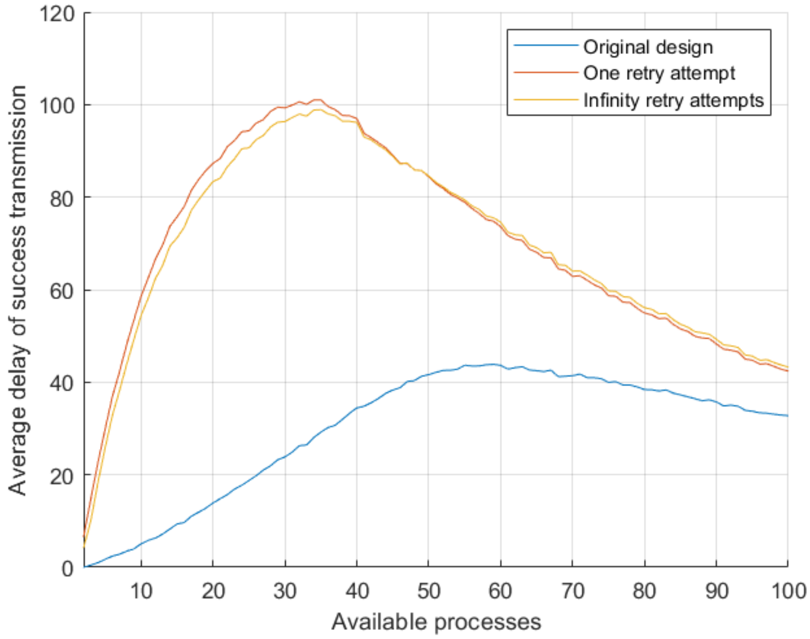



Figure 5.8 (b) Average delay for critical messages in a pair of DC and PC, AVAP from 2 to 100

The average delay for successful message transmissions of CM with the joining period in the positioning cycle reaches peak values, which is approximately 100 millisecond for 30 to 40 AVAPs. The original design has a lesser delay for successful CM transmissions with its peak value of 43.7 milliseconds. The average delay of CM decreases as the number of AVAPs increases but it cannot reach zero. For less than 50 AVAPs, infinite retry shows lesser average CM delay. For more than 50 AVAPs, one attempt rejoining shows lesser average CM delay.

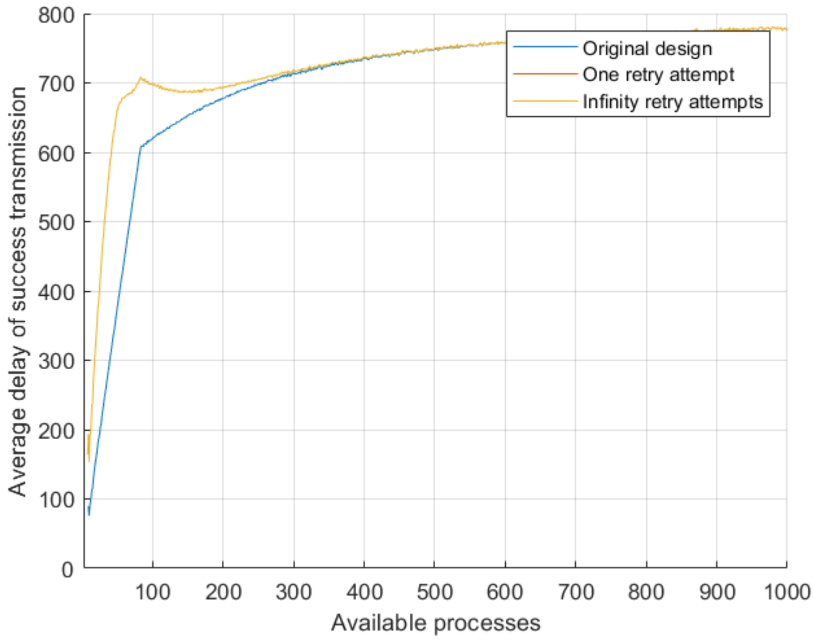



Figure 5.9 Average delay for positioning messages in a pair of DC and PC

Figure 5.9 shows the relationship between the delay of success transmission of PM and the number of AVAPs is akin to logarithmic, with the limit reaching 700 to 800 milliseconds for AVAPs ranging from 2 to 1000.

The above performance evaluation shows that for the estimated number of tags joining the network, infinite rejoining attempt and AVAPs around 109 is able to ensure TSR of CM to be over 90% and a best TSR(approximate 20%) possible for PM. For initializing and resetting the network, which can be predicted, this number of AVAPs and introducing joining period in PC is considered to be the best design choice. For an emergency situation, if the number of AVAPs is over 45, introducing a joining period in PC can provide a minimum of 80% TSR of CM transmission.

## 5.5 Chapter summary

This chapter discusses the simulation results for the three cases of the network: initializing, resetting, and unexpected emergencies. Simulation investigated the impact of the selection algorithm and introducing a joining period in PC on the overall MAC performance. For initial process selection, modulo and random selection methods show no significant performance differences when tags inside the network do not have any information about each other.

Simulation results show introducing joining period in PC can improve TSR of critical messages by 10% to 20% at the cost of up to 5% decrease in TSR of positioning messages. Due to the delay tolerance of sensor messages, the joining period does not have an effect on their TSR. Considering the significance of CMs is higher than PM, joining period in PC are can improve the overall system performance. For initialization and resetting of the network, which are predictable events, approximately 109 AVAPs can provide the best TSR for CMs and PMs. The joining period in PC can provide more than 90% TSR for critical messages for above 45 AVAPs. However, the joining period in PC increases the average delay for successful transmissions from 44 milliseconds to 100 milliseconds, while 100 milliseconds is 1/5th of the maximum delay tolerance for critical messages,.

Further, limiting joining attempts of critical tags during the joining period in PC is shown to have an effect on network performance: and one attempt shows better network performance than infinite attempts for AVAPs less than 50.

# Chapter 6 Conclusions and future works

## 6.1 Conclusion

In the project, we studied an indoor inventory system with wireless positioning and developed a MAC layer design. Parameters are specified in a particular range and closely related to real-life applications. Devices are based on Decawave 1001 modules and were able to perform UBW communications and localization. Existing LLDN in 802.15.4e and the default positioning MAC layer protocol is used as a reference to the MAC layer design.

Firs the positioning method is described in chapter 2 which shows that it requires one reader exchanging messages with tags and at least two readers to assist positioning.

Then a queuing model is studied in Chapter 3 to analyze the network, which serves as a reference for subsequent MAC layer protocol design. Preliminary analysis suggests that by adaptively allocating the channel resources of the network to three types of tags, service efficiency can be improved. This result also guides the direction for further improvement.

Due to the unique characteristics of the proposed network, there are limited works done in similar systems, and no comparison can be provided in this thesis.

The simulation section focuses on exploring the two improving methods: random initialize process selection and joining period in the Positioning Cycle.

Simulation explores the performance of two options considered currently, which is selecting the discovery process according to modulo result of unique 16-bit tag ID and random select of an available DP. In the current environment where each tag does not have any information about other tags inside the network, the two methods have the same effect on avoiding collisions that could happen in a single discovery cycle from 2 to 1000 AVAPs.

For one pair of DC and PC, the joining period method can boost the TSR for critical message transmission in simulation. However, this method sacrifices TSR for positioning messages. The quality of transmission for sensor messages are not affected by this method due to the large maximum delay tolerance. TSR for the critical message cannot reach 100%. The joining period method can provide TSR over 90% for 82 AVAPs and over 80% for 45 to 81 AVAPs. The maximum TSR of positioning message is less than 21%, which means 21 PPs is the possible maximum. This value means starvation during the positioning process for new tags is impossible. Considering this type of tag joining to be initialization, resetting, or emergency. The MAC layer protocol can provide the best service when available DPs in DC is for initialization and resetting. For an emergency, with the available slot to be 82, the joining period designs can still ensure a success rate for CMs to be over 90%. Thus the simulation results indicate the joining period method is able to improve MAC-layer performance.

## 6.2 Future works

During the design, many new problems were raised. This section explains the problems as future research goals.

### 6.2.1 Analysis Model

Theoretical modes are keys to new design directions. In the thesis, the queue length constraint for the queuing model is $M / M / 1$. Relationships between service rate, queue length, and the time quality constraint parameters were analyzed. However, the model used did not make an estimation of the client arriving. In the future, a more detailed model is to be built, for example, $G / G / n / k$ model. This would provide a relationship among the parameters more clearly than the current one in the thesis.

### 6.2.2 Unexpected communication failures

The current design does not consider situations where metal obstacles move inside the network, block or reflecting communications. Also, because all assist readers are overhearing messages transmitted in-network, every message is assumed to reach the destination. Real-life wireless communication failure was not considered. Future research would consider these factors.

### 6.2.3 Retry time limits

In the simulations in chapter 5, results show that infinite retries of uplink communication of critical messages provide better performance than only one attempt allowed. This shows that the retry time limit for CM could influence network performance. Future work is to build a model and analyze the relationship between retry time limit of critical messages and successful rate of communication.

### 6.2.4 Change in design backgrounds

The MAC layer protocol in the thesis is closely related to the application, and the real-life constraint of the Decawave 1001 module limits the available channel to be UBW channel 5. Parameters might change the number of joining tag. For every different situation or environment, this parameter is expected to change. Another parameter, such as the time for message processing, transmitting time, and guard time also varies with different hardware.

As technology advances, multiple channels are likely to be available in the future. This is expected to have a major impact on the design. One possible impact is all positioning processes use another channel and no longer creates starvation because of preventing random data communication. Future works can explore more about the relationships between parameters and design for multiple available channel situation.

# Bibliography


[1] G. Oguntala, R. Abd-Alhameed, S. Jones, J. Noras, M. Patwary, and J. Rodriguez, "Indoor location identification technologies for real-time IoT-based applications: An inclusive survey," *Computer Science Review,* vol. 30, pp. 55-79, 2018/11/01/ 2018.

[2] M. Riad, A. Elgammal, and D. Elzanfaly, "Efficient Management of Perishable Inventory by Utilizing IoT," in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2018, pp. 1-9.

[3] X. Jing and P. Tang, "Research and Design of the Intelligent Inventory Management System Based on RFID," in *2013 Sixth International Symposium on Computational Intelligence and Design*, 2013, pp. 8-11.

[4] M. Wang, J. Tan, and Y. Li, "Design and implementation of enterprise asset management system based on IOT technology," in *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*, 2015, pp. 384-388.

[5] L. Zhang, N. Alharbe, and A. S. Atkins, "An IoT Application for Inventory Management with a Self-Adaptive Decision Model," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2016, pp. 317-322.

[6] B. S. S. Tejesh and S. Neeraja, "Warehouse inventory management system using IoT and open source framework," *Alexandria Engineering Journal,* vol. 57, pp. 3817-3823, 2018/12/01/ 2018.

[7] O. Rholam, M. Tabaa, and F. M. e. Abbas Dandache, "Smart Device for Multi-band Industrial IoT Communications," *Procedia Computer Science,* vol. 155, pp. 660-665, 2019/01/01/ 2019.

[8] W. Chao, Z. X. Feng, C. Xi, X. L. dong, and W. Ying, "Application and Research of State Grid Assets Automatic Inventory Based on Cellular Narrow Band and Low Power WAN Technology," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 2018, pp. 1-5.

[9] Y. Wang, X. Geng, F. Zhang, and J. Ruan, "An Immune Genetic Algorithm for Multi-Echelon Inventory Cost Control of IOT Based Supply Chains," *IEEE Access,* vol. 6, pp. 8547-8555, 2018.

[10] F. Caro and R. Sadr, "The Internet of Things (IoT) in retail: Bridging supply and demand," *Business Horizons,* vol. 62, pp. 47-54, 2019/01/01/ 2019.

[11] S. M. Kamruzzaman, M. Jaseemuddin, X. Fernando, and P. Moeini, "Wireless Positioning Sensor Network Integrated with Cloud for Industrial Automation," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, 2017, pp. 543-546.

[12] A. Hameed and H. A. Ahmed, "Survey on indoor positioning applications based on different technologies," pp. 1-5.

[13] M. Anwar and Y. Xia, "Ieee 802.15. 4e lldn: Superframe configuration for networked control systems," in *Proceedings of the 33rd Chinese Control Conference*, 2014, pp. 5568-5573.

[14] A. Homay, M. de Sousa, and L. Almeida, "Nash equilibrium for proactive anti-jamming in IEEE 802.15. 4e (Emerging wireless sensor actuator technologies for I4. 0)," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 161-167.

[15]Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Impact of mobility on the IoT MAC infrastructure: IEEE 802.15.4e TSCH and LLDN platform," pp. 478-483.

[16]H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "IEEE 802.15. 4e in a nutshell: Survey and performance evaluation," *IEEE Communications Surveys & Tutorials,* vol. 20, pp. 1989-2010, 2018.

[17]A. R. J. Ruiz and F. S. Granja, "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis," *IEEE Transactions on instrumentation and Measurement,* vol. 66, pp. 2106-2117, 2017.

[18]J. Kulmer, S. Hinteregger, B. Großwindhager, M. Rath, M. S. Bakr, E. Leitinger*, et al.*, "Using DecaWave UWB transceivers for high-accuracy multipath-assisted indoor positioning," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 1239-1245.

[19]A. F. Molisch, K. Balakrishnan, C.-C. Chong, S. Emami, A. Fort, J. Karedal*, et al.*, "IEEE 802.15. 4a channel model-final report," *IEEE P802,* vol. 15, p. 0662, 2004.

[20]B. Bitencort, F. Vasques, P. Portugal, and R. Moraes, "SAS: Slot allocation scheme for the IEEE 802.15.4e LLDN mode," pp. 4713-4718.

[21]M. Anwar, Y. Xia, and Y. Zhan, "TDMA-Based IEEE 802.15.4 for Low-Latency Deterministic Control Applications," *IEEE Transactions on Industrial Informatics,* vol. 12, pp. 338-347, 2016.

[22]M. Drobczyk and H. Martens, "A study on low-latency wireless sensing in time-critical satellite applications," in *2016 IEEE SENSORS*, 2016, pp. 1-3.

[23]M. Raza, H. Le-Minh, N. Aslam, S. Hussain, and W. Ellahi, "A control channel based MAC protocol for time critical and emergency communications in industrial wireless sensor networks," in *2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, 2017, pp. 122-126.

[24]I. Juc, O. Alphand, R. Guizzetti, M. Favre, and A. Duda, "Energy Consumption and Performance of IEEE 802.15. 4e TSCH and DSME," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, pp. 1-7.

[25]W.-P. Wang and R.-H. Hwang, "A distributed scheduling algorithm for IEEE 802.15. 4e networks," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 2015, pp. 95-100.

[26]G. Patti and L. L. Bello, "A priority-aware multichannel adaptive framework for the IEEE 802.15. 4e-LLDN," *IEEE Transactions on Industrial Electronics,* vol. 63, pp. 6360-6370, 2016.

[27]B. Bitencort, R. Moraes, P. Portugal, and F. Vasques, "AdapTA: Adaptive timeslot allocation scheme for IEEE 802.15. 4e LLDN mode," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 304-309.

[28]H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15.4e," pp. 1-9.

[29]M. Ojo and S. Giordano, "An efficient centralized scheduling algorithm in IEEE 802.15. 4e TSCH networks," in *2016 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2016, pp. 1-6.

[30]Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Tackling mobility in low latency deterministic multihop ieee 802.15. 4e sensor network," *IEEE Sensors Journal,* vol. 16, pp. 1412-1427, 2015.

[31]T. P. Duy and Y. Kim, "An efficient joining scheme in IEEE 802.15. 4e," in *2015*

*International Conference on Information and Communication Technology Convergence (ICTC)*, 2015, pp. 226-229.

[32]E. R. Magsino, I. W.-H. Ho, and Z. Situ, "The effects of dynamic environment on channel frequency response-based indoor positioning," in *28th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2017, October 8, 2017 - October 13, 2017*, Montreal, QC, Canada, 2018, pp. 1-6.