Performance Prediction and Analysis of a Dynamic, Self-Configuring IoT Network

by

James Robert Hershaw

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr. R. Wolhuter
Co-supervisor: Prof. T. Niesler

April 2022

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ..... April 2022 .....

Abstract

Performance Prediction and Analysis of a Dynamic, Self-Configuring IoT Network

J. R. Hershaw

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MEng (Electr.)

April 2022

In this thesis a hypothetical wireless sensor network is modelled in software using discrete event simulation before a theoretical predictive model is developed for this hypothetical network using queuing theory.

A study of cases in which wireless sensor networks have been implemented both practically and in simulation is conducted before a radio module is selected with which the models can be parameterised. Various appropriate MAC layer and routing protocols are investigated and the MACAW MAC layer and AODV routing protocols are selected to be implemented in the simulated and theoretical models.

The results of both the simulated model and the theoretical model are compared across a number of different possible network topologies and different transmission arrival rates. The theoretical model was able to accurately predict the behavior of the network at duty cycles appropriate for this type of network.

Uittreksel

Prestasievoorspelling en ontleding van 'n dinamiese, selfkonfigurerende IoT-netwerk

("Performance Prediction and Analysis of a Dynamic, Self-Configuring IoT Network")

J. R. Hershaw
Departement Elektriese en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MIng (Elektr.)
April 2022

In hierdie tesis word 'n hipotetiese draadlose sensornetwerk in sagteware gemodelleer deur gebruik te maak van diskrete gebeurtenissimulasie voordat 'n teoretiese voorspellingsmodel deur toestandteorie vir hierdie hipotetiese netwerk geontwikkel is.

'n Studie van gevalle waarin draadlose sensornetwerke beide prakties en in simulasie geÃ¯mplementeer is word uitgevoer voordat 'n radiomodule gekies word waarmee die modelle geparameteriseer kan word. Verskeie toepaslike MAC-laag en roeteringsprotokolle word ondersoek en die MACAW MAC-laag en AODV roeteringsprotokolle word gekies om in die gesimuleerde en teoretiese modelle geÃ¯mplementeer te word.

Die resultate van beide die gesimuleerde model en die teoretiese model word vergelyk oor 'n aantal verskillende moontlike netwerktopologieÃ« en verskillende transmissie-aankomskoerse. Die teoretiese model was in staat om die gedrag akkuraat te voorspel van die netwerk by dienssiklusse wat geskik is vir hierdie tipe netwerk.

Acknowledgements

I would like to express my sincere gratitude to the following people

- Dr. R Wolhuter for his invaluable insight and guidance
- Professor T. Niesler and the RhinoNet research group
- My parents, whose support cannot be understated

Dedications

For Jürgen Klopp

Contents

List of Figures

List of Tables

Nomenclature

Constants

g =	$9.81\,\mathrm{m/s^2}$	

Variables

Re_{D}	Reynolds number (diameter)	[ ]
x	Coordinate	[m]
$\ddot{x}$	Acceleration	[$\mathrm{m/s^2}$]
θ	Rotation angle	[rad]
τ	Moment	[N·m]

Vectors and Tensors

$\overrightarrow{v}$	Physical vector, see equation ...

Subscripts

a	Adiabatic
a	Coordinate

Chapter 1

Introduction

1.1 Introduction

Telemetry networks are used across a variety of industries to monitor, control and maintain infrastructure[34]. These industries include, but are not limited to water, energy, mining, wildlife tracking, etc.

The network proposed in this thesis would be used to monitor water telemetry in remote regions of the Western Cape, South Africa. To that end, the network will make use of RF technology operating in the unlicensed sub-gigahertz ISM frequency bands, as it is cost effective, has good range depending on the frequency, and lower power consumption than similar equipment operating in the higher frequency bands. While the network is designed with water telemetry in mind, it could be applied to another type of monitoring network with a similar topology, provided that the appropriate sensors are used.

In order to accommodate the remote nature of the network, it will have to be a multi-hop network with routing capabilities, as it cannot be guaranteed that all nodes will be visible to each other. Due to the mostly theoretical approach for this project, power consumption is not a major consideration in this case. However, it should still be kept in mind so that the models could still be applicable, should the network be implemented in a practical setting.

Due to the deployment environment of this type of network, the choice of communications and routing protocol is very important, in order to ensure data transfer reliability. This project set out to investigate the performance, predictability and efficiency of typical routing topologies for networks of this type. The availability of a set of theoretical and simulated tools following such an investigation, would be very useful during the planning process of such an application.

1.2 Project objectives

Further to the above, the following objectives were identified:

- Present a study of LPWAN implementations in practical systems
- Present an overview of some possible LPWAN technologies prior to making a decision regarding the communications technology for the proposed network
- A suitable RF module must be selected in order to parameterise the modelled network
- To conduct a study of various suitable MAC layer and routing protocols
- To implement the selected protocols in a simulation environment
- To investigate methods based on queuing theory for creating a theoretical predictive model for the network
- To compare the results of the simulated and theoretical models

For the purposes of parameterising the models discussed later in this thesis, a specific radio module was required. The unit that was selected is the RFM98W radio transceiver, which makes use of the LoRa LPWAN technology for modulation. This particular model operates at a frequency of 433 MHz. MACAW was chosen as the MAC layer protocol to be implemented in both simulation and theoretical models, and AODV as the routing protocol.

Using the Discrete Event Simulation toolkit DESMO-J, the network is simulated for a number of different possible topologies and duty cycles. It was found that the simulated network presented good results at lower duty-cycles, the likes of which would be more suitable for a wireless sensor network of this nature, particularly for battery conservation purposes.

Using both Poisson and Markov processes, a number of theoretical predictive models were created using queuing theory. The models developed using the Poisson process were discarded as the resulting predictions were not within an acceptable threshold of the simulation results. Of the Markov process models, two of the three proposed models provided acceptable results, with the final model producing the best results. This final model was able to predict the performance of the network to a high degree at lower duty cycles.

1.3 Contributions

The contributions of this thesis include:

- A simulation framework that is reasonably flexible and can be adapted to different network layouts. This is a useful tool for the purposes of estimating network performance prior to committing resources to deployment.

- An acceptably accurate theoretical predictive model was developed which builds upon previous work, but is more general. The results of which correspond acceptably to those of the simulations.

1.4 Overview of Thesis

1.4.1 Chapter 2

Chapter 2 will present case studies where LPWANs have been used in academia and in industry to link various sensor nodes, technologies used and any benefits or shortcomings these applications may have experienced.

1.4.2 Chapter 3

Chapter 3 contains further research of a number of LPWAN technologies which would be considered in the proposed network. This is followed by an investigation into possible RF radio modules which might be suitable with the chosen LPWAN technology.

1.4.3 Chapter 4

Chapter 4 serves as a study of the Data Link Layer. A number of possible MAC Layer protocols will be investigated before one is chosen to implement in the proposed network.

1.4.4 Chapter 5

In keeping with the treatment of the Data Link Layer in Chapter 4, Chapter 5 contains an investigation of the Network Layer, prior to selection of an appropriate configuration for the specific network.

1.4.5 Chapter 6

Chapter 6 covers the importance of simulation as a part of the network design process and presents a brief overview of Discrete Event Simulation in general. The Discrete Event Simulation toolkit known as DESMO-J is then discussed in depth. Finally the implementation in DESMO-J of the particular MAC Layer and routing protocol selected in Chapters 4 and 5, respectively, is detailed and the results of the simulation presented.

1.4.6 Chapter 7

Chapter 7 provides an introduction to the basics of queuing theory and the modelling of networks using that approach. A number of possible applications

of these basic principles is discussed. The process of modelling the proposed network as a queue using a Markov chain is also covered and applications of this process explained. The results of the applications of both the basic principles of queuing theory and the Markov chain are subsequently compared and discussed.

1.4.7 Chapter 8

Chapter 8 presents final results, conclusions and possibilities for further study.

The current chapter is an introduction to the thesis. The usefulness of wireless sensor networks, as well as some challenges faced by these types of networks, was discussed. The objectives for this project were outlined and a short summary of the thesis presented. Finally an overview of the structure of this thesis was outlined. The following chapter presents a number of case studies of the use of LPWANs in industry and academia.

Chapter 2

Literature Background

This chapter will investigate previous work done in the field of Wireless Sensor Networks. The project objectives will be discussed, followed by the solutions implemented by the authors and the results of the work. Problems encountered during the respective projects, if any, will also be discussed. This chapter will only present a general review for work done in the field of Wireless Sensor Networks. Further literature relevant to the specific sections in this thesis will be reviewed in those specific sections.

Spreeth, 2008-12

The goal of this project was to research the major aspects of a WSN and create a robust prototype hardware setup and network which could be used in a number of applications. The main design concern for this project was the battery life of the sensor nodes. As for most applications of wireless sensor networks this is the major concern. The system makes use of a Zigbee radio module, the Tmote sky. Zigbee makes use of the CSMA-CA MAC layer protocol and the network uses a minimum energy communication protocol which states that in the network, between all connected nodes there is a minimum energy path. The protocol thus revolves around finding these minimum energy paths for transmissions. The project also utilized the Truetime simulation engine to simulate the power consumption in order to provide a baseline with which practical results could be compared. The power-aware routing protocol developed resulted in the identification of very stable, cost effective routes, providing a more than adequate performance for environmental monitoring systems.[37]

Goebel *et al.*, 2012

This paper set out to provide an introduction to the use of the DESMO-J framework for simulating an ad-hoc network, as opposed to the use of the framework for logistical simulations, which is what it is generally used for. The simulation made use of a version of the ALOHA MAC layer protocol and

a routing protocol developed by Goebel *et al.* which focuses on fair resource allocation across the network. The focus of the project was not to maximize the effective throughput of the network, or to maximize battery life and power efficiency, but rather to ensure that the protocol distributed resources fairly across the network as far as possible. Running the simulation using both sparsely and densely arranged fixed infrastructure, it was found that the resources of power consumption and bandwidth were sufficiently fairly distributed, with the more dense fixed infrastructure configuration, on average, using less power, more bandwidth, and completing more transmissions.

Thorstensen *et al.*, 2004

The Electronic Shepherd system was developed to assist sheep and reindeer farmers in tracking their animals during grazing season[40]. The system utilized UHF radio tags on the animals themselves, with only the flock leader equipped with the mobile access point transceiver, in conjunction with GPRS/GSM base stations in a star topology to create a low-cost network. This was not limited to the purposes of animal tracking, but can be utilized for a number of other applications. The UHF on-animal tags made use of MAC layer protocol very similar to the well-known ALOHA protocol, whilst the mobile access point makes use of GPRS to communicate with the fixed access points. The project encountered problems with the stability of the mobile access points due to inconsistent GPRS connectivity. However, using the respective devices' built in software, the UHF tags and mobile access point devices had an uptime of approximately 90%, with the down periods due to extended periods without sunshine to charge the power supplies of the devices. To try and combat the inconsistent GPRS connection on the mobile access points, a 802.11 network configuration was implemented between the mobile access points. This was however abandoned for use as the core network before deployment. It is unclear as to why, but presumably it was not possible to implement it fully before the grazing season started. The biggest challenge for the project was extending the battery life of the end devices as much as possible. It was initially estimated that only 80% of the nominal battery capacity could be expected, due to temperature variations, varying current load, etc. Field testing indicated, however, that the capacity was closer to 50% of the nominal capacity. The devices do have solar recharging capabilities, but further development is still required to extend the battery life for periods of low sunshine.

Wotherspoon, 2019

The goal of this project was to develop a hardware solution for a low power WSN node, both for the on-animal sensor nodes and the fixed nodes, as well as to develop and implement a multi-hop ad-hoc network for communication from the on-animal tags to a server node. This hardware solution included a MCU, RF module, on device storage, a GPS module, a power supply with

solar energy harvesting capabilities, as well as various environmental sensor devices. The details of the hardware design are largely unimportant for the network proposed in this thesis, with the obvious exception of the choice of RF module, and by association the LPWAN technology. Wotherspoon made use of the RFM95W RF module, which is a LoRa device operating at a frequency of 433 MHz. As previously stated, Wotherspoon had to design a multi-hop ad-hoc network, consisting of mobile, on-animal tags communicating with fixed nodes which would then need to relay transmissions across the network in a multi-hop fashion to a server node. A number of protocols, both MAC layer and routing protocols, were investigated before MACAW and AODV were selected and implemented respectively.

The primary focus of the network was reliability in the form of ensuring that the nodes remained powered and minimizing packet loss. A simulation tool was developed to predict long term changes to battery charge under various conditions. This, combined with practical testing of the battery and solar harvesting unit indicated that the node could be powered at the maximum expected load for 10 days without the battery's voltage dropping to below 3.7V.

The networking protocols were implemented on the FIT IOT-LAB platform in Lille, France, a system for testing Internet of Things networks. I was found that for transmissions that required no hops and one hop, no transmissions were lost, and for transmissions requiring two hops, 98.4% of transmissions made it across the network to a server node. Testing the physical nodes in an indoor laboratory setup gave results wherefore in cases with up to two hops, all messages were successfully received by a server node, and with three hops, 99.95% of messages were received.

Overall, the power supply and solar harvesting unit were capable of powering the node for long periods, and the networking protocols were able to successfully relay messages from the on-animal tags to a server node.

Van Staden, 2012-12

This project aimed to investigate a number of MAC layer protocols commonly used in NBT networks and their use in practical systems before creating reliable theoretical predictive models for these protocols using queuing theory, and thereafter supporting these predictive models by means of developing a simulation environment to predict the behaviour of these theoretical networks. The modelled networks all made use of NB-IOT as the LPWAN technology, with the topology for each network being that of a star topology, i.e. there was no routing required and no need for networking protocols for communication between sensor nodes and server nodes. The MAC protocols for which theoretical and simulation models were developed were CSMA-CA, Adaptive Tree Walk, and Round Robin Polling. The predictive models, and supporting simulations, showed that Round Robin Polling outperformed the other protocols

when the generation of new payload transmissions is very high. It was shown that when data frames were small, Adaptive Tree Walk performs best at both low and high arrival rates. The performance of CSMA-CA was reduced by the high data latency of the network. Reducing the likelihood of collisions, and subsequent backoff, and reducing the latency and the size of the data frames, resulted in CSMA-CA outperforming both Round Robin Polling and Adaptive Tree Walk. However, at very high arrival rates, the performance of CSMA-CA degrades completely, while Adaptive Tree Walk still outperforms Round Robin Polling. So to summarize, under certain conditions, any one of these protocols may outperform another.

Amongst the practically implemented networks discussed in this chapter, the battery life of the nodes, as well as the percentage of transmissions lost were the main focus of the design process of the network. This is due to the remote nature of many of these types of networks where reliability is key. It is not feasible to go out to where these devices are deployed in order to change the batteries if they fail, so the nodes must be designed to be very cost effective with regards to power consumption. At the same time there is very little point of having a sensor network if the data that the nodes are sensing is lost because of an unreliable network. As such packet loss must be minimized.

Using the knowledge gained in reviewing the work in this chapter, a network will be designed and modelled with the aim to maximize effective throughput and minimize lost transmissions for the particular application in mind.

2.1 Summary

This chapter summarized a number of cases in which WSNs were deployed or simulated in order to gain a greater understanding of the challenges faced by these types of networks. Of particular interest was the work of Van Staden, who compared the performance of various communications protocols under varying conditions using both the DES toolkit DESMO-J as well as queuing theory.

The following chapter will investigate LPWAN technologies before deciding on which LPWAN technology to model the network. The particular radio module and operating frequency on which the model parameters of Chapters 6 and 7 will be based, are also covered.

Chapter 3

Hardware Considerations

While the proposed network will only be analysed using simulation and theoretical modelling, certain hardware components must be selected. In order to accurately parameterise both the simulation and theoretical models, a very important component is the radio module, the following chapter will report on the radio module proposed for the simulated and theoretical models.

3.1 Choosing a LPWAN Technology

Low Power Wide Area Networks (LPWANs) are made up of nodes that consume little energy to cover large range for machine-to-machine communication with a low data transmission rate[30]. This makes LPWAN technologies well suited for IoT applications as many IoT networks are comprised of devices situated in remote regions and must thus consume little energy while transmitting over a long range.

While different LPWAN technologies are designed to focus on a certain aspect of the network, they are all designed with certain goals in mind, namely:

- Long Range
- Low Power
- Low Cost
- Scalability

The degree to which a LPWAN technology achieves these goals, as well as the priorities of the proposed network, will determine which LPWAN technology is appropriate for the proposed network.

3.1.1 Popular LPWAN Technologies

Many LPWAN technologies have emerged in recent years, such as LoRa, SigFox, NB-IoT, Ingenu, Dash7 and Weightless, despite the fact that the term

"LPWAN" only came into existence after 2013. Of these many LPWAN technologies, the leaders in the field are LoRa, SigFox, and Narrow Band (NB) IoT[28]. In this section, the technical differences between these LPWAN technologies will be further investigated to assess each one's suitability for the proposed network.

3.1.1.1 LoRa

When discussing LoRa as a LPWAN it is first necessary to define and distinguish between its two aspects, LoRa and LoRaWAN.

LoRa is a physical layer technology that is used to modulate the signals for transmission, whereas LoRaWAN is a LoRa-based communication protocol standardized by the LoRa-Alliance[28].

LoRa modulates signals for transmission in unlicensed ISM bands, such as 868 and 433MHz in Europe and Africa, 915MHz in North and South America, etc. LoRa technology provides bidirectional communication by making use of Chirp Spread Spectrum modulation, the benefits of which include high robustness, resistance to multipath fading, low power consumption, low latency, resistance to the Doppler Effect, and immunity to in-band interference.

LoRaWAN makes use of different classifications for end devices depending on the requirements of the IoT application:

Class A-

allow for bi-directional communication. This is the classification with the lowest power consumption and is mostly in sleep mode, waking up according to a schedule to transmit[30].

Class B-

builds on Class A by having scheduled receive windows.

Class C-

is always listening to receive data, except when it is transmitting.

3.1.1.2 SigFox

SigFox is a LPWAN operator that offers IoT connectivity solutions based on its own patented technology. SigFox deploys its own base stations with software defined radios connected to backend servers via an IP based network.

End devices communicate with the base stations using Binary Phase Shift Keying (BPSK) modulation using an extremely narrow band, only 100Hz. Like LoRa, SigFox makes use of unlicensed sub-GHz ISM bands. By using a very narrow bandwidth SigFox experiences very low noise levels, which results in very low power consumption, high receiver sensitivity, and low-cost antenna

design. However, this comes at the expense of a maximum throughput of only 100 bps.

SigFox supports a maximum of 140 uplink messages per end device per day, with a maximum payload of 12 bytes per message, and while SigFox does support bi-directional communication, it is only for a maximum of 4 downlink messages per day, which means that not every uplink message can be acknowledged.

3.1.1.3 NB-IoT

NB-IoT, (Narrow Band IoT), is able to coexist with GSM and LTE in licensed frequency bands with a frequency bandwidth of 200 kHz, equivalent to one GSM or LTE resource block. NB-IoT is capable of three operation modes:

3.1.1.3.1 Stand-alone:

GSM frequency bands are used.

3.1.1.3.2 Guard-band:

Unused resource blocks in an LTE carriers guard band are used.

3.1.1.3.3 In-band:

LTE carrier resource blocks are used.

NB-IoT is capable of supporting connection of 100 000 end devices per cell. This can be scaled up by adding more carriers. NB-IoT makes use of single-carrier FDMA in the uplink and orthogonal FDMA in the downlink. Quadrature phase-shift keying (QPSK) is utilized for signal modulation.

3.1.2 Conclusion

	SigFox	LoRa/LoRaWAN	NB-IoT
Modulation	BPSK	CSS	QPSK
Frequency	Unlicensed ISM	Unlicensed ISM	LTE
Bandwidth	100Hz	250kHz and 125kHz	200kHz
Maximum data rate	100bps	50kbps	200kbps
Bidirectional	Limited	Yes	Yes
Range	40km (rural)	20km (rural)	10km (rural)
Interference immunity	Very high	Very high	Low
Authentication and encryption	None	AER128b	LTE
Allow private network	No	Yes	No

Table 3.1: Comparison of LPWAN technologies

Table 3.1 shows a comparison of various characteristics of the LPWAN technologies being considered. Another aspect of these technologies to consider is the cost. While SigFox and LoRa make use of unlicensed ISM bands, NB-IoT uses licensed LTE bands and will, therefore, not be free. SigFox deploys its own base stations which use patented technology. Therefore the use of this LPWAN technology in this network will not be feasible with the remote nature of this network. LoRa/LoRaWAN has good range, consumes little power, has very high interference immunity, supports bidirectional communication, allows for unlimited communication, is cost effective and configurable with regards to the end devices (class A, B, or C). For these reasons LoRa/LoRaWAN is the LPWAN technology which will be used in this network.

3.2 Choosing an RF module

To enable each node in the proposed IoT network to communicate with each other across the network, a low-power radio transceiver is required. Before choosing a radio transceiver module, however, the environment into which the sensor nodes will be deployed must first be considered in order to make sure the requirements of the network as a whole are met. Upon considering the environment into which the nodes will be deployed, the following requirements for the radio transceiver are defined:

Range-

In order to maximize the efficiency of the network it is important to minimize the number of relay nodes required in the network by maximizing the range of each individual node.

Power consumption-

As the placement of the nodes in the network will be very remote and inaccessible, the radio module should consume as little power as possible in order to conserve the life of the battery powering the node.

Reliability-

The radio transceiver module should provide reliable communication between nodes with minimal transmission errors.

Modulation-

The radio transceiver module must make use of LoRa technology, using CSS modulation.

Operating frequency-

The operating frequency of the radio transceiver must fall under a license free, ISM, band for region 1, enabling the network to be deployed anywhere in Africa and Europe, and parts of the Middle-East and Asia.

Most of the radio transceiver modules available on the market making use of LoRa technology are only enabled to operate at frequencies which are within the ISM bands. The two most important factors when deciding on the radio transceiver module to be used in the proposed network are, therefore, the range and power consumption of the module.

3.2.1 Range

The main influences on the range capabilities of a radio transceiver are operating frequency, transmission antenna, bandwidth, receiver sensitivity, RF power output, the height of the antenna above the ground, and topography.

Generally speaking, when comparing two radio waves travelling through the same medium, transmitted with the same power via identical antennas, the wave with the longer wavelength will be attenuated less. This is shown by simplifying the Friis equation in 3.1 where P_r and P_t represent the received and transmitted power, respectively, of a radio wave received and transmitted by antennas with gains of G_r and G_t respectively, with wavelength λ, where r is the distance between the receiving and transmitting antenna.

$$P_r = P_t \cdot \frac{G_r \cdot G_t \cdot \lambda^2}{(4\pi)^2 r^2} \tag{3.1}$$

When using equation 3.1 to compare two radio transmissions, equation 3.2 is obtained and can be simplified to 3.3, which shows that between two signals travelling through the same medium, being transmitted by the same antenna and with the same power, the signal with the lower frequency is less attenuated.

$$P_t \cdot \frac{G_r \cdot G_t \cdot \lambda_1^2}{(4\pi)^2 r_1^2} = P_t \cdot \frac{G_r \cdot G_t \cdot \lambda_2^2}{(4\pi)_2^2 r^2} \tag{3.2}$$

$$\frac{\lambda_1}{\lambda_2} = \frac{r_1}{r_2} \tag{3.3}$$

Based on this observation, and only considering the effect that frequency has on the attenuation of the radio signal, a lower frequency radio wave will be less attenuated compared to a signal of a higher frequency.

The antenna used by the radio transceiver module, while having a large effect on the range of the module, is not part of the module itself and thus has no bearing when choosing a radio transceiver module.

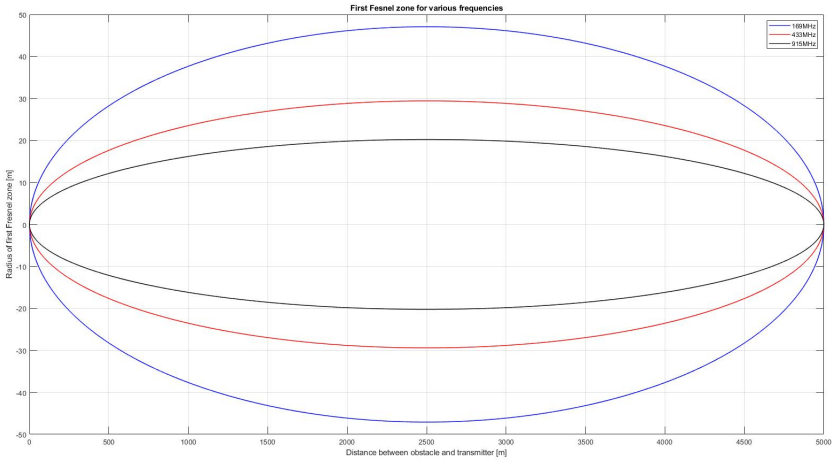


Figure 3.1: First Fresnel Zones of select ISM bands

The height of the antenna, like the antenna itself, does not directly influence the choice of radio transceiver module, but rather, the operating frequency of the radio transceiver module may influence the height at which the antenna will be placed. This is due to the fact that visual line of sight does not guarantee unobstructed radio wave propagation. The Fresnel zones are the areas surrounding visual line of sight in which radio waves spread out after leaving the antenna. In order to minimize signal loss, the first Fresnel zone should remain at least 60% clear of obstruction. The size of the Fresnel zone is inversely proportional to the frequency of the radio wave, so while a lower frequency wave will attenuate less, and in theory with a longer range, it has a larger Fresnel zone, which makes it more likely to encounter blockages. Figure 3.1 shows a comparison between the sizes of the first Fresnel zone for three popular ISM bands, namely 169MHz, 433MHz, and 915MHz. Equation 3.4 describes the cross sectional area of the first Fresnel zone F, where D is the distance between the two antennas and f is the frequency of the transmission.

$$F = \frac{1}{2} \cdot \sqrt{\frac{c \cdot D}{f}} \tag{3.4}$$

When comparing two radio waves travelling the same distance through the same medium, this simplifies to

$$F_1 \cdot f_1 = F_2 \cdot f_2 \tag{3.5}$$

Table 3.2 shows the increase in cross sectional radius of the first Fresnel zone for the frequencies of 169 MHz, 433MHz and 915MHz. Table 3.3 shows the relative increase in range for transmissions at those frequencies.

Wotherspoon found that the height of the antenna Hardware a bigger influence on the range of the radio transceiver module than the operating frequency,

F_1 \ F_2	169MHz	433MHz	915MHz
169MHz		0.39	0.185
433MHz	2.56		0.473
915MHz	5.41	2.11	

Table 3.2: Relativity of First Fresnel Zone sizes

f_1 \ f_2	169MHz	433MHz	915MHz
169MHz		0.39	0.185
433MHz	2.57		0.474
915MHz	5.41	2.11	

Table 3.3: Relativity of Distances the radio waves will travel

and was able to achieve a range of 5.5km with 98% accuracy using a radio transceiver module operating at 433MHz and a wide band log-periodic dipole array antenna. While for Wotherspoon's network the height of the antenna above ground was restricted, the network proposed for this project has much more flexibility for this. The operating frequency for the radio transceiver module intended for use in the proposed network is chosen to be 433MHz.

Another attribute which is important when considering the performance of a radio transceiver module is the sensitivity of the radio, defined as the minimum power level at which the receiver can clearly receive the transmitted bits.

Table 3.4 shows a comparison between various radio transceiver modules, all of which are capable of performing LoRa modulation and able to use an operating frequency of 433 MHz.

Device	RFM98W	Ra-02	RN2483
Manufacturer	Hope RF	Ai-Thinker	Microchip
Sensitivity [dBm]	-144	-141	-146
Tx power [dBm]	+20	+18	+14
Supply [V]	3.3	3.3	3.3
Tx current [mA]	120	93	40
Rx current [mA]	12.1	12.15	14.2
Sleep mode current [μA]	0.2	1600 (Standby)	2.6
Interface	SPI	SPI	UART

Table 3.4: Comparison of radio transceiver modules

Of the modules compared in Table 3.4, only the RFM98W is readily available, and reasonably priced. Based on the information in Table 3.4 and

the overall performance that Wotherspoon received in his implementation of the RFM96W, the proposed system will make use of the RFM98W radio transceiver module.

3.3 Summary

The purpose of this chapter was to select a radio module for the purposes of parameterising the models developed in Chapers 6 and 7. After investigating a number of LPWAN technologies, it was decided that the radio transceiver module would make use of LoRa technology. The operating frequency was selected as 433 MHz and finally the radio module was chosen as the RFM98W.

The following chapter will discuss the Data Link Layer. Common problems faced by WSNs are investigated before reviewing a number of suitable MAC layer protocols and deciding on one to implement.

Chapter 4

Data Link Layer

This chapter describes the proposed Data Link Layer of the network under discussion. The Data Link Layer is the protocol layer which provides functional and procedural means for the transfer of data between two nodes in a network. The Data Link Layer is split into two parts, namely the Media Access Control and Logical Link Control sublayers.

The protocols that make up the MAC layer of the network, control access to the shared communication medium for transferring data across the network. These protocols resolve data collisions and channel access contention that may occur in the shared medium. The choice of MAC protocol in a network influences both the reliability of the network, as well as the energy consumption of network nodes. The choice of MAC protocol for this IoT network is, therefore, very important to the overall success of the network.

The Logical Link Control sublayer acts as an interface between the network layer and the MAC sublayer, allowing a number of network protocols to operate simultaneously.

For the proposed network, it is important that the chosen MAC protocol be both efficient and reliable in order to conserve as much power as possible. The protocol should also be computationally efficient to further conserve power. The network will be situated in very remote regions when implemented. Therefore, an important aspect in the efficiency of the protocol will be its solving of the "hidden node" and "exposed node" problems, which will be discussed in 4.1

4.1 Challenges

Some common problems that networks face and need to be solved by the implemented MAC protocol, include the Hidden Node Problem, the Exposed Node Problem, and issues surrounding the movement of nodes. For the proposed network, node mobility is not a feature and will, therefore, not be a problem.

4.1.1 The Hidden Node Problem

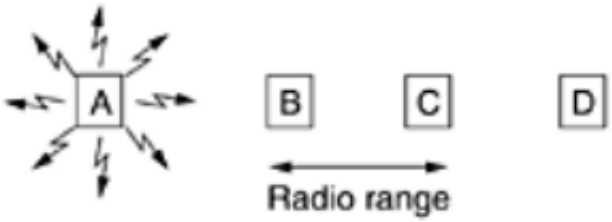


Figure 4.1: Representation of the Hidden Node Problem

Figure 4.1 illustrates an example of the Hidden Node Problem. In the example, Node A is attempting to communicate with Node B, however, while Node A is within range of Node B, it is not within range of Node C. Node C cannot sense that Node A is communicating with Node B because it is out of range, or hidden. Assuming that the medium is free, Node C also attempts to communicate with Node B, interfering with the communication from Node A. This inability to detect competition for access to the communications medium is known as the Hidden Node Problem.

4.1.2 The Exposed Node Problem

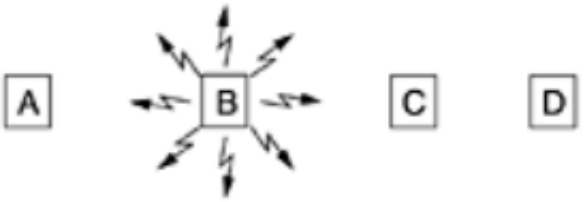


Figure 4.2: Representation of the Exposed Node Problem

In another scenario, as an example referring to Figure 4.2, Node B is transmitting to Node A. Node C can sense this transmission and incorrectly assume that it cannot transmit to Node D when actually the two transmissions would not interfere at either receiving node, this is known as the Exposed Node Problem.

4.2 Choosing a MAC Protocol

There are many ways in which to classify MAC protocols, sections 4.2.1 and 4.2.2 will discuss Contention Based and Collision Free protocols respectively. Contention Based protocols allow multiple users to use the same spectrum, without any pre-coordination, by defining what should happen when two or more transmitters attempt to access the medium at the same time[13]. Collision Free protocols aim to resolve the contention for the medium without any collisions[31]. Sections 4.2.1 and 4.2.2 will discuss a number of MAC protocols before comparing their suitability for the proposed network and deciding on which protocol will be implemented.

4.2.1 Contention Based Protocols

4.2.1.1 Aloha

Aloha is one of the more simple contention based protocols. Nodes transmit whenever they have data to be transmitted. Of course this results in collisions but, because of the feedback property of broadcasting, the sender can detect a destroyed frame by listening to the channel that it is transmitting on[31]. If simultaneous listening and transmitting is not possible, an acknowledgement signal is required from the destination.

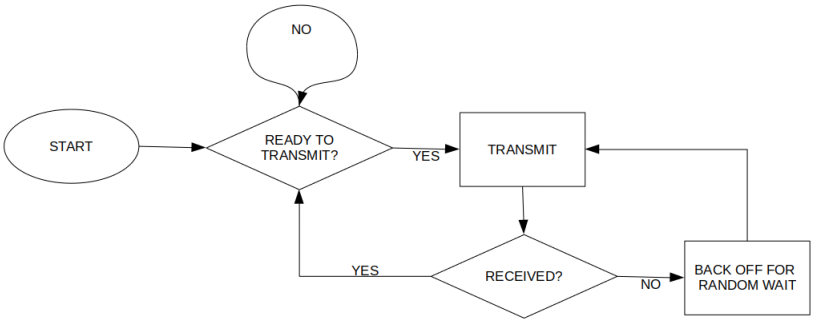


Figure 4.3: Simple visualisation of ALOHA

Aloha is a very simple MAC protocol, and as such it should be relatively simple to implement. Another benefit of the Aloha protocol is that a version of the Aloha protocol is the default MAC protocol for LoRaWAN, meaning that it is already implemented in the LoRaMAC library.

With regards to the suitability for the proposed network, however, the efficiency of the protocol is a concern. Tanenbaum[31] calculates that the peak one can expect from a pure Aloha protocol is 18.4% of the transmitted packets successfully received, which is unacceptable for the proposed network as this will result in most packets requiring to be transmitted at least twice,

further depleting the power supply. Even using Slotted Aloha, an altered implementation of Pure Aloha, the efficiency of the protocol peaks at 36.8% of the transmitted packets successfully received.

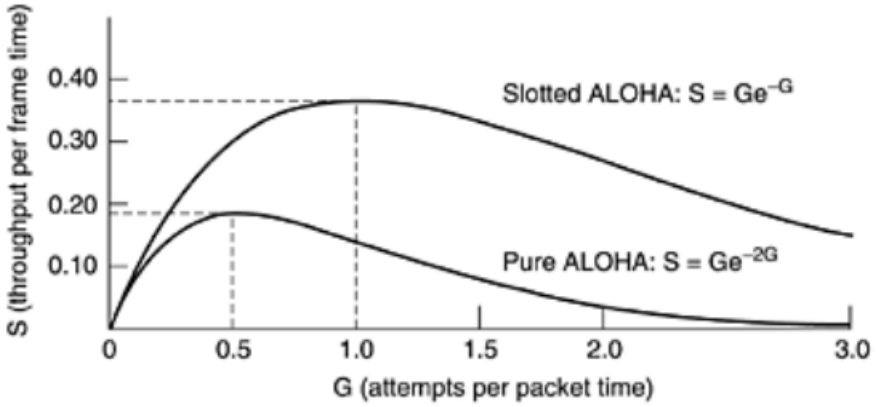


Figure 4.4: ALOHA vs. Slotted ALOHA w.r.t. successful transmission[31]

4.2.1.2 CSMA

CSMA, or Carrier Sense Multiple Access, aims to improve on the success rate of Aloha. When a node has a packet to send, it first listens to the channel to detect any ongoing transmissions. If the channel is busy, the node waits for a random amount of time before attempting to retransmit. Although the node waits for a free channel before transmitting, it is still possible for two nodes to begin transmitting at the exact same time, resulting in a collision. The process of transmitting with CSMA can be seen in Figure 4.5

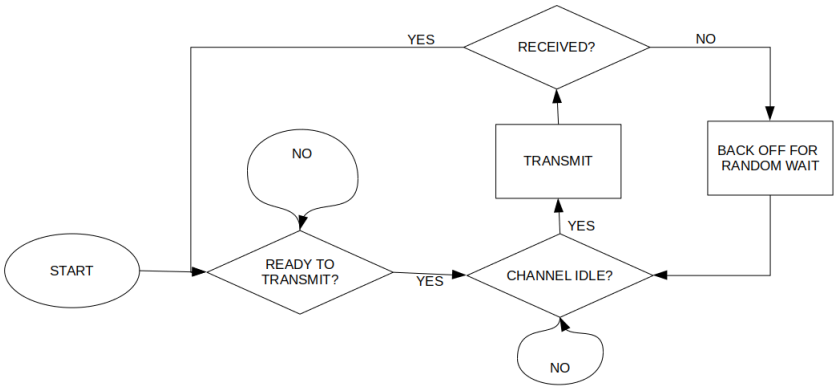


Figure 4.5: Simple visualisation of CSMA

In order to improve on this further, CSMA with Collision Detection (CSMA/CD) can be used. It differs slightly from CSMA in that as soon as a collision is detected, the transmission is aborted, unlike with CSMA where the entire packet

is transmitted and the transmitting node only knows that there was a collision when it doesn't receive an acknowledgement of its transmission. This collision is detected by sensing the medium while transmitting.

Going one step further than CSMA/CD is CSMA with Collision Avoidance (CSMA/CA), which aims to eliminate collisions entirely by initially sending a Request to Send packet (RTS), which includes an estimate of the time the transmission will occupy the medium. After determining that the channel is idle, and only after receiving a Clear to Send (CTS) packet in return will the node begin transmitting. This is able to eliminate many collisions as a receiving node can only send a CTS packet to one transmitting node at a time.

While the CSMA protocols are more complex to implement than Aloha, they are able to achieve a much higher transmission success rate. However, even by making use of CSMA/CA in the proposed network, the Hidden Node Problem is not solved and the protocol may require alteration in order to address this.

4.2.1.3 MACA and MACAW

Multiple Access with Collision Avoidance is a protocol designed for wireless LANs and can best be described by using Figure 4.6 as an example.

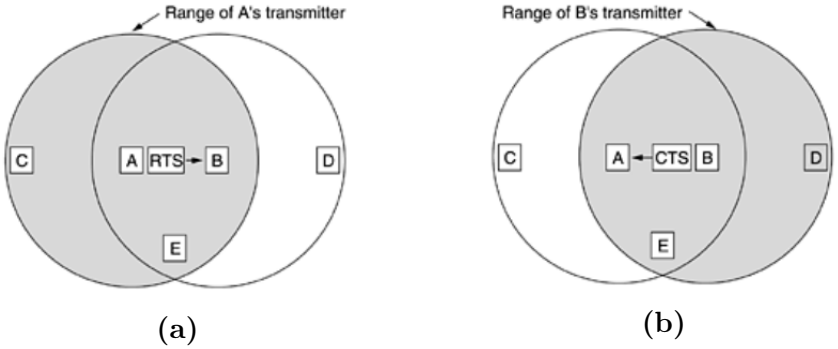


Figure 4.6: Visualisation of the MACA Protocol[31]

In Figure 4.6a, Node A is transmitting a packet to Node B, it begins by sending a RTS packet. Node B responds by sending a CTS back to Node A, depicted in 4.6b. Any node which overhears the RTS from A must not transmit until A receives a CTS, and any node which overhears a CTS from B must not transmit during the upcoming transmission from whichever node is transmitting to B. Nodes C, D, and E respond as follows:

- C overhears the RTS from A, but not the CTS from B, as long as it does not interfere with the CTS it can transmit during the transfer between A and B.

- D does not hear the RTS from A, but overhears the CTS from B to A, D will now not transmit anything until the transfer between A and B is complete
- E overhears both the RTS and the CTS, meaning that it is within range of both nodes. As such it must not transmit until the transfer between A and B is complete.

It is still possible for collisions to occur, however, for example if two nodes were to simultaneously transmit RTS packets to the same node. The response for this collision is for both nodes to wait for a random interval before trying again.

MACA for Wireless (MACAW) aims to improve upon MACA in a number of ways, these include:

- Transmitting an acknowledgement (ACK) frame after each successful data frame is received.
- Implementing some carrier-sense functionality in order to prevent RTS frames from being transmitted simultaneously.
- Running the backoff algorithm for each data stream separately, rather than for each node, to improve the fairness of the algorithm.
- A mechanism was added which allowed nodes to exchange information about congestion.

4.2.2 Collision Free Protocols

In Collision Free Protocols, there is no contention for bandwidth except during the initial phase in which nodes reserve their slot for communication. Once the slots are reserved, no other node can communicate during another node's slot, this results in a fair distribution of bandwidth.

4.2.2.1 TDMA

Time Division Multiple Access (TDMA) is an example of a Collision Free Protocol. The receiving node divides the time in which it receives data into slots and nodes communicating with this node are only able to transmit to it during their assigned slots.

TDMA is, however, more suited to networks which require nodes to be communicating full time, such as mobile telephony networks, so it is not fully suited to the purposes of the proposed network.

4.3 Summary

This chapter began by outlining some common problems faced by WSNs that need to be addressed by the MAC protocol before reviewing a number of potential MAC layer protocols. The MACAW protocol was chosen as the protocol to be implemented in both simulated and theoretical models.

The following chapter will follow a similar format to this one but for the Network Layer. Some important concepts with regards to routing algorithms will be discussed before a number of different routing algorithms are investigated and one is selected for the network.

Chapter 5

Network Layer

This chapter describes the Network Layer of the proposed network. This layer is concerned with getting packets from source to destination. Depending on the topology of the network, this may require packets being relayed by intermediate nodes on the way to the destination. In order for a node to direct packets to the destination, the node must first know the topology of the network, but given that the proposed network will be an ad-hoc network, nodes will not know the topology of the network before they are introduced. The Network Layer is responsible for avoiding congestion in the network as well as discovering routes for packets. However, as the proposed network will operate in the 433MHz ISM band, it will have a duty cycle limited to 10%[37], congestion will not be a major concern.

This chapter will briefly discuss the important characteristics of routing algorithms before reviewing possible routing algorithms and finally selecting and implementing an appropriate routing algorithm.

5.1 Choosing a Routing Algorithm

When a packet arrives at a node, and the node is not the packets final destination, it undergoes a process known as forwarding wherein the next hop on the route is looked up in the routing table. The process of filling in and updating the routing table is handled by the Routing Algorithm.

There are a number of desirable traits for a routing algorithm, these include:

- Correctness
- Simplicity
- Robustness
- Stability
- Fairness

- Optimality

Correctness and Simplicity are obvious, the routes generated by the algorithm should be correct and the algorithm itself should be simple to implement.

In order to be robust, the routing algorithm must be able to deal with changes to the hardware and topology of the network. This is particularly desirable in the case of ad-hoc networks, such as the proposed network, where new nodes may be introduced, or existing nodes may fail or be removed.

A routing protocol can be described as stable if it can maintain a limited number of packets across the network at all times [17]. Some routing protocols never reach equilibrium. A stable routing protocol will reach equilibrium and then remain there. Minimizing packet delay and maximizing the total throughput of the network are both approaches towards optimising a routing protocol. They are often contradictory, however, as maximizing the total throughput of the network means that it will be operating at close to its capacity, which will result in queueing delays. Many networks instead seek to minimize the number of hops on a route, reducing both the delay as well as the bandwidth consumed, which improves the throughput.

Fairness is also a trait which at first may seem obviously desirable, but in fact is often contradictory to optimality. Figure 5.1 shows an example case of the trade off between optimality and fairness. In this example, it is known that there is enough traffic between A and A', B and B', and C and C' to saturate their respective links. To do so, however, would not be fair to the connection between X and X', and so a compromise is needed.

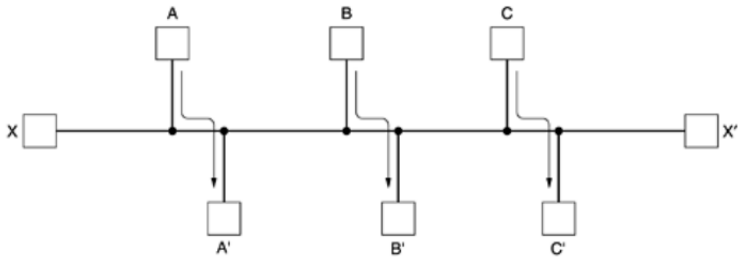


Figure 5.1: Example of the tradeoff between optimality and fairness [31]

5.2 Important Concepts

This section will explain some core concepts which will feature later in this chapter when discussing various routing algorithms.

5.2.1 The Optimality Principle

It can be stated that, in the general case, if node A is on the optimal route from node B to node F, then the optimal route from node A to node F follows the same route. This is known as the optimality principle.

As a result of the optimality principle, as can be seen in Figure 5.2, the set of optimal routes from all other nodes in the subnet, to a destination, which in this case is node B, form a tree rooted at the destination. This is known as a sink tree. It must be noted that the sink tree for any specific node is not necessarily unique. It is the goal of a routing algorithm to discover and implement the sink trees for all routers in a subnet.

(a) A subnet **(b)** The sink tree for node B

Figure 5.2: [31]

5.2.2 Shortest Path Routing

Shortest path routing is widely implemented because of its simplicity and low overhead.

Initially, a graph of the subnet is built, much like in Figure 5.2a. Once the graph is drawn, finding a route between two nodes is as simple as choosing the shortest path. The term "shortest path", however, requires specification, as the shortest possible path is different depending on the distance metric used. There are many possible distance metrics, such as the physical distance, the number of hops, the measured delay, the average traffic, or the cost of communication. Generally speaking, the distance is calculated as a function of a number of these metrics weighted appropriately.

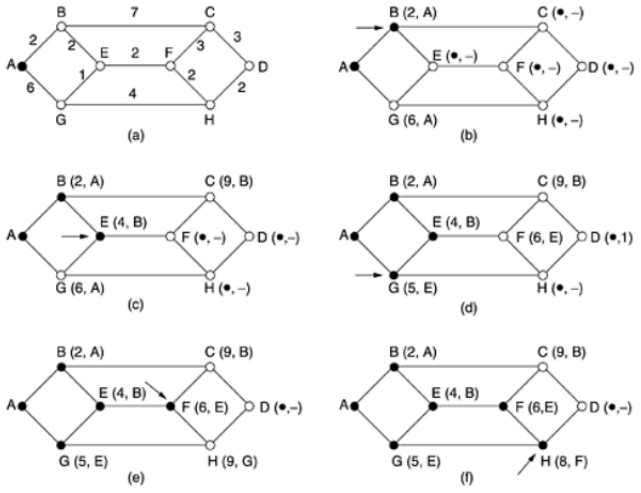


Figure 5.3: Example of the Dijkstra's Algorithm [31]

Figure 5.3 shows an example of Dijkstra's algorithm for computing the shortest distance between two nodes on a graph. All nodes are labelled with their distance from the source node along the shortest known route. At first no paths are known, but as the algorithm executes the labels are updated. Labels are either permanent or tentative, all labels are initially tentative, but once a a label represents the shortest route possible from the source to the node, the label is made permanent.

In Figure 5.3a, Node A is marked as permanent, because it is the source node. Each node adjacent to Node A, the working node, is then inspected and relabelled with the distance to A as well as with the node from which the probe was made. Once all the adjacent nodes are relabelled, all tentatively labelled nodes are inspected and the one with the lowest distance is made permanent, becoming the new working node, this process can be seen in Figure 5.3b .

Consider Figures 5.3c and 5.3d, with Node E as the working node. Again, all adjacent nodes are inspected, if the sum of the label on Node E and the distance from Node E to the Node being inspected, in this case Node G, is less than the label on that node, then this is a shorter path than that node's tentative label, and the node is relabelled. Once all adjacent nodes have been inspected and tentatively relabelled, all tentative labels in the graph are examined, and again, the smallest one is made permanent and becomes the new working node.

This process is repeated until all nodes have permanent labels, the shortest path to each node can then be determined by following the probing nodes in the labels of each node. For example, in 5.3f, to get the shortest path from A to H, one would start at H, reading its label one can see that it was probed by F, which in turn was probed by E, which was probed by B, which was probed by A, making the shortest path A-B-E-F-H.

5.2.3 Flooding

Flooding is a technique in which every packet received by a node, is sent out on every line except the line that it came in on. This will obviously result in an infinite number of duplicate packets unless some measures are taken. One possible measure is to include a hop counter in the header of each packet, decrementing every time it is retransmitted until the counter reaches zero and the packet is no longer retransmitted. Ideally this hop counter will be initialised to the number of hops on the route from the source to the destination. However, this is only applicable if the route to the destination is known. If the route to the destination is unknown, the hop counter can be initialised to the full diameter of the subnet.

Another solution to the problem of infinite duplicates is for each node to keep track of which packets it has flooded, so that no node will flood the same packet twice. This is accomplished by having the source node insert a sequence number into the header of the packet it is flooding. Each node then keeps a list of sequence numbers that it has already seen and flooded from each possible source. If the sequence number on an incoming packet from a specific source is on the relevant list of seen sequence numbers, then the node has already flooded that packet and does not need to do so again, and so discards it.

Selective flooding is a more practical implementation, in which, instead of sending an incoming packet out on every outgoing line, the packet is only sent out on lines that are in the general direction of the destination. Unless the network topology is very unorthodox, it doesn't make sense for a packet going to a destination in the West, for example, to be sent on Eastern outgoing lines.

In the case of wireless networks, any outgoing transmission can be received by any receiver within radio range. This is technically flooding, and while flooding in most cases is impractical, it is often used to some degree in wireless routing algorithms.

5.3 Dynamic Routing Algorithms

Shortest path routing and flooding are both static routing algorithms. This means that the routes are defined before the network is operating, changes to the current network load and topology are not taken into account. This approach is not suitable for the proposed network because in order for the network to be a robust, ad-hoc network, it must be able to adjust to changes in the network load and topology and still maintain operation.

Dynamic routing algorithms are more suited to the needs of the proposed network in this regard. This section will discuss two popular approaches to dynamic routing, namely Distance Vector Routing and Link State Routing.

5.3.1 Distance Vector Routing

Distance Vector Routing algorithms operate by having each router maintain a routing table with the best known distance to each possible destination in the subnet, as well as how to get there. The tables are updated by routers exchanging information with their neighbours. Each table is indexed by, and contains an entry for, each router in the subnet, where each entry shows which outgoing line to use to get to that router as well as how far away that router is, using some distance metric.

5.3.1.1 Count-to-Infinity Problem

One serious drawback to the original Distance Vector Routing algorithm was its tendency to take a long time to converge. The algorithm reacts very well to good news, i.e. a shorter path being discovered, but very slowly to bad news, i.e. a broken link.

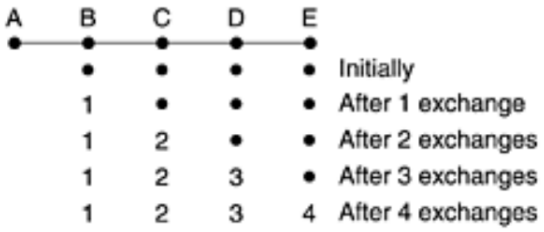


Figure 5.4: Example of the good news converging quickly [31]

Consider the example in Figure 5.4. Initially, the link between A and B is broken, and so all the nodes have their distance metric to node A set to infinity. When the link between A and B is restored and a vector exchange takes place between all the nodes, node B now knows that it is only one hop away from node A. This information is steadily spread through the subnet each time there is a vector exchange.

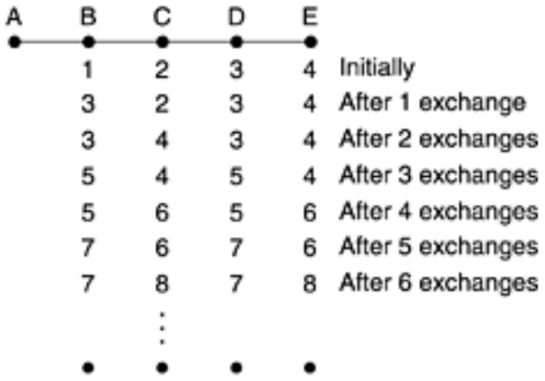


Figure 5.5: Example of the bad news converging slowly [31]

Now consider the example in Figure 5.5. Initially, the link between Nodes A and B is intact, and all the nodes have their distance metrics to node A set appropriately. When the link between A and B is broken and a vector exchange takes place, Node B sees that its direct link to A is infinite, but it also knows that Node C has a route to A that only entails two hops. It does not know that Node C's path to Node A is via Node B itself, and so sets its distance metric to Node A to the value of Node C's distance metric to Node A plus one. At the next vector exchange Node C sees that its path to Node A through Node B has been altered, and sets its distance metric to Node A to the value of Node B's distance metric to Node A plus one. This continues amongst all the nodes until they reach infinity, or some very high value. This is known as the Count-to-Infinity Problem.

Two Distance Vector algorithms that address the Count-to-Infinity problem are Destination Sequenced Distance Vector Routing and Ad-Hoc On-Demand Distance Vector Routing.

5.3.1.2 Destination Sequenced Distance Vector Routing

Destination Sequenced Distance Vector, or DSDV, Routing is a proactive routing protocol. This means that every node in the subnet maintains a routing table that is comprehensive, it contains routes to every other node in the subnet. These routing tables are updated periodically by nodes advertising their own routing tables to their neighbours, who then update their own routing tables and advertising themselves.

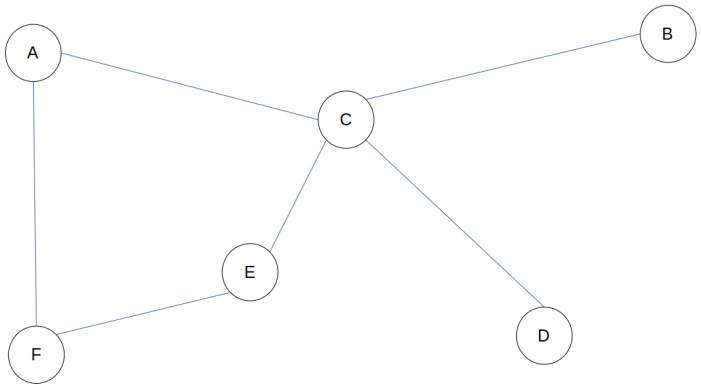


Figure 5.6: Example of a Subnet using DSDV

Consider the subnet in Figure 5.6 as an example. The routing table for Node B can be seen in Table 5.1. The first three columns are relatively simple, they represent the possible final destination final destination for a packet, the immediate destination for the packet where it will be routed, and the distance metric for how far away the final destination may be, in this case Hops.

The next column represents the Sequence Number in the format **SNNND**, where S indicates that this is a sequence number, NNN is a timestamp defining when this route was updated, and D is the destination node. From the table, it can be seen that the route to Node E was most recently updated, and that none of the links along any of the routes are broken [25], as all digits in the unit space of the sequence numbers are even.

The Install column shows when a Destination became known to this Node, in the format **TNNNS**, where T indicates that is an install number, NNN is a timestamp defining when this destination became known to this node, and S is the source. From the example routing table it can be seen that all of the destinations became available to Node B at more or less the same time, most likely due to Node B being the most recently added Node to the subnet.

The final column, Stable Data, contains a pointer pointing to a table describing the stability of the route [4].

When a node receives an advertised routing table and compares it with its own, then if the advertised route has a higher, i.e. more recent, sequence number, it will update its routing table to match the advertised route. Or else it will discard the advertised route.

If the sequence numbers of the advertised route and the nodes own routing table entry are the same, i.e. they are based on the same information, but follow different routes to the destination, then the route with the lowest distance metric is chosen.

Destination	Next Hop	Metric	Sequence Number	Install	Stable Data
A	C	2	S406A	T001B	Ptr1_A
C	C	1	S128C	T001B	Ptr1_C
D	C	2	S564D	T001B	Ptr1_D
E	C	2	S710E	T002B	Ptr1_E
F	C	3	S050F	T001B	Ptr1_F

Table 5.1: Example of an Internal Routing Table for Node B in Figure 5.6

Table 5.2 shows the routing table that Node B advertises to its neighbours. Note that it does not advertise its entire routing table, only the fields relevant to its neighbours, eg. the destination, how quickly a packet can get to the destination from this node, and how recently this information was updated.

Destination	Metric	Sequence Number
A	2	S406A
C	1	S128C
D	2	S564D
E	2	S710E
F	3	S050F

Table 5.2: Example of an Advertised Routing Table for Node B in Figure 5.6

If, for example, the link between nodes A and C were to break down, the topology of the subnet would change to that of Figure 5.7. In this scenario, Node C would detect the broken link between itself and A, and set the distance metric between itself and A to infinity, or some very high value. It would then advertise its updated routing table to its neighbours. Node E, seeing this very high value would understand that the link is broken, and also know that it has a link to node A through node F. Node E then updates its own routing table to reflect the broken link and advertises this information to its neighbours. The information regarding the broken link propagates through the network. The updated route from C to A is now C-E-F-A.

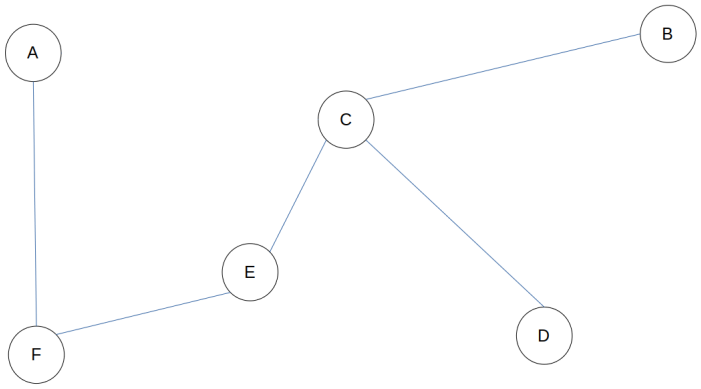


Figure 5.7: Example of a Subnet with the link between A and C broken

An advantage of DSDV is that, as each node contains a comprehensive routing table, there is little to no delay when trying to find a path to the destination. An important disadvantage, however, is the low scalability of the protocol. Because each node contains a comprehensive routing table, if there are more nodes in the network, there will be more storage required for each node, as well as more bandwidth used when updating the routing tables.

5.3.1.3 Ad-Hoc On-Demand Distance Vector Routing

Ad-Hoc On-Demand Distance Vector, or AODV, Routing is a reactive routing protocol, meaning that each node is capable of finding a route to any other node on demand.

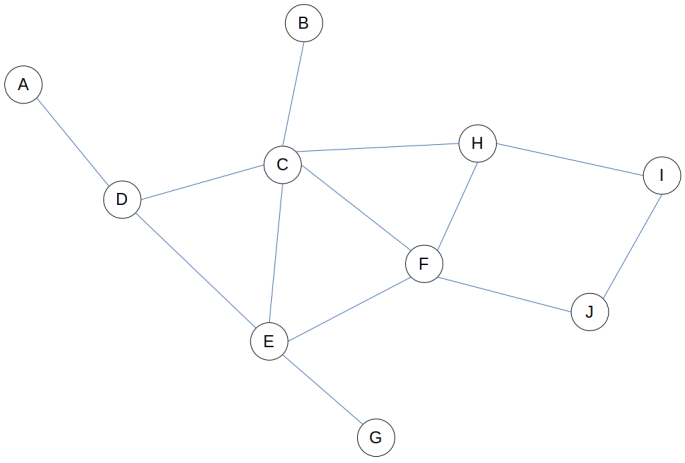


Figure 5.8: Example of a Subnet using AODV

AODV was designed with the following goals in mind [2]:

- To minimize control and processing overhead
- Allow for multi-hop routing
- Utilize dynamic topology maintenance
- Prevent the formation of loops

Consider the subnet in Figure 5.6 as an example. Node A has a packet that it needs to send to Node J. Neither Node A, nor any other node in the subnet have any knowledge of a route from Node A to Node J. Node A floods the subnet with a Route-Request packet, labelled as RREQ in Figure 5.7. When a node receives a RREQ signal, it can forward the request, or if the receiving node is also the destination, it can respond with a Route-Reply signal, labelled as RREP. If a RREQ is received more than once, the duplicates are discarded, this is both because the latter RREQ has taken longer to arrive at this node and thus is clearly not the optimal route, as well as to prevent the formation of loops, which is known to be a problem when flooding.

When a RREQ signal is broadcast, it accumulates the addresses of the nodes it has been broadcasted by, so that when the destination node responds with a RREP signal, it is addressed to the node that initiated the routing process and will follow the same path back to that node.

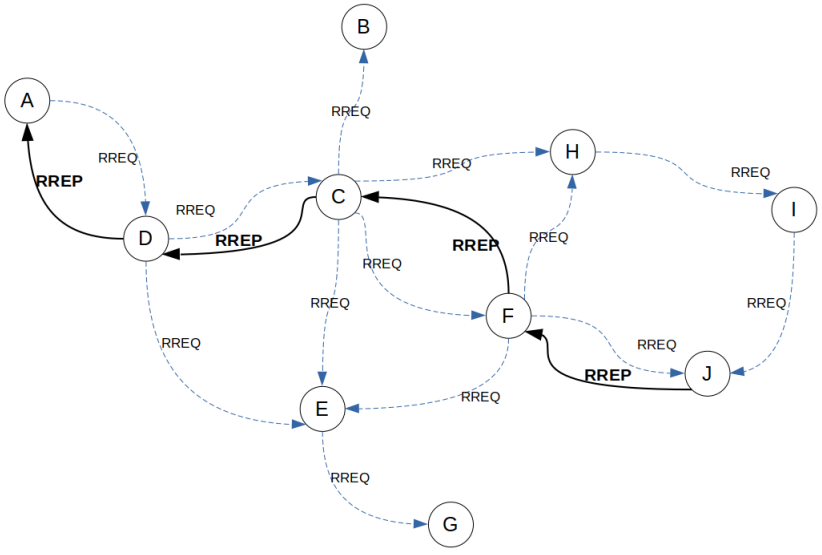


Figure 5.9: Example of a AODV applied to Figure 5.8

If a broken link is detected along a route, the node will send a RREP signal back to the source node with the distance metric to the destination set

to infinity, or a very high value. All nodes along this route will receive the RREP and delete the route from their routing tables. When the source node receives the RREP, it will delete the route from its routing table and begin the routing process again by flooding the subnet with a RREQ signal.

An important advantage of AODV is the much lower bandwidth and storage requirements than a protocol like DSDV, as the routing table for each node is not comprehensive, but rather only contains routes that are used and relevant. One disadvantage of AODV is can be a delay could be caused when a node is transmitting to a destination for the first time and must initiate the routing process.

5.3.2 Link State Routing

Distance Vector Routing was used in the first wide area packet switching network, ARPANET, until 1979 [31]. There were two reasons for its retirement. Firstly, Distance Vector Routing made use of queue length as its distance metric, disregarding bandwidth when selecting routes. This was not an issue at first as all lines had a bandwidth of 56 kbps, but once some lines began upgrading to 230 kbps or 1.544 Mbps disregarding the bandwidths of lines became a problem. Changing the distance metric was an option, but because of the second problem with Distance Vector Routing at the time, the Count-to-Infinity Problem, it was replaced with a completely new routing algorithm, Link State Routing.

There are five main steps that each node must follow in the Link State Routing algorithm, namely:

- Discovering its neighbours
- Measuring the distance metric to each neighbour
- Constructing a packet containing everything it has just learned
- Sending this packet to the other nodes in the subnet
- Using the information it receives from the other nodes in the subnet to compute the shortest path to each destination.

Discovering Neighbours

When a router joins the subnet, its first task is to construct a "HELLO" packet to send to its neighbours, introducing itself. The neighbouring routers then reply to their new neighbour telling it their addresses.

Measuring the distance to each neighbour

Depending on the distance metric used this process may vary. If the metric is simply the number of hops, then all the neighbours are obviously only one hop away. If the metric is, or incorporates, the delay between a packet being transmitted by the source and received by the destination, then the source node can construct an "ECHO" packet to send to each neighbour, which the neighbour must then send back immediately. Measuring the time that this takes and dividing it by two gives the source node a reasonable estimate of the delay between the two nodes.

Constructing the Link State Packets

Once each node has collected the necessary information from its neighbours, it builds a packet containing this information. The packet begins with the name or address of the sending node, followed by a sequence number and age, and then the neighbours and the information about them. An example of a subnet and the Link State Packets for each node in the subnet can be seen in Figure 5.10.

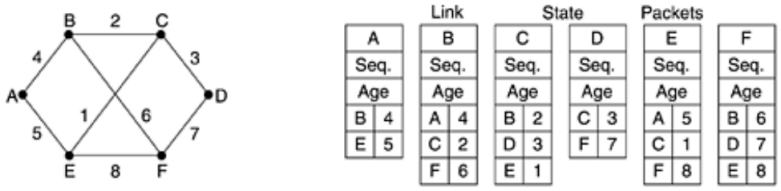


Figure 5.10: Example of the Link State Packets for a subnet [31]

Distribution of Link State Packets

The reliable distribution of the Link State Packets is the most challenging aspect of the algorithm. As the packets are distributed and installed, routers closest to the sources will receive the new routing information first and subsequently change their routes. This means that different routers may operating with different versions of the topology, which can lead to inconsistencies and loops.

The packets are distributed by flooding, each packet contains a sequence number to avoid loops. When a new packet comes in, its sequence number is checked, if it is new, it is forwarded on all lines except the one it came in on. If it is older, then it is discarded. There are three potential problems with the use of these sequence numbers as described. Firstly, If sequence numbers wrap-around then using them in the first place is pointless. The solution to this is just to use a 32-bit number for the sequence number, which will never feasibly wrap around. The second problem is that of a source losing track of its sequence number if it crashes. It would then have to start counting from

zero again. The third problem is sequence number corruption, if a sequence number is received as 65540 instead of 4 for example.

A solution to all of these problems is to include an Age field in each packet which will decrement once every second. Once the age hits zero, the packet is discarded from that router.

Computing Routes

Once each router has a full set of Link State packets, it is able to construct the full subnet graph. Once the subnet graph is constructed, any shortest path computation algorithm, such as Dijkstra's algorithm described in Section 5.2.2, can be used to calculate the routes.

5.4 Summary

This chapter presented an overview of some important concepts of routing protocols in WSNs before several possible protocols were described and AODV was chosen as the routing protocol to implement in the models presented in Chapters 6 and 7.

The following chapter will describe the importance of simulating a network in the network design process. Thereafter some aspects of Discrete Event Simulation will be discussed before DESMO-J, the tool used to simulate the network, is further explained. The implementation of the proposed network in DESMO-J is then outlined before the simulation results are presented.

Chapter 6

Discrete Event Simulation of an Ad-Hoc Network

In order to explore the potential final behaviour of the communication system being developed, it will be helpful if it could be modelled and simulated. There are a number of advantages to modelling and simulating the system before implementing it physically. By modelling the system, one can gain a greater understanding of the system's potential behaviour, because of the need to consider all aspects of the system when implementing it in code. In this way the logic of the system is clarified and inconsistencies can be more easily identified. Modelling and simulation can also be very useful when physical implementation is impractical or too expensive. Data obtained from simulations can support and improve the quality of the choices made regarding the configuration of the system, such as comparing the performance of various MAC and network layer protocols in order to decide which is the best fit for the system.

While modelling and simulation would clearly be have advantages, there are a number of pitfalls which should be taken note of to ensure that the process is worthwhile. Since modelling a system involves a certain degree of simplification, it can result in instances where there is a large discrepancy between the model and the system that it represents. It can misinterpret relationships or overlook relevant aspects of the system. Therefore, assumptions regarding the relationships in the system must be validated carefully [10]. It should always be remembered that results obtained from a simulation will not necessarily correspond perfectly with the performance of the system in reality, and that overestimating the accuracy of the model can lead to the system underperforming.

When modelling a system it is important to note which attributes of the simulation are being tracked and used to analyse the performance of the simulated system. For the simulated network, the most important information will be the number of retransmissions required for each transmission.

The network will be modelled using the Discrete Events Simulation method for modelling real world systems that can be broken down into a series of

seperate processes. Within a DES each event occurs within a logical process at a specific time, the output of each event can result in a new event, to be processed at a future specified point on the logical timeline[7]. "DES is a flexible modeling method characterized by the ability to represent complex behavior within and interactions between individuals, populations,and their environments" - Karnon *et al.*

A defining trait of DES is representation of system states and their change over time [10]. The majority of DE simulations describe stochastic experiments, making use of various distributions to generate random numbers. Section 6.1 describes some basic concepts and key components of a DES. DES, as described in this introduction, is perfectly suited to the modelling and simulation of a network, the process of which is further detailed in this chapter.

6.1 Basic Concepts of DES

Entities

Entities are objects that model a system's real components [10], are able to interact with each other and actively move through simulation time. Entities have attributes, experience events, use resources and occupy queues [20].

Attributes

Attributes are the characteristics of a specific entity that carry information and may influence how it responds to changes in the state of the simulation. The values of attributes can be modified at any time in the simulation and analysed outside of the simulation.

Events

Events are occurrences that change the states of entities at specific points in simulation time. When an event occurs, the simulation clock is updated accordingly, and any necessary transformations to the system state are made. These state changes generally lead to other events being triggered. Once an event is complete, the next event must be attended to. To manage this, the simulation controller keeps all candidate events in a data structure known as an event list. The event list is iterated over until there are no more scheduled events.

Resources

A resource is an object that provides a service to an entity [20]. Representing resources allows the DES to capture spatial factors.

Queues

If there are not enough resources to service all the entities that require the resource, then these entities must form a queue, and use the resources as they become available. Queues can be configured in a number of ways, such as having a maximum capacity or having different approaches to calling entities from the queue (first-in-first-out, first-in-last-out, etc.).

Time

A model represents system structure and behaviour, which are static and dynamic respectively. A dynamic model must consider all states and how these states change over time. Model time is the time that is modelled within the simulation, as opposed to real time which passes while the simulation is executing. Model time is therefore independent of the time that the simulation takes to execute and allows results to be interpreted in a pseudo real-time environment. The discrete handling of time allows for efficient advancement from one event to the next, as unnecessary computations are avoided.

6.2 Dominant Approaches to DES

6.2.1 Process Oriented

The process oriented approach to DES groups all activities "owned" by an entity into a process. These processes include all the operations that the entity will carry out in its lifetime [15]. The system, therefore, consists of a set of interacting processes, and model time passes during the active phases of the processes. Conceptually, the lifecycles of these entities are executed in parallel, and control is passed back to the scheduler when the model time changes.

A process is able to:

- Modify entity attributes
- Generate new entities
- Activate and deactivate processes
- Change or cancel the active phases of processes
- Deactivate itself
- Terminate itself and other processes

The process oriented approach is the most popular approach to DES. This approach tends to produce more modular code and is often easier to write and read [23].

6.2.2 Event Oriented

The event oriented approach to DES focuses on the set of all events for all relevant entities occurring at a specific time. Instead of grouping chains of events into process descriptions, events are clustered based on the time at which they occur. Unlike the process oriented approach, where descriptions of structure and behaviour are intertwined, there is a clear separation between the specifications of the structure and the behaviour of a system.

An event oriented model consists of static and dynamic components. The static components are the entities which model classes of objects as well as permanent individual objects. The dynamic components are all model events and temporary objects whose creation and destruction is triggered by the events.

Events can:

- Change attribute values
- Create and destroy temporary objects
- Add new events to, and deleting events from, the model's events list

6.3 DESMO-J

Discrete-Event Simulation Modelling in Java, or DESMO-J, is an object-oriented simulation framework for DES. It offers a collection of pre-packaged *black box* model components, modification of which is limited to parameter changes. *White box* components are more abstract and require adaptation and customization to fit the needs of the simulation.

DESMO-J's black box components include:

- A DES's infrastructure, components such as the events list, scheduler, and simulation clock, as a part of the Model class.
- Means for recording the results of the DES, as a part of the Experiment class.
- Constructs for modelling queues and random processes.
- Constructs for collecting statistical data.

DESMO-J's white box features offer generic objects with active behaviour, i.e. the Model, Entity, Event or SimProcess classes, which must be subclassed as fits the model.

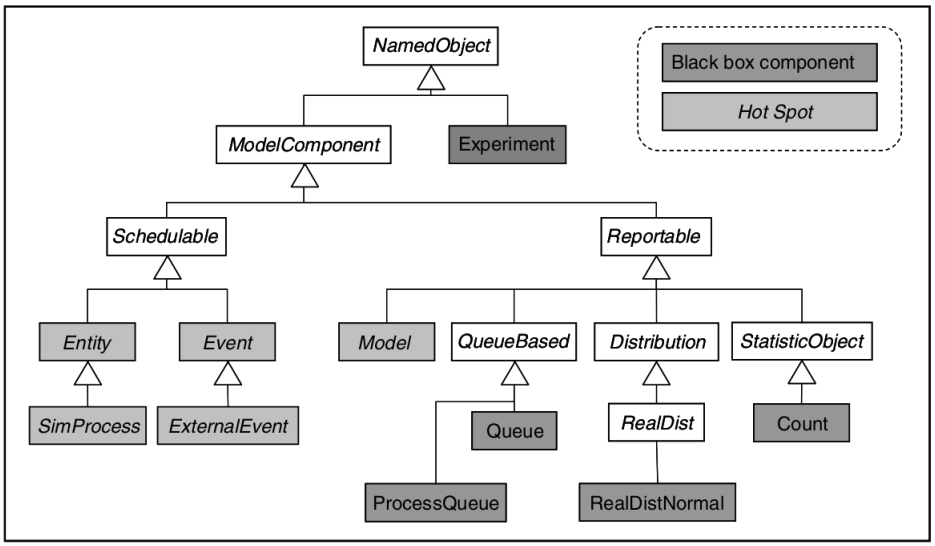


Figure 6.1: DESMO-J class heirarchy[10]

6.3.1 The Experiment Class

In DESMO-J, the Experiment class acts as an interface to the model's infrastructure components: the events list, the scheduler, and the simulation clock. This means that those using DESMO-J only need to understand the Experiment class's interface and not directly interact with any of the aforementioned infrastructure components.

6.3.2 Model Components

As shown in Figure 6.1, all objects outside of the Experiment class fall either under the Schedulable or Reportable classes.

6.3.2.1 Schedulable Classes

These include all objects that are controlled by the scheduler, such as Events, Entities and SimProcesses. Methods available to objects extending the Schedulable class include:

- schedule(Entity who, SimTime when) - schedules an event, involving the specified entity, to occur, at the specified model time.
- scheduleAfter(Schedulable after, Entity who) - schedules an event, involving the specified entity, to occur after an event involving the specified Schedulable object.
- scheduleBefore(Schedulable before, Entity who) - schedules an event, involving the specified entity, to occur before an event involving the specified Schedulable object.

6.3.2.2 Reportable

Includes all objects which are required to publish their information. The Reportable class provides the necessary functions for displaying the collected information in reports. In Figure 6.1 it can be seen that this includes the Model class itself, all queues and distributions, as well as data collectors which make up the StatisticObject class. These data collectors include:

- Count - provides a counter to record the number of occurrences of events.
- Tally - provides the ability to record the mean and standard deviation of a series of values.
- Accumulates - like Tally, provides the ability to record the mean and standard deviation of a series of values, but each value is weighed by the interval during which the value has endured, providing time-weighted statistics.
- Histograms - ranks the measured values into a predefined class and draws a corresponding diagram.
- Regressions - allows for regressional analysis of two collected values. reports the number of observations, means, regression and correlation coefficients, standard deviations and interceptions.
- Time Series - provides functionality for logging a time series into a text file.

6.4 Model Description

By simplifying the model construction process, it can be split into three main tasks, namely:

- Choosing the *black box* components, i.e. data collectors, statistical distributions and queues.
- Defining the lifecycles of the *white box* components by customising the relevant classes, such as the Entity or Event classes.
- Initialising and parameterising model components as a part of the Model class.

6.4.1 Distributions, Queues, and Data Collectors

6.4.1.1 arrivalStream

In order to comply with radio transmission standards, the duty cycle of the nodes cannot exceed 10%. The arrivalStream distribution, therefore, can be sampled to obtain the time between transmission generation.

6.4.1.2 transmissionTimeStream

The effective time on air for the transmission is sampled from this distribution, based on the size of the transmission. This value will also be used in the RTS and CTS transmissions for certain protocols.

6.4.1.3 tmQueue

This queue models of the storage of each NodeEntity. Transmissions are stored in this queue while they wait to be transmitted to their next destination.

6.4.1.4 tmArrivalQueue

This queue models the receiver of the NodeEntity. Any transmission attempting to reach the NodeEntity enters this queue. This also makes it simple to keep track of transmissions colliding and failing.

6.4.2 Entities

This model adopts the Process Oriented approach to model construction. As such, all entities described in this section have their activities grouped into a lifecycle method.

6.4.2.1 NodeEntity

The NodeEntity extends the SimProcess class and makes use of three enums to represent the various states of the NodeEntity. These enums are:

- **State**, which has three possible values - RF_IDLE, RF_TX_RUNNING, and RF_RX_RUNNING
- **TX**, which has five possible values - RTS, CTS, TM, ACK, and NONE
- **RX**, which has five possible values - RTS, CTS, TM, ACK, and NONE

The variables used by the NodeEntity are described in Table 6.1. When the NodeEntity is created and activated, its **node** and **destinationNode** are initialised to the appropriate values and its **currentState**, **currentTX** and **currentRX** are set to **RF_IDLE**, **NONE**, and **NONE** respectively.

Variable Name	Type	Description
currentState	State	The state of the NodeEntity
currentTX	TX	What the node is transmitting
currentRX	RX	What the node is receiving
n	int	Counter used for Binary Exponential Backoff
K	int	Upper boundary for the random number generation in Binary Exponential Backoff
rand	Random	Random object used to generate random number for Binary Exponential Backoff
backoff	int	Multiplier for wait period for Binary Exponential Backoff
node	int	The number of this NodeEntity
destinationNode	int	The number of the NodeEntity that this NodeEntity will transmit to
ctsNode	int	The number of the NodeEntity that is transmitting to this NodeEntity
quiet	boolean	Boolean describing whether this NodeEntity should be silent or not
wait	TimeInstant	Time until which the NodeEntity must be silent
ackWait	TimeSpan	How long the NodeEntity should wait after sending a RTS, or TM, for a CTS, or ACK

Table 6.1: Description of variables in the NodeEntity object

The NodeEntity's lifecycle consists of it entering an infinite loop, in which it operation is determined by the values of **currentState**, **currentTX** and **currentRX**, as described in Subsections 6.4.2.1.1 to 6.4.2.1.3.

6.4.2.1.1 RF_IDLE

When the node is activated in the RF_IDLE state, it inspects the values of the RX and TX variables to see whether or not the node has something to receive or send. If the TX value is not NONE and the RX value is NONE, this means that the node has a transmission that it can send.

Certain MAC protocols require nodes to be "quiet" for a period of time so as to prevent collisions. A flow diagram for such a protocol can be seen in Figure 6.2. The node must first confirm that it does not need to be quiet before beginning its sending procedure. If the node is required to be quiet, it will wait for the appropriate amount of time before re-entering the loop.

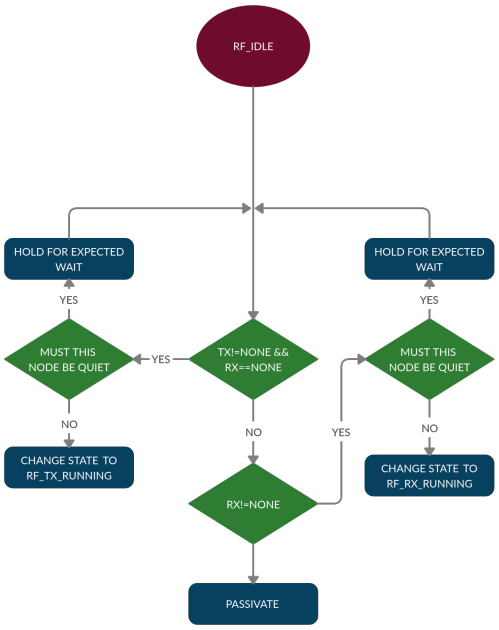


Figure 6.2: RF_IDLE state for protocols which require RTS and CTS signals.

If RX is not NONE, it means that the node is receiving a transmission. Similarly to when the node is transmitting, the receiving node must first check if it must be quiet. If the node must be quiet, it cannot reply to a RTS transmission with a CTS transmission. This, of course, only applies to protocols that make use of RTS and CTS transmissions.

At least one of either RX or TX must be NONE at any given time. If both RX and TX are NONE, then the node has nothing to send or receive and it goes to "sleep".

6.4.2.1.2 RF_TX_RUNNING

When the node is activated in the RF_TX_RUNNING state it operates based on the value of the TX variable, which may, in this case, not be NONE.

If TX is set to RTS, the node creates and activates a RTSEntity. Activating this entity also starts transmitting it. The node then waits for an appropriate amount of time to receive a CTS. When the node is activated again, it is either because the wait period has expired, meaning that the RTS transmission has failed, in which case the node waits for a period determined by Binary Exponential Backoff before re-entering the loop and attempting to send the RTS again. Or, it is because a CTSEntity has arrived at this node, meaning that the RTS was successful, in which case the TX variable is set to NONE and the state of the node is set to TF_RX_RUNNING.

If TX is set to CTS, the node creates and activates a CTSEntity, sets TX to NONE and sets the state of the node to RF_IDLE. Similarly to the case

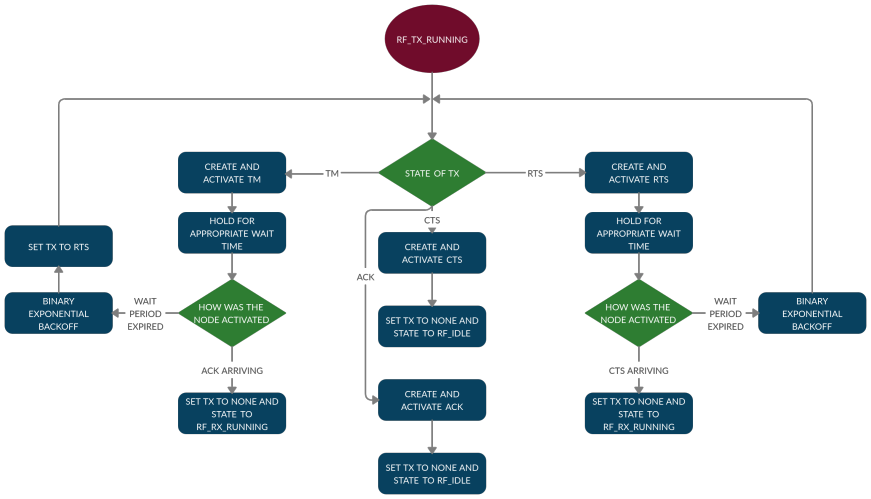


Figure 6.3: RF_TX_RUNNING for protocols which require RTS and CTS signals.

when TX is set to ACK, the node creates and activates an ACKEntity before setting TX to NONE and the state of the node to RD_IDLE.

If TX is set to TM, as with RTS, a TMEntity is created and activated before the node holds for an appropriate amount of time to receive an ACK. If the node is activated by an ACKEntity arriving then TX is set to NONE and the state of the node is set to RF_RX_RUNNING. If the transmission was unsuccessful and the node was activated by the time expiring then the node waits for a period determined by Binary Exponential Backoff, it then sets TX to RTS before re-entering the loop.

6.4.2.1.3 RF_RX_RUNNING

When the node is activated in the RF_RX_RUNNING state it operates on the value of the RX variable. As with the TX variable, when RX is set to NONE, the node should not be in this state and so it is not considered.

If RX is set to RTS the node sets RX to none, TX to CTS and the state to RF_TX_RUNNING. Similarly if RX is set to CTS then the node sets RX to NONE, TX to TM, and the state to RF_TX_RUNNING.

If RX is set to TM, the node must take the TMEntity from its tmArrivalQueue and insert it into its tmQueue. The node then sets the RX to NONE, the TX to ACK, and the state to RF_TX_RUNNING.

If RX is set to ACK, the node must remove the TMEntity from its tmQueue and reset its counter for Binary Exponential Backoff before it sets its RX to NONE and state to RF_IDLE.

6.4.2.2 TMEntity

The TMEntity extends the SimProcess class and makes use of one enum to describe the state of the transmission, **TX_State**, which has possible values

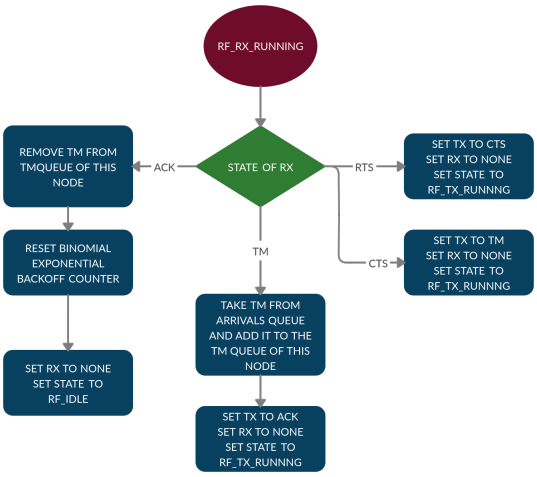


Figure 6.4: RF_RX_RUNNING for protocols which require RTS and CTS signals.

Variable Name	Type	Description
currentNode	int	The number of the NodeEntity that the TMEntity is currently at
destinationNode	int	The number of the NodeEntity that the TMEntity is being transmitted to
route	int[ ]	Array containing the number of each NodeEntity that the TMEntity must travel to on its route
routeIndex	int	Counter to keep track of the TMEntity along its route
currentTX	TX_State	Current state of the TMEntity

Table 6.2: Description of variables in TMEntity

of **DESTROYED** or **SUCCESS**.

The variables used in the TMEntity are described in Table 6.2. When the TMEntity is created and activated, its **route** is initialised and its **routeIndex** is set to 0. The **currentNode** and **destinationNode** are set to the values of the first second entries in the **route** respectively.

The lifecycle of the TMEntity involves it looping over the instructions for transmitting until it has reached the final NodeEntity of the route.

Once the TMEntity enters the loop, it obtains a list of all the NodeEntities within transmission range of its currentNode from the model. For each node in range that is not the destinationNode, the TMEntity sets the currentTX of all other TMEntities in the tmArrivalQueue of that NodeEntity to DESTROYED. For the destinationNode, the TMEntity inserts itself into the tmArrivalQueue of the destinationNode, it then determines the number of TMEntities in this tmArrivalQueue, if there is more than one, the TMEntity sets the currentTX

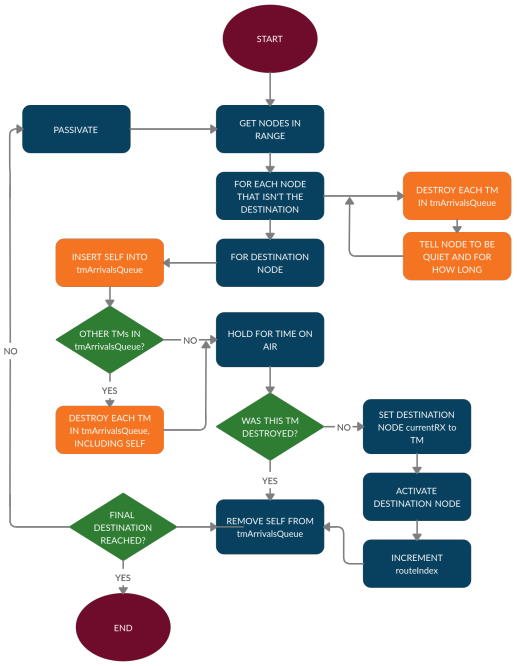


Figure 6.5: TMEntity lifecycle

of all TMEntities in the queue, including itself, to DESTROYED.

The TMEntity then holds for a period of time sampled from the transmissionTimeStream distribution. If the value of currentTX is SUCCESS, then the TMEntity sets the value of currentRX at the destinationNode to TM and activates the destinationNode before incrementing the routeIndex and passivating itself. The TMEntity repeats this process until the routeIndex is equal to the length of the route.

6.4.2.2.1 TMEntity subclasses

There are three entities that inherit attributes from the TMEntity class namely the RTSEntity, CTSEntity, and ACKEntity classes. The lifecycles of these entities are similar to that of the TMEntity class, but each with notable differences. The major difference between these entities and the TMEntity class is that they do not loop. Where the TMEntity loops until it reaches its final destination, these subclasses simply travel from their sending NodeEntity to the immediate next NodeEntity on the TMEntity's route, in other words, new instances of these transmissions must be generated for each hop in the route.

Aside from the looping, the only differences between the subclasses and the parent class are the changes they make to the NodeEntity's states upon a successful transmission. Instead of setting the currentRX of the destinationNode to TM, it is set to RTS, CTS, or ACK, respectively.

6.4.2.3 TMGeneratorEntity

The creation and initial placement of TMEntities is handled by TMGeneratorEntities. There is a dedicated TMGeneratorEntity for each NodeEntity, as such each TMGeneratorEntity has a variable "start" which corresponds to the number of the NodeEntity in which it places the generated TMEntity. When the TMGeneratorEntity is created, the "start" variable is initialised and the entity begins its lifecycle. Like the NodeEntity, the TMGeneratorEntity loops infinitely in its lifecycle. The first action in the loop is to sample a value from the arrivalStream distribution. This value sets the time the TMGeneratorEntity must wait for before generating a new TMEntity in order to comply with the LoRa dutycycle. After waiting for this period, the TMGeneratorEntity inspects the tmQueue of the NodeEntity. If there is not already a TMEntity in the queue, the TMGeneratorEntity creates a new TMEntity, places it in the queue, and activates the NodeEntity.

6.4.3 Parameterisation

Parameters for the simulation include:

- The number of nodes in the network, as well as which of these nodes are gateways.
- The time on air for each transmission
- The visibility of each node, i.e. which nodes can see, and be seen by, other nodes.
- The frequency with which new transmissions are generated, which is determined by how often data is collected by sensors to be sent to a gateway.

With the exception of node visibility, the parameters for each simulation were identical and are described as follows.

6.4.3.1 Number of nodes

The number of nodes in each simulation was chosen to be thirteen nodes and one gateway per topology. This could obviously be a different number, but it was felt that this initial choice was not unreasonable.

6.4.3.2 Time on air

It is possible to calculate the time on air for any specific payload length using a program made available by Semtech.

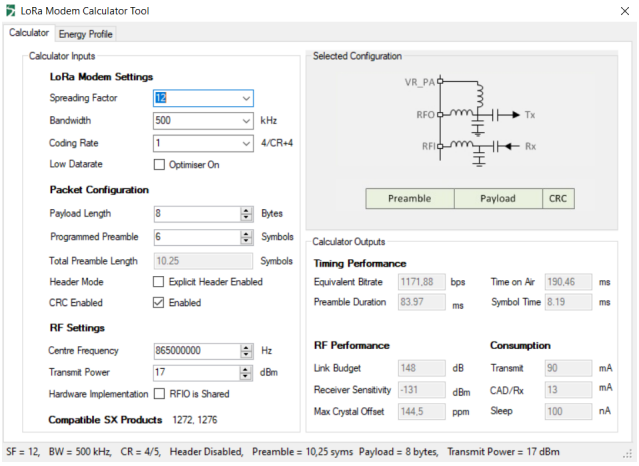


Figure 6.6: User interface for the LoRa time on air calculator program

The parameters required for the calculation are as follows,

Parameter	Value
Spreading factor	7
Bandwidth	250 kHz
Coding rate	1
Payload length	254, 6, and 10 bytes for the payload transmission, routing protocol and MAC protocol transmissions respectively
Pre-amble length	6 bytes
Centre frequency	433 MHz
Tx power	17 Dbm

Using these parameters, the various time on air values are calculated and can be seen in Table 6.3.

Transmission type	time on air (ms)
RREQ/RREP/RREPACK	14.46
RTS/CTS/ACK	17.02
TM	196.22

Table 6.3: Table of results for time on air calculations

The time on air values are used both as fixed values, for the purposes of a deterministic, or fixed, process (Chapter 7), and as the mean of a Poisson distribution for the purposes of a Poisson or exponential process (Chapter 7).

6.4.3.3 Node visibility

There are fifteen possible topology configurations within the model, ranging from a pure star topology to a pure daisy chain. With the exception of these two extremes, all of the topologies were set up in Radio Mobile and relate to real geographic locations.

Within Radio Mobile the general properties of each network were defined as:

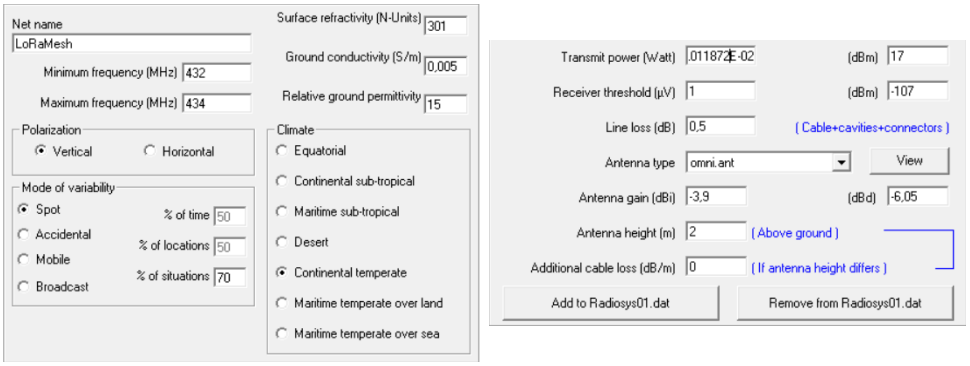
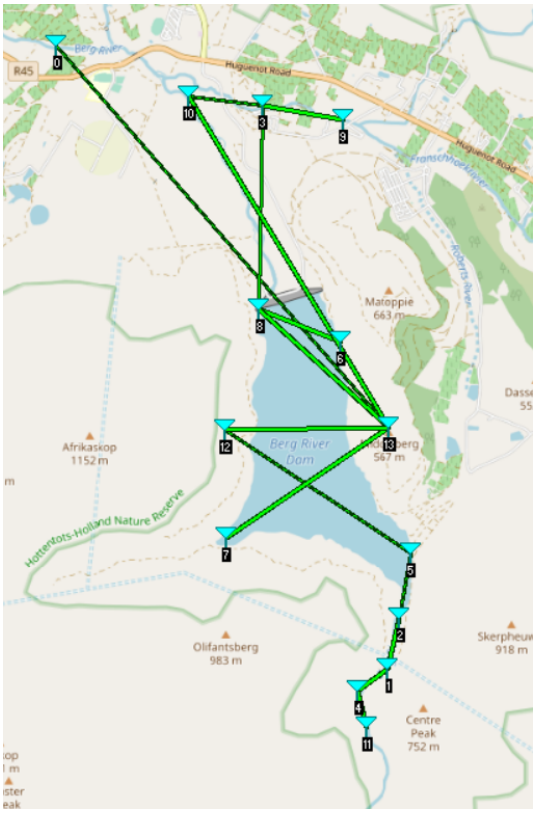


Figure 6.7: Radio Mobile network properties

These hypothetical network configurations were then used to define the visibility between various nodes for each configuration within the model. An example of a configuration is shown in Figure 6.8. The remaining configurations can be found in Appendix A.



(a) Topology of network configuration

```
Net members:          #  01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
0                     01    01 01 37 01 01 36 01 11 37 37 01 01 54 Node
1                     02 01    61 01 70 45 09 01 01 01 01 19 01 20 Node
2                     03 01 61    03 26 53 10 01 15 01 01 15 07 21 Node
3                     04 37 01 03    01 01 18 01 58 62 53 01 01 51 Node
4                     05 01 70 26 01    11 01 01 01 01 01 61 01 10 Node
5                     06 01 45 53 01 11    07 23 34 01 02 08 55 22 Node
6                     07 36 09 10 18 01 07    51 69 01 38 01 24 44 Node
7                     08 01 01 01 01 01 23 51    05 01 01 01 05 61 Node
8                     09 11 01 15 58 01 34 69 05    32 51 01 08 61 Node
9                     10 37 01 01 62 01 01 01 01 32    47 01 01 04 Node
10                    11 37 01 01 53 01 02 38 01 51 47    01 01 60 Node
11                    12 01 19 15 01 61 08 01 01 01 01 01    01 22 Node
12                    13 01 01 07 01 01 55 24 05 08 01 01 01    66 Node
13                    14 54 20 21 51 10 22 44 61 61 04 60 22 66    Terminal

                      Quality = 50 + signal margin in dB
```

(b) Signal strength between nodes

Figure 6.8: The first topology created using Radio Mobile

6.4.3.4 Frequency of transmission generation

As with the time on air, the frequency at which transmissions are generated can either be Poisson or exponentially distributed, or deterministic fixed. In either case, the mean time rate at which a transmission is generated at each node is one transmission every fifteen minutes.

6.4.3.5 Overview of Simulation Process

The simulation begins at the time 0 seconds and at this point only the TMGeneratorEntities are activated. These entities then create the TMEntities and place them into their respective starting nodes. From here the behaviour of the NodeEntities and TMEntities follows that described in Section 6.4.2.1. At this stage of the simulation no node has a route to a gateway and so the routing protocol must be followed. Each node begins this by creating and activating the necessary TMEntities which find a way through the network and a path to the gateway is found. With a path to the gateway now known the node is able to follow the MAC layer protocol and the TMEntities make their way through the network to the gateway. If a node is idle and has no TMEntity

that it is waiting to send, that node's TMGeneratorEntity will create another TMEntity and insert it into the appropriate queue at that node, if the duty cycle allows it.

As soon as a node first begins to try to send a TM, whether this it still needs to find a route to a gateway and sends a RREQ transmission or it has a route and begins with a RTS transmission, the time is logged. Once this TM reaches a gateway the time is logged again and using these recorded times the latency is determined. Along the TMEntity's path to a gateway, every failed transmission between nodes is also recorded, as well as what type of transmission it was that failed.

6.5 Results

6.5.1 Collisions

Configuration	Total Transmissions	Total Failed	RTS/CTS	TM	ACK
1	12600	180	160	20	0
2	12600	280	240	40	0
3	12600	240	180	60	0
4	12600	360	280	80	0
5	12600	100	80	20	0
6	12600	380	180	120	80
7	12600	80	80	0	0
8	12600	120	120	0	0
9	12600	140	140	0	0
10	12600	202	182	20	0
11	12600	320	240	80	0
12	12600	280	240	40	0
13	12600	280	280	0	0
14	12600	360	300	60	0
15	12600	220	180	40	0
Overall Mean	12600	236.13	192.13	38.67	5.33

Table 6.4: Failed transmission statistics from the simulations

Table 6.4 provides statistics on the success rate for transmissions across each topology configuration. Note that the Total Failed column represents transmissions that failed to successfully send on one or more occassion, not transmissions that are completely lost and abandoned. It can be seen that when a transmission fails, the majority of failures occur during the RTS/CTS portion

of the MAC layer protocol. On average, a transmission fails 1.87% of the time, somewhere along it's route to it's destination.

Configuration	Mean Time on Air (ms)	Mean Duty Cycle (%)
1	49610.2157	0.115
2	42647.430	0.0987
3	81032.9760	0.1880
4	84678.80107	0.1960
5	46010.9257	0.1065
6	57207.99714	0.1324
7	40995.28857	0.09490
8	26764.70142	0.06196
9	40826.322857	0.0945
10	65381.9252	0.1513
11	113815.961429	0.26350
12	158056.4757	0.3659
13	23245.59857	0.0538
14	97522.650	0.2257
15	85246.011430	0.1973
Overall Mean	67536.2187	0.1563

Table 6.5: Time on air statistics for the network

Table 6.3 shows the time on air statistics for the various topology configurations in the simulation. It can be seen that each configuration is well within the limit of the Duty Cycle for the 433 MHz frequency band.

6.5.2 Effective throughput

The effective throughput of a network is defined as the number of bytes in a payload TM, not including the bytes making up the MAC and Routing headers as defined in 5, divided by the total time the payload transmission is in the network, from generation until it reaches its destination.

Each payload transmission within the simulation is the same size, 244 bytes with the MAC and Routing headers consisting of 4 bytes and 6 bytes respectively. The effective throughput is thus calculated by dividing the 244 bytes by the mean total transmission time for each topology configuration obtained from the simulation. The results of this can be seen in Table 6.6.

Configuration	Mean Total Transmission Time (s)	Mean Effective Throughput (bytes/s)
1	0.7436	328.13
2	0.7152	341.16
3	1.1937	204.41
4	1.3279	183.75
5	0.6259	389.84
6	0.8580	284.38
7	0.5192	469.95
8	0.3865	631.31
9	0.5925	411.81
10	0.9981	244.46
11	1.6852	144.79
12	2.2471	108.58
13	0.4575	533.33
14	1.5217	160.35
15	1.2288	198.57
Overall Mean	1.0067	242.38

Table 6.6: Mean total transmission time and mean effective throughput

6.5.3 Varying the Arrival Stream

For the purpose of comparing the strength of various predictive models contained in Chapter 7, the simulation was run with arrival streams ranging from one transmission arriving at every node every one minute, to one transmission arriving at every node every fifteen minutes.

Arrival rate (minutes)	Mean Time on Air (ms)	Mean Duty Cycle (%)
1	340674.335	0.7886
2	374041.483	0.8658
3	249950.292	0.5786
4	200739.183	0.4647
5	161263.709	0.3733
6	126329.633	0.2924
7	116874.687	0.2705
8	106967.367	0.2476
9	81187.857	0.1879
10	80007.763	0.1852
11	76225.686	0.1764
12	66610.755	0.1542
13	67536.085	0.1563
14	67549.600	0.1563
15	67536.219	0.1563
Overall Mean	151139.888	0.34986

Table 6.7: Time on air statistics for varying arrival streams

Table 6.7 shows the mean time on air for each node for every topology configuration across the range of arrival streams. Understandably, as more transmissions are being generated more frequently, the nodes spend more time on air. Even at the relatively high frequency of one transmission being generated at each node every minute, the duty cycle is still within the 10% restriction.

Arrival rate (minutes)	Mean Total Transmission Time (s)	Mean Effective Throughput (bytes/s)
1	3.7486	65.09
2	1.8088	134.9
3	1.4462	168.72
4	1.2783	190.88
5	1.188	205.39
6	1.0437	233.78
7	1.047	233.05
8	1.0594	230.32
9	1.1029	221.23
10	0.99	246.46
11	1.0992	221.98
12	1.0079	242.09
13	1.0055	242.67
14	1.0421	234.14
15	1.0067	242.38
Overall Mean	1.325	207.54

Table 6.8: Network performance for varying arrival streams

Table 6.8 shows the mean latency and effective throughput for each topology configuration across the range of arrival streams. At the arrival rate of one minute, the traffic in the network has the greatest impact on the latency, but once that arrival rate is lowered to five minutes, the traffic is low enough to have very little impact on the latency while the network is in a steady-state condition.

6.6 Summary

This chapter began by stating the importance of simulation in the network design process before outlining the simulation style known as Discrete Event Simulation and its dominant approaches. The DES toolkit known as DESMO-J was then discussed and the modelling of the proposed network using DESMO-J explained. The model was parameterised using the characteristics of the radio module selected in Chapter 3. The simulation was run for a number of different topologies and duty cycles. It performed better at lower duty cycles, more in line with the duty cycles that a practically implemented network of this nature would make use of in order to minimize battery usage. The effective throughput of the simulated network is acceptable.

The following chapter concerns the development of a theoretical predictive model using queuing theory. The basic concepts of queuing theory will be

outlined before the application of these concepts, in order to both attempt to develop a closed form expression able to describe the behaviour of the network, as well as to make use of a Markov chain and state-space diagram to model the behaviour of the network.

Chapter 7

Theoretical Modelling Using Queuing Theory

The MACAW MAC layer protocol, as well as MAC layer protocols in general, are essentially queues. Arriving transmissions waiting to be received, or serviced, can be seen as elements within a queue, with the receiving node being seen as the server. The mathematical theory behind queuing analysis can be complex, although the application is relatively straightforward[33]. This chapter will review some basic principles of queuing theory before implementing and comparing a number of different models in an attempt to model the steady state performance of a LPWAN using the MACAW protocol.

7.1 Basic Principles

7.1.1 Single-Server Queues

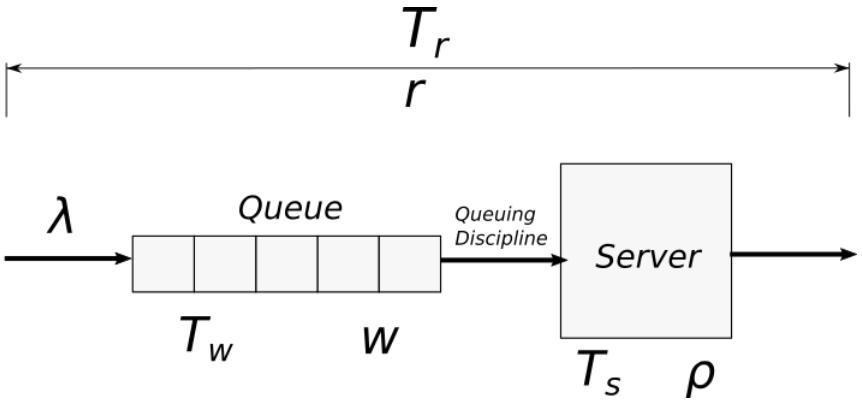


Figure 7.1: Structure of a single-server queue

Figure 7.1 is an example of a simple single-server queue. For this queue, customers arrive from either a finite or non-finite source at a mean **Arrival Rate** (λ). Customers enter the queue and remain there for a mean **Wait Time** (T_w). Customers leave the queue according to a queuing discipline, First In First Out, Last In First Out, etc., at a mean **Service Rate** (μ) before being serviced and leaving the system. This service rate is directly proportional to both the mean **Service Time** (T_s) and the **Server Utilization** (ρ). The mean total residence time of a customer in the system is represented as T_r and the mean number of customers in the system at any given period during steady-state operation is represented by r.

7.1.1.1 Kendall's Notation

Queues are characterised by six parameters, namely:

- Source Population
- Arrival Process
- Server Process
- Number of Servers
- Queuing Discipline
- Queue Length

Source Population

The source population is either finite or infinite, and typically the population is assumed to be infinite[33]. An infinite source population means that the arrival rate of the system is not influenced by size of the source population. In the case of the proposed network, however, the source population cannot be infinite. The size of the buffer of the selected radio module is the same as the size of the payload transmission. Because of this, a node is unable to generate a payload-transmission if it already has one in its buffer waiting to be sent. So, at any given time, there is a finite number of payload-transmissions that can arrive in the system, therefore, the arrival rate is finite and is dependent on the size of the source population.

Arrival Process

The arrival process is the rate at which new payload-transmission arrive in the system[34]. This arrival rate is the mean of a Probability Density Function which describes the probability that a customer will arrive in the system. In a queuing system with an infinite source population, the mean arrival rate will remain constant. However, in the case of the proposed network, the arrival rate

will be dependent on the varying size of the source population. When a payload transmission arrives into the system, the source population will decrease by one, simultaneously decreasing the arrival rate. As a payload-transmission is successfully received by the gateway, a node is able to generate another payload transmission, meaning that the source population has increased by one, increasing the arrival rate.

Server Process

The server process determines the rate at which customers will leave the queue and be serviced by the server before eventually leaving the system. The mean waiting time in the queue, as well as the mean residence time in the system, are determined by the server and arrival processes.

Number of Servers

The number of servers in a queuing system varies depending on what is being modelled. However, in a system with a Markovian arrival/service processes, which will be described later, the change in number of servers impacts the complexity of the application of the queuing theory principles minimally. In the application of queuing theory to the proposed network, and to MAC layer protocols in general, there is only one true server[34], i.e. the gateway.

Queuing Discipline

Queues can be services in a number of ways, the most popular of these being FIFO and LIFO. These disciplines will need to be adjusted to closer match the MAC layer protocols. For example, a payload transmission may be the first to enter the queue, it will then proceed to be serviced by the gateway, but if the transmission fails for any reason, the sending node will have to go into a backoff state and will not be able to be serviced for the length of that backoff period. In that backoff period the gateway will be able to service other transmissions that entered the system after the transmission whose sending node is now in backoff.

Queue Length

The length of a queue is determined by the operational parameters of the queue as well as the type of queue, and has an influence on the stability of the network. Because of the fact that, in the proposed network, the source population is finite, the length of the queue cannot grow infinitely. The length of the queue is limited by the size of the source population.

Kendall's Notation

Kendall's Notation is used to describe a queue in terms of these parameters and is in the from $A/B/m/k/l$, where:

- A defines the Arrival Process
- B defines the Server Process
- m is the number of servers in the system
- k is the maximum queue length
- l is the queuing discipline

Arrival and Server processes can be Markovian, General, Deterministic/Fixed-Length, Hyper-exponential, or Erlang, noted by M, G, D, H, and E respectively. The queue describing the proposed network will be M/M/1/N/FIFO.

7.2 Poisson Process

The Markov arrival and server processes make use of an exponential PDF. The Poisson process resembles a number of physical phenomena and is appropriate for an arrival process that involves a large number of similar, independent sources. The Poisson distribution is given as:

$$P[X(t) = k] = \frac{(\lambda t)^k}{k!} \cdot e^{-\lambda t} \tag{7.1}$$

7.2.1 Properties

Before modelling the MAC layer protocol of the proposed network, there are several important properties of the Poisson distribution which are applicable in determining the arrival and server processes.

7.2.1.1 Superposition property

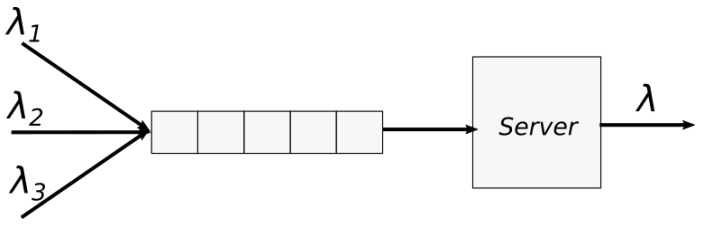


Figure 7.2: Superposition property of the Poisson distribution

The superposition property states that if k number of independent Poisson processes with mean rates of λ_1, λ_2, ..., λ_k are merged, the resulting stream is also Poisson, with a mean rate $\lambda = \lambda_1 + \lambda_2 + ... + \lambda_k$

7.2.1.2 Decomposition property

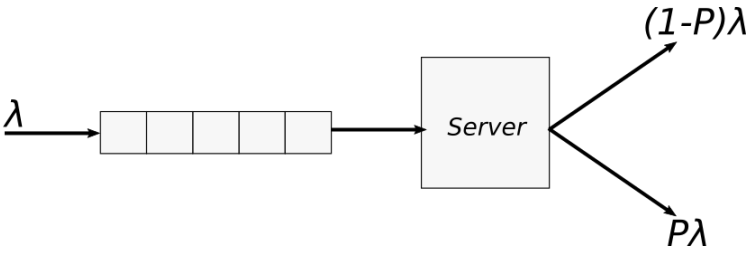


Figure 7.3: Decomposition property of the Poisson distribution

Conversely, the decomposition property states that a Poisson stream with mean arrival rate λ can be divided into k number of independent Poisson streams according to probability P_i where $i\epsilon[1, k]$. Each Poisson stream will then have a mean arrival rate $\lambda_i = P_i\lambda$.

7.2.1.3 Exponentially distributed inter-arrival times

The Poisson distribution mirrors the exponential distribution[34]. If the source population of the queuing system is exponentially distributed, the arrivals at time interval $(0, t)$ is given by the Poisson distribution and the arrival process is Poisson. The probability distribution for arrivals in time interval $(0, t)$ is given by:

$$\begin{aligned} P(\tau \leq t) &= 1 - P(\tau > t) \\ &= 1 - P[X(t) = 0] \\ &= 1 - e^{-\lambda t} \end{aligned} \tag{7.2}$$

Where:
$P(\tau \leq t)$ is the probability that an arrival occurs in the interval $(0, t)$
$P(\tau > t)$ is the probability that no arrival occurs in the interval $(0, t)$
$P[X(t) = 0]$ is the probability that exactly 0 arrivals will occur in the interval $(0, t)$, according to the Poisson distribution
This is the exponential distribution.

7.2.2 Jackson's theorem

Jackson's theorem makes it possible to analyse networks of queues and is based on three assumptions:

1. The network of queues consists of m nodes, each providing an independent exponential service
2. Customers arriving into the system, at any node in the network, do so according to a Poisson arrival rate
3. Once the customer has been served, it immediately goes to another node, or leaves the system.

The theorem states that in this network of queues, each node represents its own independent queuing system. Thus each node can be analysed separately as a M/M/1 or M/M/N queue, and the results of each node can be combined using statistical methods. This means that one can model a packet switching network as a network of queues and analyse it thusly. There are, however, two flaws to this approach. Firstly, the first of the assumptions of the theorem is violated by the application to a packet switching network because the service times at a node in a packet switching are not independent, but are in fact dependent on the length of the queue. However, it has been shown that, because of the averaging effect of merging and partitioning, assuming independent service times gives a good approximation[35]. The second problem is that, for the purposes of the proposed network which aims to be robust, the results of this approach are very dependant on the topology of the network.

7.2.3 Example of an M/M/1 queue system

Using Figure 7.1 as an example of an M/M/1 queue, one can determine the mean residence time, mean system population, mean wait time in the queue, and the mean queue length if the arrival rate and service time are known.

$$\rho = \lambda \cdot T_s \tag{7.3}$$

With the arrival rate and service time known, the server utilization can be obtained using 7.3.

$$T_w = \frac{\rho T_S}{1 - \rho} \tag{7.4}$$

Making use of the server utilization and service time, the mean waiting period within the queue is defined by 7.4.

$$T_r = T_w + T_s \tag{7.5}$$

$$r = \lambda T_r \tag{7.6}$$

$$w = \lambda T_w \tag{7.7}$$

7.5 defines the mean residence time of the system as the sum of the mean waiting period within the queue, and the mean service time. 7.6 represents

Little's Theorem, which states that the mean population of the system is equal to the mean arrival rate of the system multiplied with the mean residence time of the system. This can also be applied to the queue specifically to determine the mean queue length, as in 7.7.

As an example consider an M/M/1 queuing system where 20 customers arrive every minute and the mean service time is 0.6 seconds per customer.

$$T_s = 0.6 \qquad (seconds/customer) \tag{7.8}$$

$$\begin{aligned} \lambda &= 20 \qquad (customers/minute) \\ &= \frac{1}{3} \qquad (customers/second) \end{aligned} \tag{7.9}$$

The server utilization can then be calculated as:

$$\begin{aligned} \rho &= \frac{1}{3} \cdot 0.6 \\ &= 0.2 \end{aligned} \tag{7.10}$$

Substituting 7.10 into 7.4, and then making use of 7.5, 7.6, and 7.7, the following is obtained:

$$T_w = 0.15 \qquad (s) \tag{7.11}$$

$$w = 0.05 \qquad customers \tag{7.12}$$

$$T_r = 0.75 \qquad (s) \tag{7.13}$$

$$r = 0.25 \qquad customers \tag{7.14}$$

7.3 Markov Process

A stochastic, or random, process can be used to model a real time process. If the future conditions of within the process are only dependant on the most recent state, the process is a Markov process. Different states that are linked by a Markov process combine to form what is known as a Markov chain. Markov chains can be either discrete-time or continuous-time. The proposed network, however, will be modelled as a continuous-time Markov chain.

A continuous-time Markov chain is defined by the possibility that a change from one state to another can occur at any time.

7.3.1 Birth-Death Process

A Markovian queuing system with a Poisson arrivals process and an exponential service time can be characterised by the Birth-Death process. The Birth-Death process is a special instance of the continuous-time Markov chain where a state can only transition to it's neighbouring state from the current state.

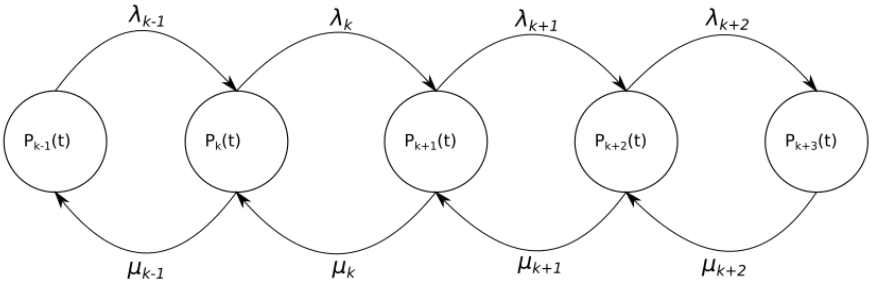


Figure 7.4: Birth-Death process

When a queue is of length k at time t, the chain has the probability $P_k(t)$ of being in state k. A state change from k to $k+1$ is a birth and occurs at rate λ_k. A state change from k to $k-1$ is a death and occurs at rate μ_k.

λ_k and μ_k are calculated as

$$\lambda_k = q_{k,k+1} \tag{7.15}$$

$$\mu_k = q_{k,k-1} \tag{7.16}$$

where $q_{i,j}$ represents the instantaneous rate of transitioning from state i to state j.

The birth-death process can be used to model a single queue Markovian system, in other words an M/M/1 queue.

7.3.1.1 Global and local balance and queue analysis

The probability P_k describes the probability that the system will be in state k. Another way to describe this probability is the fraction of normalised time that the system will be in state k. λP_k and μP_k are the expected rates of transition from state k to $k+1$ and from state k to $k-1$ respectively. These quantities are known as stochastic flow. For equilibrium conditions, the flow out of state k must equal the flow into state k. This is known as the global balance and is depicted in Figure 7.5.

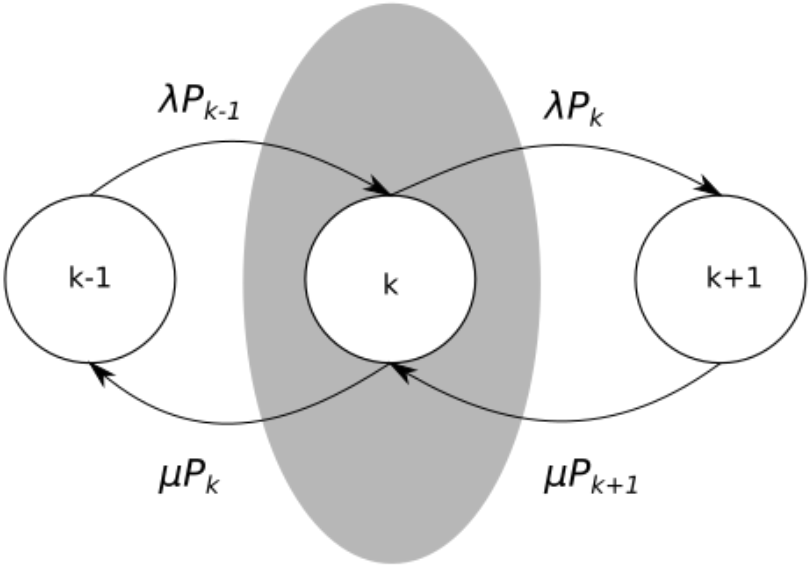


Figure 7.5: Global balance

$$\lambda P_k + \mu P_k = \lambda P_{k-1} + \mu P_{k+1} \tag{7.17}$$

The global balance can be divided into smaller regions, as shown in Figure 7.6. This is known as the local balance.

Applying the equilibrium equation to boundary A in Figure 7.6:

$$\lambda P_{k-2} = \mu P_{k-1} \tag{7.18}$$

and again to boundary B:

$$\lambda P_{k-1} = \mu P_k \tag{7.19}$$

P_k can be calculated using equations 7.18 and 7.19 as follows:

$$\begin{aligned} P_k &= \frac{\lambda}{\mu} P_{k-1} \\ &= \frac{\lambda}{\mu}(\frac{\lambda}{\mu} P_{k-2}) \\ &= (\frac{\lambda}{\mu})^2 P_{k-2}) \\ &= (\frac{\lambda}{\mu})^k P_0 \\ &= \rho^k P_0 \end{aligned} \tag{7.20}$$

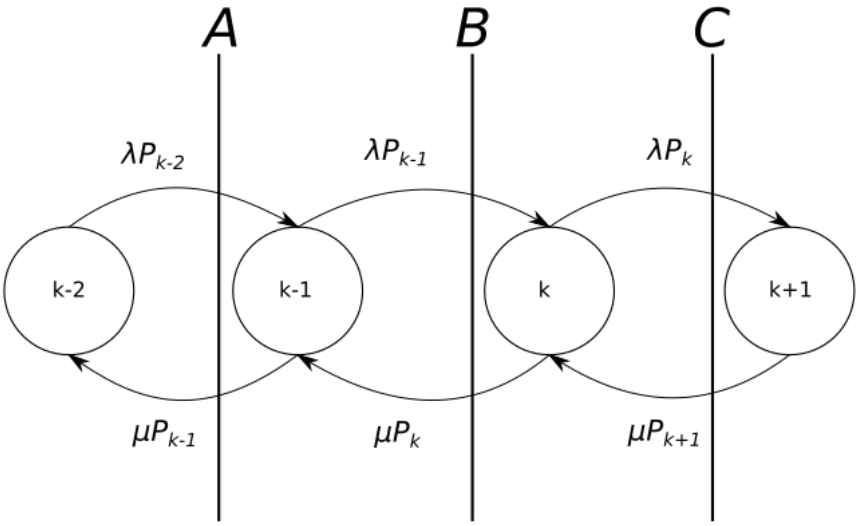


Figure 7.6: Local balance

For a queuing system, the system must be in one of k states, therefore,

$$\sum_k P_k = 1 \tag{7.21}$$

Substituting 7.20 into 7.21 results in:

$$\sum_{k=0}^{\infty} P_k = \sum_{k=0}^{\infty} \rho^k P_0 = 1 \tag{7.22}$$

by making use of the geometric series it is found that:

$$\begin{aligned} \sum_{k=0}^{\infty} \rho^k P_0 &= 1 \\ \frac{P_0}{1-\rho} &= 1 \\ P_0 &= 1-\rho \end{aligned} \tag{7.23}$$

and

$$\begin{aligned} P_k &= \rho^k P_0 \\ &= \rho^k (1-\rho) \end{aligned} \tag{7.24}$$

The length of the queue when the probability that there are P_k customers in the queue, is $L_q = kP_k$. Therefore, summing all k probabilities will provide the actual queue length.

$$\begin{aligned} L_q &= \sum_{k=0}^{\infty} kP_k \\ &= \sum_{k=0}^{\infty} k\rho^k (1-\rho) \\ &= \rho(1-\rho) \sum_{k=0}^{\infty} k\rho^{k-1} \\ &= \frac{\rho(1-\rho)}{(1-\rho)^2} \\ &= \frac{\rho}{1-\rho} \end{aligned} \tag{7.25}$$

Making use of Little's theorem and equation 7.25, the mean wait time can be calculated:

$$\begin{aligned} T &= \frac{L_q}{\lambda} \\ &= \frac{\frac{\rho}{1-\rho}}{\lambda} \\ &= \frac{\frac{\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}}}{\lambda} \\ &= \frac{1}{\mu-\lambda} \end{aligned} \tag{7.26}$$

7.3.2 Finite single queue Markovian system applied to a network

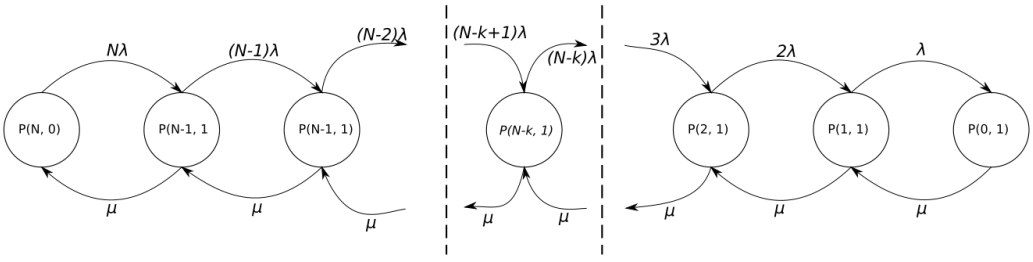


Figure 7.7: Finite telemetry queue state diagram

In Figure 7.7, the probability is given within each state and is used to define the expected transition rate between states. The probability that the system will be in a particular state is given as $P(n, m)$, where n describes the state of the system as well as the size of the source population. m Represents the state of the server itself, a value of 0 indicates that the server is idle, and a value of 1 indicates that the server is busy servicing a transmission.

The states are described as follows:

- P(N, 0) is the probability that N stations can produce new transmissions. This means that N stations are idle and able to produce new transmissions which can enter the queue. From this state, the system can only transition to the P(N-1, 1) state, at a rate of $N\lambda$ as N stations are able to produce a transmission.

- P(N-1, 1) is the probability that that $N-1$ stations can produce a new transmission and 1 station is transmitting and being serviced by the server. From here the system can transition into one of two states. First, it can revert back to the previous state wherein N stations can produce new transmissions and the server is idle. It transitions into this state when the station currently transmitting is successfully serviced by the server without interruption. The other state that the system can transition into, with a new, slower, arrival rate, is when $N-2$ stations are capable of generating a new transmission and the server is still busy servicing one of the two stations trying to transmit, the other station being in waiting to be serviced. The system transitions into this state when a station generates and attempts to send a transmission, while the server is still busy servicing the current transmission.

- P(N-2, 1) is the probability that $N-2$ stations can generate a transmission, one transmission is being serviced by the server and one is waiting

to be serviced. From this state, the process of transitioning to other states is as in the case of the P(N-1,1) state, with the only difference being that the arrival rate of the next transmission is slower, changing from $(N-1)\lambda$ to $(N-2)\lambda$. This pattern continues until the final state P(0,1).

- P(0, 1) is the probability that there are no stations able to generate transmissions, one transmission is being serviced and $N-1$ stations are waiting to be serviced. No new arrivals can occur because all of the stations are currently either being serviced or waiting to be serviced. The only state that the system can transition to from here is the state with probability P(1,1), and can transition here once the transmission being serviced has been fully serviced, at service rate μ.

The global balance equation for the general case $P(N-k,1)$ is as follows:

$$P(N-k,1)((N-k)\lambda+\mu) = P(N-k+1)(N-k+1)\lambda + P(N-k-1)\mu \quad (7.27)$$

$$P(N-k,1)((N-k)\lambda+\mu) - P(N-k+1)(N-k+1)\lambda - P(N-k-1)\mu = 0 \quad (7.28)$$

Applying this balance equation to each possible state k, the equations can be arranged into a $k \times k$ transition matrix A, a $k \times 1$ probability matrix B, and the product of the two matrices, C.

$$A = \begin{bmatrix}
N\lambda & -\mu & 0 & \cdots & & & & & & & & \\
-N\lambda & (N-1)\lambda+\mu & -\mu & \cdots & & & & & & & & \\
0 & -(N-1)\lambda & (N-2)\lambda+\mu & \cdots & & & & & & & & \\
\vdots & \vdots & \vdots & & & & & & & & & \\
\vdots & \vdots & \vdots & & & & & & & & & \\
 & & & & \vdots & \vdots & \vdots & & & & & \\
 & & & & \vdots & \vdots & \vdots & & & & & \\
 & & & \cdots & -(N-k+1)\lambda & (N-k)\lambda+\mu & -\mu & \cdots & & & & \\
 & & & & \vdots & \vdots & \vdots & & & & & \\
 & & & & \vdots & \vdots & \vdots & & & & & \\
 & & & & & & & \vdots & \vdots & \vdots & \vdots & \\
 & & & & & & & \vdots & \vdots & \vdots & \vdots & \\
 & & & & & & & \cdots & -3\lambda & 2\lambda+\mu & -\mu & 0 \\
 & & & & & & & \cdots & 0 & -2\lambda & \lambda+\mu & -\mu \\
 & & & & & & & \cdots & 0 & 0 & -\lambda & \mu
\end{bmatrix}$$

$$B = \begin{bmatrix} P(N,0) \\ P(N-1,1) \\ P(N-1,1) \\ \vdots \\ \vdots \\ \vdots \\ P(N-k+1,1) \\ P(N-k,1) \\ P(N-k-1,1) \\ \vdots \\ \vdots \\ \vdots \\ P(2,1) \\ P(1,1) \\ P(0,1) \end{bmatrix} \qquad C = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Equation 7.21 shows that the sum of all probabilities in B must be 1. The matrix A is then altered as follows to make use of this property in order to solve for matrix B.

$$A = \begin{bmatrix} N\lambda & -\mu & 0 & \cdots & & & & & & & & & \\ -N\lambda & (N-1)\lambda+\mu & -\mu & \cdots & & & & & & & & & \\ 0 & -(N-1)\lambda & (N-2)\lambda+\mu & \cdots & & & & & & & & & \\ \vdots & \vdots & \vdots & & & & & & & & & & \\ \vdots & \vdots & \vdots & & & & & & & & & & \\ & & & & \vdots & \vdots & \vdots & & & & & & \\ & & & & \vdots & \vdots & \vdots & & & & & & \\ & & & \cdots & -(N-k+1)\lambda & (N-k)\lambda+\mu & -\mu & \cdots & & & & & \\ & & & & \vdots & \vdots & \vdots & & & & & & \\ & & & & \vdots & \vdots & \vdots & & & & & & \\ & & & & & & & \vdots & \vdots & \vdots & \vdots & \\ & & & & & & & \vdots & \vdots & \vdots & \vdots & \\ & & & & & & & \cdots & -3\lambda & 2\lambda+\mu & -\mu & 0 \\ & & & & & & & \cdots & 0 & -2\lambda & \lambda+\mu & -\mu \\ & & & & & & & \cdots & 0 & 0 & -\lambda & \mu \\ 1 & 1 & 1 & \cdots & & & & & \cdots & 1 & 1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} P(N,0) \\ P(N-1,1) \\ P(N-1,1) \\ \vdots \\ \vdots \\ \vdots \\ P(N-k+1,1) \\ P(N-k,1) \\ P(N-k-1,1) \\ \vdots \\ \vdots \\ \vdots \\ P(2,1) \\ P(1,1) \\ P(0,1) \end{bmatrix} \qquad C = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Making use of the pseudo-inverse of the transition matrix A, the probability matrix B is calculated as:

$$\begin{aligned} AB &= C \\ B &= A^{+}C \end{aligned} \tag{7.29}$$

With the probability matrix known, the length of the queue is:

$$L_q = \sum_{k=1}^{N} kP(N-k,1) \tag{7.30}$$

The average wait time cannot be calculated in the same manner as in equation 7.26, as the arrival rate, λ, is different for each state. Instead the queue length for the probability that of each possible queue length can be calculated,

$$L_{qk} = kP(N-k,1) \tag{7.31}$$

Little's theorem is then applied to each length L_{qk}

$$\begin{aligned} T_k &= \frac{L_{qk}}{\lambda_k} \\ &= \frac{L_{qk}}{[(N+1)-k]\lambda} \end{aligned} \tag{7.32}$$

Each individual waiting time is then summed to provide the mean waiting period for the system.

$$T = \sum_{k=1}^{N} \frac{kP(N-k,1)}{[(N+1)-k]\lambda} \tag{7.33}$$

7.4 Implementations

This section will detail a number of approaches to modelling the proposed network as a M/M/1 queue system using both the queuing theory fundamentals and Poisson process as described in Section 7.2, as well as the Markovian state space model approach discussed in Section 7.3.

7.4.1 Poisson Process

For the attempts to find a closed form expression describing the behaviour of the network, the arrival rate, λ, is defined as the number of nodes generating transmissions in the network divided by the period in which each node generates one transmission, measured in seconds.

$$\lambda = \frac{N}{T} \tag{7.34}$$

For the case of this network, the arrival rate is then defined as

$$\begin{aligned} \lambda &= \frac{13}{15 * 60} \\ &= 0.0144 \qquad (arrivals/s) \end{aligned} \tag{7.35}$$

The base service time for the system is defined as the time that the gateway node takes to receive an RTS, respond with a CTS, receive a TM and finally respond with an ACK.

$$\begin{aligned} T_s &= 0.01702 + 0.01702 + 0.19622 + 0.01702 \\ &= 0.24728 \qquad (s) \end{aligned} \tag{7.36}$$

These values will be used as a basis for all the approaches using the Poisson process.

7.4.1.1 Single queue, single server with adjusted arrival rate

As seen in Figure 7.8, the configuration of the queue is the same as in that of the example in Section 7.2.3. The service time and arrival rate, however, are adjusted.

The adjustments are in order to account for the possibility that a transmission is unsuccessful and will require retransmission. The probability P is the

probability that a transmission will need to be retransmitted and is defined as the number of times a transmission fails divided by the total number of transmissions attempted, this data is taken from the results of the simulation in Chapter 6.

The new arrival rate is defined as

$$\begin{aligned} \lambda_{adj} &= \lambda \cdot (1 + P) \\ &= \frac{13}{15 * 60} \cdot (1 + \frac{12}{630}) \\ &= 0.01472 \end{aligned} \tag{7.37}$$

The service rate for this configuration is redefined as a weighted average for transmissions that succeed on the first try and transmissions that require retransmission. With the backoff period included as

$$T_{backoff} = 10 \qquad (s) \tag{7.38}$$

$$\begin{aligned} T_{s-adj} &= P(2 \cdot T_s + T_b ackoff) + (1 - P)(T_s) \\ &= P(10.49456) + (1 - P)(0.24728) \\ &= 0.4424662857 \qquad (s) \end{aligned} \tag{7.39}$$

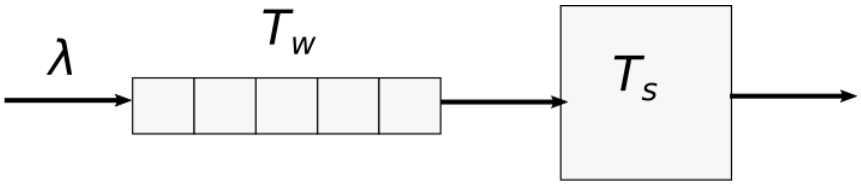


Figure 7.8: Adjusted service time queue

These adjusted values are then used to calculate ρ, T_w, w, T_r, and r, as in equations 7.3 to 7.7.

7.4.1.2 Separate queue and server for transmissions in backoff

The following approaches all model the transmission failure, and subsequent backoff process, as a separate queue and server. The probability P is defined as the probability that a transmission will fail and will need to enter the backoff state, as in section 7.4.1.1.

The first approach to modelling the network in this way is shown in Figure 7.9. The first queue and server function in the same way as for the approach in section 7.4.1.1, the difference coming at the output, where according to probability P, the rate λ_1 is adjusted. These arrivals do not leave the system and instead enter a second queue system in which the service time is increased to take the backoff period into account.

The service time for the second queue is defined as

$$T_{s2} = 10.49456 \qquad (s) \tag{7.40}$$

And the arrival rate for the second queue is defined as

$$\lambda_2 = P \cdot \lambda_1 \tag{7.41}$$

The values for T_{w1} can then be adjusted using equations 7.3 and 7.4, and similarly for T_w2 using T_{s2} and λ_2 in the same equations. T_r is then calculated as

$$T_r = P(T_{w1} + T_{s1} + T_{w2} + T_{s2}) + (1 - P)(T_{w1} + T_{s1}) \tag{7.42}$$

w and r can then be calculated using Little's theorem.

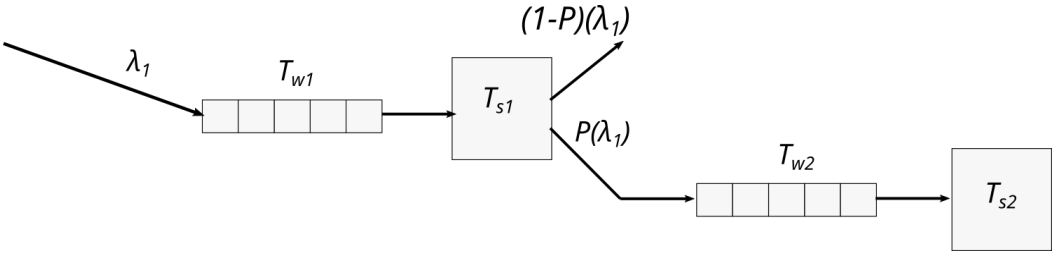


Figure 7.9: Backoff queue 1

The problem with this approach is that it assumes that a transmission will only fail once before it is guaranteed to succeed on the next attempt after backoff. The approach depicted in Figure 7.10 addresses this by looping the output of the second server back to the input of the first queue. Here, T_{s2} is only the backoff period and does not include the second transmission's time on air. T_{w2} is also set at 0, because a transmission that fails does not have to wait in order to backoff, it simply backs off. For this to be true however, the second queue and server relationship is actually that of a M/M/N queue, with N being ∞.

In this configuration, the relationship between λ_1 and λ_2 is as follows

$$\begin{aligned} \lambda_2 &= P(\lambda_1 + \lambda_2) \\ \lambda_2 - P\lambda_2 &= P\lambda_1 \\ (1 - P)\lambda_2 &= P\lambda_1 \\ \lambda_2 &= \lambda_1 \cdot \frac{P}{1 - P} \end{aligned} \tag{7.43}$$

$\lambda_1 + \lambda_2$ can then be calculated as

$$\begin{aligned} \lambda_1 + \lambda_2 &= \lambda_1 + \lambda_1 \cdot \frac{P}{1 - P} \\ &= \lambda_1(1 + \frac{P}{1 - P}) \end{aligned} \tag{7.44}$$

ρ is then calculated by substituting equation 7.44 into 7.3. T_{w1} can then be calculated as in the other approaches. T_r is then calculated as

$$T_r = P(T_{w1} + T_{s1} + T_{w2} + T_{s2}) + (1 - P)(T_{w1} + T_{s1}) \tag{7.45}$$

and Little's theorem is used to calculate w_1 and r.

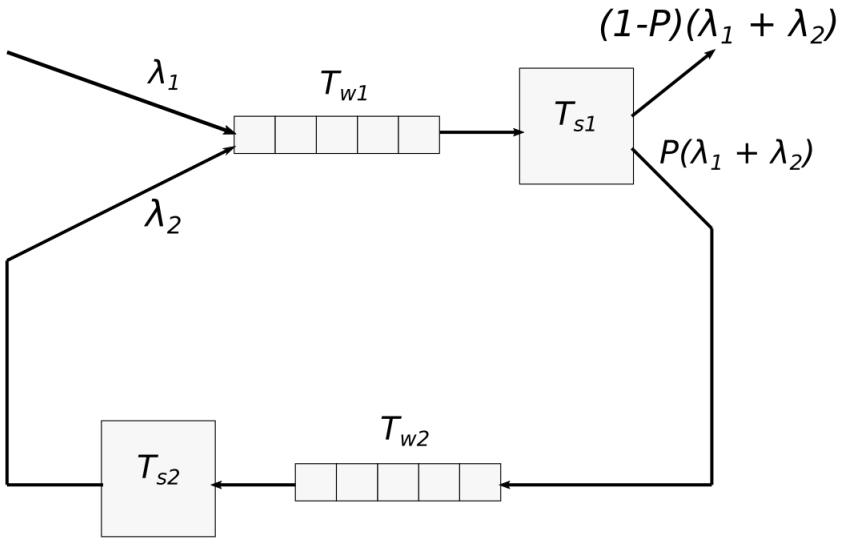


Figure 7.10: Backoff queue 2

The approach shown in Figure 7.11 shifts the arrival rate split to before the transmissions enter the queues, and divides them into separate queues according to probability P. In this case the second server system is still an M/M/1 queue, and the waiting time for the second queue is not 0. In this approach, once the transmissions are split up according to probability P, the initial transmission and then backoff period is combined to form a new T_{s2}.

$$T_{s2} = 10.24728 \qquad (s) \tag{7.46}$$

The relationship between λ_1 and λ_2 is as in the previous approach. ρ_1 and ρ_2 are, therefore, respectively defined as

$$\begin{aligned} \rho_1 &= (1 - P) \cdot (\lambda_1(1 + \frac{P}{1 - P})) \cdot T_{s1} \\ \rho_2 &= P \cdot (\lambda_1(1 + \frac{P}{1 - P})) \cdot T_{s2} \end{aligned} \tag{7.47}$$

T_{w1} and T_{w2} are then calculated using formula 7.4. T_r is calculated as

$$T_r = P(T_{w2} + T_{s2}) + (1 - P)(T_{w1} + T_{s1}) \tag{7.48}$$

and Little's theorem is used to calculate w_1, w_2 and r.

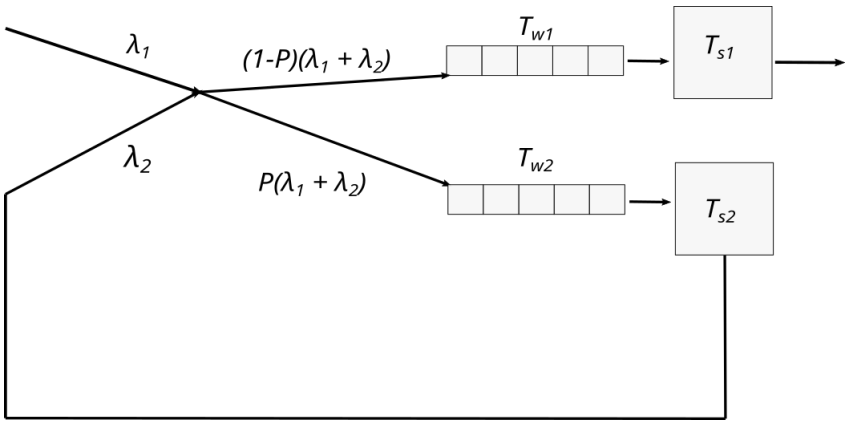


Figure 7.11: Backoff queue 3

7.4.1.3 separate queue and server for multi-hop

The final approach attempts to account for the multi-hop nature of the network. In this instance P_1 is probability of failure, and P_2 is the probability that if successful, the node that a transmission arrived at was a gateway. The combination of this multi-hop with the approach is shown in Figure 7.12. The relationships between λ_1, λ_2 and λ_3 are as follows

$$\lambda_3 = (1 - P_2)(1 - P_1)(\lambda_1 + \lambda_2 + \lambda_3) \tag{7.49}$$

$$\lambda_2 = P_1(\lambda_1 + \lambda_2 + \lambda_3) \tag{7.50}$$

$$\lambda_2 = \frac{P_1(\lambda_1 + \lambda_3)}{1 - P_1} \tag{7.51}$$

Substituting Equation 7.51 into 7.49 results in:

$$\begin{aligned} \lambda_3 &= (1 - P_2)(1 - P_1)(\lambda_1 + \frac{P_1(\lambda_1 + \lambda_3)}{1 - P_1} + \lambda_3) \\ \frac{\lambda_3}{1 - P_2} &= \lambda_1(1 - P_1) + P1(\lambda_1 + \lambda_3) + \lambda_3(1 - P1) \\ &= \lambda_1 + \lambda_3 \end{aligned} \tag{7.52}$$

$$\lambda_3 = \frac{\lambda_1(1 - P_2)}{P_2} \tag{7.53}$$

Equation 7.53 is then substituted into 7.50

$$\begin{aligned} \lambda_2 &= P_1(\lambda_1 + \lambda_2 + \frac{\lambda_1(1-P_2)}{P_2}) \\ \frac{P_2\lambda_2}{P_1} &= P_2\lambda_2 + P_1 \\ \lambda_1 &= (\frac{P_2}{P_1} - P_2)\lambda_2 \\ \lambda_2 &= \frac{\lambda_1}{\frac{P_2}{P_1} - P_2} \end{aligned} \tag{7.54}$$

Using equations 7.53 and 7.54 results in

$$\lambda_1 + \lambda_2 + \lambda_3 = \lambda_1(1 + \frac{1}{\frac{P_2}{P_1} - P_2} + \frac{1-P_2}{P_2}) \tag{7.55}$$

This relationship is then used to obtain ρ_1 and T_{w1}, T_r is then calculated as

$$T_r = P_1(T_{w1} + T_{s1} + T_{s2}) + (1-P_1)(T_{w1} + T_{s1}) \tag{7.56}$$

and w_1 and r can be calculated using Little's theorem.

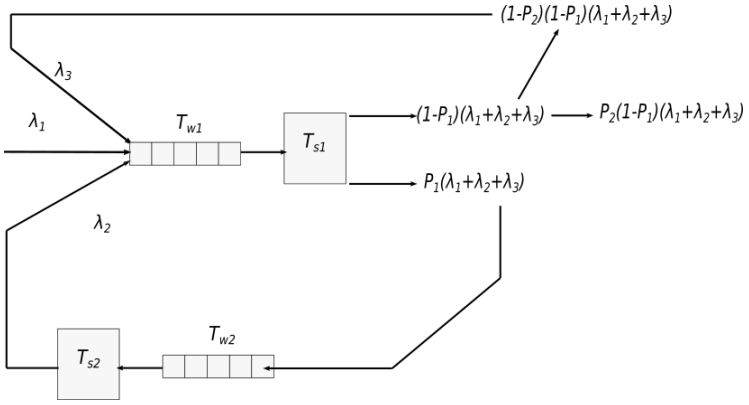


Figure 7.12: Multi-hop queue

7.4.2 Markov Process

7.4.2.1 Existing state-space model

Figure 7.13 depicts the state space model presented by [36]. This example has 5 nodes.

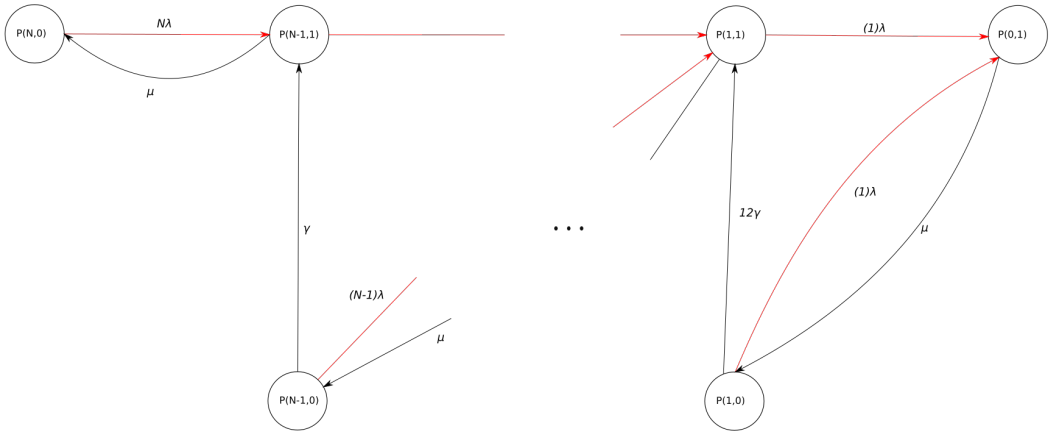


Figure 7.13: Existing state-space model

The variables in Figure 7.13 are:

λ	Poisson arrival rate per station
μ	Poisson service rate for the system
γ	Backoff rate per station
N	Number of stations in the system
$P(n, m)$	Probability that n stations are available to enter the system, with m describing the state of the server, a value of 0 meaning that the server is idle and 1 meaning that the server is busy servicing a transmission

The service rate μ is defined as:

$$\mu = \frac{1}{t_{dat} + t_{ack}} = \frac{1}{t_{ser}} \tag{7.57}$$

where

t_{dat}	The time required for a node to transmit its data packet
t_{ack}	The time required for the server to transmit its ACK packet
t_{ser}	$t_{dat} + t_{ack}$

The states can be described as follows:

P(N, 0)

This represents the probability that the system is in a state in which the server is idle and there are N nodes available to generate a transmission. From this state, the system can only transition into the P(N-1, 1) state. The transition rate is $N\lambda$, as there are N nodes capable of generating a transmission at the

base arrival rate λ

P(N-1, 1)

The probability that the system is in a state in which there are $N-1$ nodes available to generate a transmission, and the server is busy servicing 1 transmission. From this state, the system can transition into the state P(N-2, 1) at a rate of $(N-1)\lambda$. This signifies another node generating a transmission and inserting it into the queue. The other possible transition from this state is into the state P(N, 0). This state change signifies the server completely finishing servicing the current transmission before a new transmission is generated. This change takes place at the service rate μ.

P(N-1, 0)

The probability that the system is in a state in which there are $(N-1)$ nodes available to generate a transmission and the server is idle. This state can only be transitioned into from the state P(N-2, 0). The process of transitioning to and from this state are as follows. While the system is still in state P(N-1, 1), another node generates a transmission which enters the queue, in reality disrupting the transmission of the original transmission being serviced. One of the nodes will then go into backoff while the other is being serviced. Once the node has finished being serviced the system transitions from state P(N-2, 1) to state P(N-1, 0). In this state the server is idle while the remaining transmission is in backoff. From this state the system transitions into the state P(N-1, 1) when the node comes out of backoff. This occurs at rate γ. This pattern of state transition between P(N-1, 1), P(N-2, 1), and P(N-1, 0) repeats until the state P(0, 1)

P(0, 1)

The probability that the system is in a state in which there are no nodes available to generate transmissions, $(N-1)$ nodes are in backoff and 1 nodes is being serviced by the server. From here the system can only transition into state P(1, 0) by finishing servicing the 1 transmission.

The global balance equations in matrix form are then as follows:

$$
\begin{bmatrix}
N\lambda & -\mu & 0 & 0 & 0 & 0 & \cdots & & & & \\
-N\lambda & (N-1)\lambda+\mu & -\gamma & 0 & 0 & 0 & \cdots & & & & \\
0 & 0 & (N-1)\lambda+\gamma & -\mu & 0 & 0 & \cdots & & & & \\
0 & -(N-1)\lambda & -(N-1)\lambda & (N-2)\lambda+\mu & -\gamma & 0 & \cdots & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & & & & \\
 & & & & & & & \vdots & \vdots & \vdots & \vdots \\
 & & & & & & & \vdots & \vdots & \vdots & \vdots \\
 & & & & & & & \cdots\ 0 & 0 & \lambda+(N-1)\gamma & -\mu \\
 & & & & & & & \cdots\ 0 & -\lambda & -\lambda & \mu \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \cdots & 1 & 1 & 1 & 1
\end{bmatrix}
\times
$$

$$
\begin{bmatrix}
P(N,0) \\
P(N-1,1) \\
P(N-1,0) \\
P(N-2,1) \\
P(N-2,0) \\
\vdots \\
\vdots \\
P(1,1) \\
P(1,0) \\
P(0,1)
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 1
\end{bmatrix}
$$

The probability matrix B can be solved using the same manner as in 7.3.2. The formulae for the queue length L_q and waiting period T need adjustment, however. Firstly, the probability that the system has a queue length of each possible value must be calculated. To find this the probabilities for each state with the same queue length are summed, the result of which is multiplied by

the relevant queue length in order to obtain L_{qk}, each value of L_{qk} is then summed, resulting in L_q.

$$L_q = \sum_{k=1}^{N} k[P(N-k,1) + P(N-k,0)] \tag{7.58}$$

The mean waiting period of the queue is then calculated as

$$T = \sum_{k=1}^{N} \frac{k[P(N-k,1) + P(N-k,0)]}{(N-k+1)\lambda} \tag{7.59}$$

7.4.2.2 Expanded state-space model

The model in 7.4.2.1 makes use of two possible states for each queue length k. However, this model requires the service time to be adjusted to take collisions and noise into account.

Van Staden[34] expands the existing model to include the other states of the MAC protocol, requiring fewer adjustments to the service time. The expanded model shown in Figure 7.14 is based on Van Staden's expansion.

A description of the symbols in Figure 7.14 are as follows:

λ	Arrival rate per node
μ'	Service rate for the RTS and CTS transmissions
μ''	Service rate for the payload transmission
μ'''	Service rate for the ACK transmission
μ	The rate at which the system returns from the collision state to the idle state
γ	The return rate from the idle state to the RTS/CTS transmission state
N	The total number of nodes in the model
$P(m,n)$	The probability of being in a particular state in which m nodes are able to generate a transmission, and n refers to one of five possible states.

The possible values of n are as follows:

0	Both the server and the communication channel are idle
1	The server is busy servicing an RTS transmission and responding with a CTS transmission.
2	This is the collision state, a new arrival has occurred while the receiving node was busy servicing a transmission.
3	The server is busy servicing a payload transmission
4	The server has serviced the payload transmission and is responding with an ACK transmission.

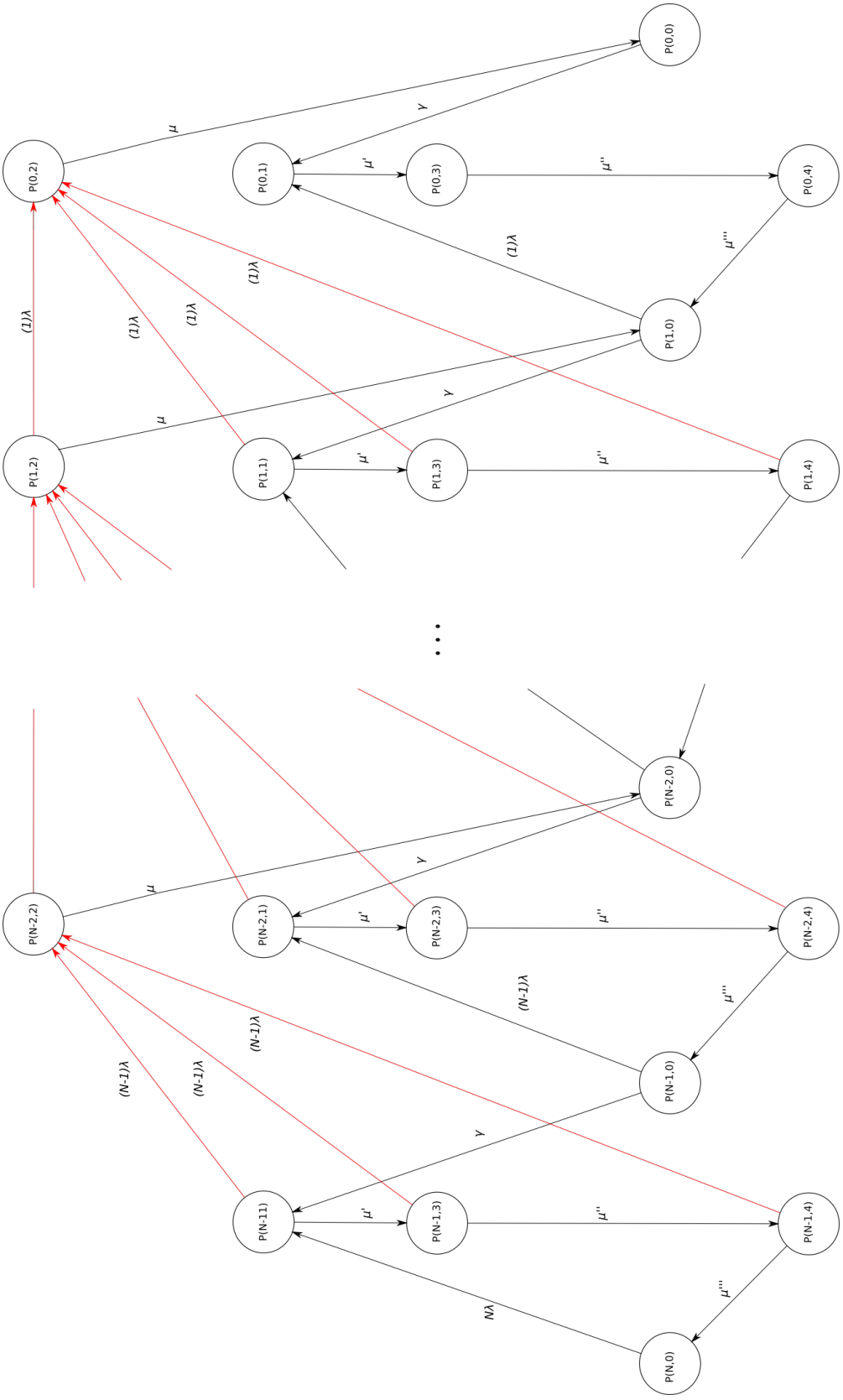


Figure 7.14: Expanded state-space model

Below is a description of the flow between states in the model.

P(N, 0)

No nodes have generated transmissions which are being serviced, or waiting to be serviced. The arrival rate for any node generating a transmission while in this state is, therefore, $N\lambda$. From this state, a node can generate a transmission at the arrival rate $N\lambda$ and transition to state $P(N-1,1)$.

P(N-1, 1)

In this state, $(N-1)$ nodes can generate a transmission and one node has generated a RTS transmission and is being serviced. Any newly generated transmissions will arrive at a rate $(N-1)\lambda$. There are two possible states into which the system can transition from this state. The first possibility is that the RTS is successfully serviced and the server successfully responds with a CTS transmission, all without being interrupted. The system will then transition into the state $P(N-1,3)$ according to the service rate μ'. The second possibility is that a new transmission is generated at rate $(N-1)\lambda$, the process of the server servicing the initial RTS transmission is interrupted and the system transitions to the state $P(N-2,2)$ according to the arrival rate.

P(N-1, 3)

This state behaves in the same way as the previous state, $P(N-1,1)$. It can either transition to the state $P(N-1,4)$ when the server completes servicing the payload transmission, the system will then transmission to state $P(N-1,4)$ at the service rate μ''. The other possibility is that the service is interrupted by an arriving transmission, in which case the system will transition into the state $P(N-2,2)$ according to the arrival rate $(N-1)\lambda$.

P(N-1, 4)

This state represents the final stage in the MAC protocol. When the server finishes sending the ACK transmission, the system will transition back to the idle state $P(N,0)$ according to the service rate μ'''. As in the previous two states, the transmission may also be interrupted by an arriving transmission, in which case the system will transition to the collision state $P(N-2,2)$ according to the arrival rate $(N-1)\lambda$.

P(N-2, 2)

This state is the collision state. When a transmission arrives into the system while the server is not idle, the system will transition to this state. From this state the system can transition into two possible states. The first possibility is the state $P(N-3,2)$, this occurs when a new transmission is generated and arrives while the system is still in the collision state, this transition will take place according to the arrival rate $(N-2)\lambda$. The other possibility is that the two transmissions finish colliding and transition to an idle state. The system will then transmission to the state $P(N-2,0)$ at service rate μ.

P(N-2, 0)

In this state both the server and communications channel are idle and there are two transmissions in the backoff state, waiting to be serviced. There are two possible states into which the system can transition from the state. The first of which is back into the state $P(N-2,1)$ as one of the waiting transmissions exits the backoff state and restarts the MAC protocol. This transitions takes place according to the backoff rate γ. The other possibility is that a new transmission is generated and arrives in the system while the two waiting transmissions are in backoff. The system then transitions into the state $P(N-3,1)$ according to the arrival rate $(N-2)\lambda$.

The global balance equations in matrix form are as follows:

$$
\begin{bmatrix}
N\lambda & 0 & 0 & -\mu''' & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
-N\lambda & (N-1)\lambda+\mu' & 0 & 0 & -\gamma & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
0 & -\mu' & (N-1)\lambda+\mu'' & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
0 & 0 & -\mu'' & (N-1)\lambda+\mu''' & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
0 & 0 & 0 & 0 & (N-1)\lambda+\gamma & 0 & 0 & -\mu''' & 0 & \cdots & & & & & & \\
0 & -(N-1)\lambda & -(N-1)\lambda & -(N-1)\lambda & 0 & (N-2)\lambda+\mu & 0 & 0 & 0 & \cdots & & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
 & \cdots & 0 & -(N-k+1)\lambda & 0 & (N-k)\lambda+\mu' & 0 & 0 & -\gamma & 0 & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & -\mu' & (N-k)\lambda+\mu'' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & -\mu'' & (N-k)\lambda+\mu''' & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & -\mu & 0 & 0 & 0 & (N-k)\lambda+\gamma & 0 & 0 & 0 & -\mu''' & 0 & \cdots & \\
 & \cdots & 0 & 0 & -(N-k)\lambda & -(N-k)\lambda & -(N-k)\lambda & -(N-k)\lambda & 0 & (N-k-1)\lambda+\mu & 0 & 0 & 0 & 0 & \cdots & \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & & \cdots & 0 & -\lambda & -\lambda & -\lambda & -\lambda & 0 & \mu & 0 & 0 & 0 & 0 \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & \lambda & 0 & \mu' & 0 & 0 & -\gamma \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu' & \mu'' & 0 & 0 \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu'' & \mu''' & 0 \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 & 0 & \gamma \\
1 & 1 & 1 & 1 & \cdots & & & & & & \cdots & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\times
$$

$$\begin{bmatrix} P(N,0) \\ P(N-1,1) \\ P(N-1,3) \\ P(N-1,4) \\ P(N-1,0) \\ P(N-2,2) \\ P(N-2,1) \\ P(N-2,3) \\ P(N-2,4) \\ P(N-2,0) \\ \vdots \\ \vdots \\ P(1,0) \\ P(0,2) \\ P(0,1) \\ P(0,3) \\ P(0,4) \\ P(0,0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The probability matrix B can be solved using the pseudo inverse of the transition matrix A, as in 7.4.2.1.

The queue length L_q is given by the following equation:

$$L_q = \sum_{k=1}^{N} k[P(N-k,0)+P(N-k,1)+P(N-k,2)+P(N-k,3)+P(N-k,4)] \tag{7.60}$$

And the mean waiting period is given by:

$$L_q = \sum_{k=1}^{N} \frac{k[P(N-k,0)+P(N-k,1)+P(N-k,2)+P(N-k,3)+P(N-k,4)]}{(N+1-k)\lambda} \tag{7.61}$$

7.4.2.3 Alterations to the expanded state-space model

The model in 7.4.2.2, while suited to Van Staden's network, is designed with a star topology in mind. In order to model the proposed mesh network of this

investigation, this model will need to be adjusted. This section will discuss three possible approaches to modelling a multi-hop-mesh network by adjusting the state space model in 7.4.2.2.

Weighted average service rate

The first approach is to adjust the service rates according to data obtained from the DESMO-J simulations. From the simulations, the average number of retransmissions, for traversing the network and not for transmission failure, is tracked and implemented in the form of a weighted average. For this average, $P(h)$ is used to denote the probability that a payload requires h retransmissions to successfully traverse the network and arrive at a gateway. 7.62 shows the process of calculated this adjusted service rate for μ'

$$\begin{aligned} t_{rts/cts-adj} &= \sum_{h=1}^{H} P(h) \cdot h \cdot t_{rts/cts} \\ \mu'_{adj} &= \frac{1}{t_{rts/cts-adj}} \end{aligned} \tag{7.62}$$

This same adjustment is applied to all service rates in the model in 7.4.2.2.

Alternative flow

The next approach, instead of adjusting the service rates for each transmission based on the average number of retransmissions, directly changes the flow between states in the configuration. This adjusted model is shown in Figure 7.15.

The change comes in the $P(N-k,4)$ states, wherein the server has received the payload transmission and is replying with an ACK transmission. Instead of either the server completing this tast successfully and the transmission leaving the system, and a new arrival interrupting the system and causing a collision, it is now possible for an ACK transmission to be successful, with the number of entities in the system remaining the same. In Figure 7.15, it shows that the from the $P(N-k,4)$ state, the system can transition to either the $P(N-k+1,0)$ state, as in 7.4.2.2, or to the $P(N-k,1)$ state. To achieve this the service rate μ'''' is introduced and both μ'''' and μ''' are adjusted according to probability $P(v)$. $P(v)$ is defined as the probability that if a successful transmission takes place between two nodes, one of those nodes is a gateway. This is determined by the configurations used in the simulations themselves and is defined as:

$$P(v) = \frac{V}{N} \tag{7.63}$$

Where V is the average number of nodes that have a direct link to a gateway, and N is the total number of nodes (excluding the gateway).

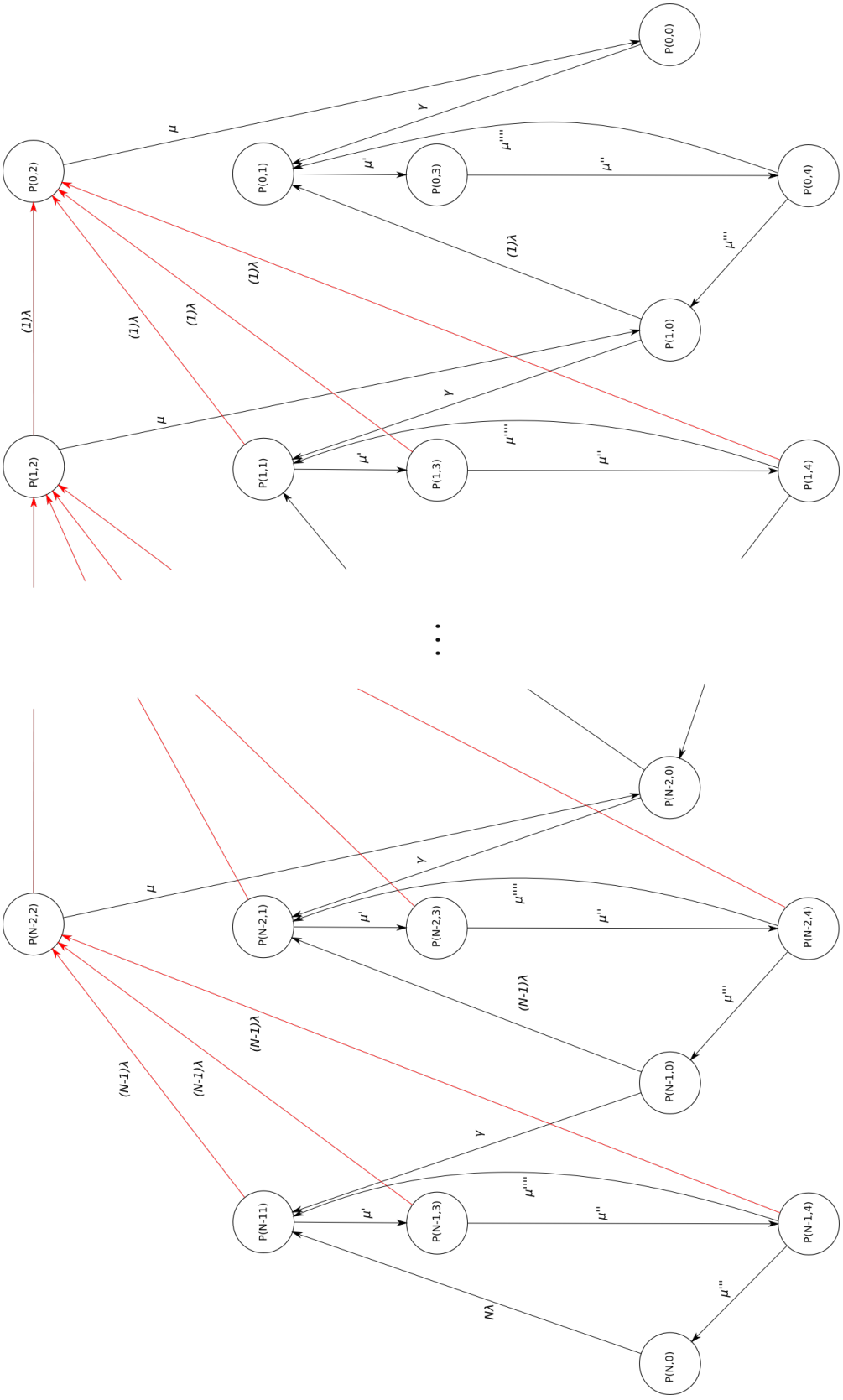


Figure 7.15: Adjusted state-space model with loop

The service rates are then calculated as follows:

$$\mu''' = P(v) \cdot \frac{1}{t_{ack}} \qquad \mu'''' = (1 - P(v)) \cdot \frac{1}{t_{ack}} \tag{7.64}$$

The adjusted transition matrix A is shown below:

$$\begin{bmatrix}
N\lambda & 0 & 0 & -\mu''' & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & & \\
-N\lambda & (N-1)\lambda+\mu' & 0 & -\mu'''' & -\gamma & 0 & 0 & 0 & 0 & \cdots & & & & & & & \\
0 & -\mu' & (N-1)\lambda+\mu'' & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & & \\
0 & 0 & -\mu'' & (N-1)\lambda+\mu'''+\mu'''' & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & & \\
0 & 0 & 0 & 0 & (N-1)\lambda+\gamma & 0 & 0 & -\mu''' & 0 & \cdots & & & & & & & \\
0 & -(N-1)\lambda & -(N-1)\lambda & -(N-1)\lambda & 0 & (N-2)\lambda+\mu & 0 & 0 & 0 & \cdots & & & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & & & & \\
 & & & & & & & & & & & & & & & & \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & \\
 & \cdots & 0 & -(N-k+1)\lambda & 0 & (N-k)\lambda+\mu' & 0 & -\mu'''' & -\gamma & 0 & 0 & 0 & 0 & 0 & \cdots & & \\
 & \cdots & 0 & 0 & 0 & -\mu' & (N-k)\lambda+\mu'' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & \\
 & \cdots & 0 & 0 & 0 & -\mu'' & (N-k)\lambda+\mu'''+\mu'''' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & \times \\
 & \cdots & 0 & 0 & -\mu & 0 & 0 & 0 & (N-k)\lambda+\gamma & 0 & 0 & 0 & -\mu''' & 0 & \cdots & & \\
 & \cdots & 0 & 0 & -(N-k)\lambda & -(N-k)\lambda & -(N-k)\lambda & -(N-k)\lambda & 0 & (N-k-1)\lambda+\mu & 0 & 0 & 0 & 0 & \cdots & & \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & \\
 & & & & & & & & & & & & & & & & \\
 & & & & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
 & & & & \cdots & 0 & -\lambda & -\lambda & -\lambda & -\lambda & 0 & \mu & 0 & 0 & 0 & 0 & \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & \lambda & 0 & \mu' & 0 & -\mu'''' & -\gamma & \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu' & \mu'' & 0 & 0 & \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu'' & \mu'''+\mu'''' & 0 & \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 & 0 & \gamma & \\
1 & 1 & 1 & 1 & \cdots & & & & & & \cdots & 1 & 1 & 1 & 1 & 1 &
\end{bmatrix}$$

The queue length L_q and waiting time T can then be calculated as in 7.4.2.2.

Service time based on queue length

In each of the state space models used to describe a mesh network in this section, the server in the model is not just the gateway node, but is actually modelled as all of the nodes in the network that could possibly receive a transmission.

The previous approach changed the flow of the state space model itself. This new approach, again, alters the service rate of the server. The distinction that this approach makes is that, while more than one node is generating and attempting to send a transmission, it is not necessarily true that these nodes are all attempting to communicate with the same node. Therefore, in this case, the service rate will increase. This is simply implemented by multiplying each service rate by the number of nodes generating and attempting to send a transmission.

$$\mu_{adj} = k\mu \tag{7.65}$$

The transition matrix is then as follows:

$$
\begin{bmatrix}
N\lambda & 0 & 0 & -\mu''' & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
-N\lambda & (N-1)\lambda+\mu' & 0 & -\mu''' & -\gamma & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
0 & -\mu' & (N-1)\lambda+\mu'' & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
0 & 0 & -\mu'' & (N-1)\lambda+\mu'''+\mu'''' & 0 & 0 & 0 & 0 & 0 & \cdots & & & & & & \\
0 & 0 & 0 & 0 & (N-1)\lambda+\gamma & 0 & 0 & -\mu''' & 0 & \cdots & & & & & & \\
0 & -(N-1)\lambda & -(N-1)\lambda & -(N-1)\lambda & 0 & (N-2)\lambda+2\mu & 0 & 0 & 0 & \cdots & & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
 & \cdots & 0 & -(N-k+1)\lambda & 0 & (N-k)\lambda+k\mu' & 0 & -k\mu'''' & -\gamma & 0 & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & -k\mu' & (N-k)\lambda+k\mu'' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \\
 & \cdots & 0 & 0 & 0 & 0 & -k\mu'' & (N-k)\lambda+k\mu'''+k\mu'''' & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \times \\
 & \cdots & 0 & 0 & -k\mu & 0 & 0 & 0 & (N-k)\lambda+\gamma & 0 & 0 & 0 & -k\mu''' & 0 & \cdots & \\
 & \cdots & 0 & 0 & -(N-k)\lambda & -(N-k)\lambda & -(N-k)\lambda & -(N-k)\lambda & 0 & (N-k-1)\lambda+(k+1)\mu & 0 & 0 & 0 & 0 & \cdots & \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & & & & \cdots & 0 & -\lambda & -\lambda & -\lambda & -\lambda & 0 & N\mu & 0 & 0 & 0 & 0 \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & \lambda & 0 & N\mu' & 0 & -N\mu'''' & -\gamma \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -N\mu' & N\mu'' & 0 & 0 \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -N\mu'' & N\mu''' + N\mu'''' & 0 \\
 & & & & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & -N\mu & 0 & 0 & 0 & \gamma \\
1 & 1 & 1 & 1 & \cdots & & & & & & \cdots & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

At first glance, this approach seems too broad in the sense that it implies that no two nodes are sending to the same destination node. However, for the model to be more specific, the model and flow would have to be reconfigured for each different network topology, which would be impractical and not at all suited to the general case.

7.5 Results

Based on the results in the table it is clear that the biggest factor influencing the residence time for the Poisson models is the failure rate of transmissions. This is due to the large discrepancy between the time it takes to transmit a payload and the time in which a transmission is in the backoff state. The failure rate for the transmissions was taken from the data obtained from the simulations in Chapter 6.

The results for the attempts to model the network by making use of a closed form expression describing the network were poor. However, it was known beforehand that it would be difficult, if not impossible, to model a multi-hop mesh network in this way with any degree of accuracy.

Method	Section	Residence population	Residence time (s)	Queue length	Wait time (s)
Adjusted Arrival Rate	7.4.1.1	0.00653678	0.44409	2.38615-5	0.00162107
Backoff queue 1	7.4.1.2	0.006526	0.44335	1.28083e-5	0.00090323
Backoff queue 2	7.4.1.2	0.006459	0.43865988	1.33067e-5	0.000903681
Backoff queue 3	7.4.1.2	0.006467	0.439188	2.10880e-5	0.030423
Multi-hop queue	7.4.1.3	0.0140289	0.4397	6.2735e-5	0.001966359

Table 7.1: Results of Poisson process implementations

Table 7.2 shows the results of the attempts to model the network using a state-space model and a Markov chain. Immediately notable is the variation in the results of the different approaches.

At first glance it would appear as though the existing model in 7.4.2.1 performs very poorly in comparison to the results obtained from the simulation in Chapter 6. However, adjusting the service time to account for failed transmissions and collisions as Wolhuter[36] did does improve this result. It does not account for multi-hop networks though.

The expanded model in 7.4.2.2 builds on the existing model in an attempt to negate the need for an adjusted service time. This still does not account for multi-hop networks, however, and as such performs poorly when compared to the results in Chapter 6. It is not entirely without merit though, as the result is within 17% of the simulation result for a star topology network.

The first attempt at adjusting the expanded model to account for a multi-hop network delivered poor results. Simply adjusting the service rates according to a weighted average of the number of hops required to reach a gateway proved to be a poor approach. The result is within 72.6% of the simulated result which is unacceptable.

The next approach, in which the flow of the state-space model was altered, offered greatly improved results, being within 21.4% of the simulated result. The most promising result, however, is the final approach, which was also the simplest, multiplying the service rate by the current queue length. This re-

sulted in a model that was within 13.3% of the simulated result.

Method	Section	Residence population	Residence time (s)
Existing model	7.4.2.1	0.04484956	0.2599158
Expanded model	7.4.2.2	0.00612925	0.53578668
Service time 1	7.4.2.2	0.01897506	1.73763958
Alternate flow	7.4.2.2	0.01362505	1.2222596
Service time 2	7.4.2.2	0.00962577	1.14047776

Table 7.2: Results of Markov process implementations

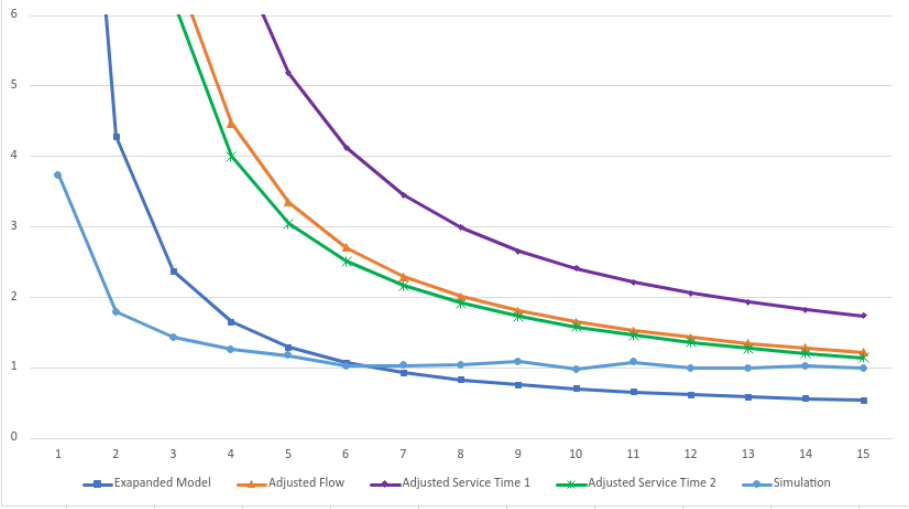


Figure 7.16: Comparison between predictive models and simulation for different arrival rates

Figure 7.16 shows how the previously mentioned theoretical predictive models compare to the simulated results in Chapter 6 for arrival rates ranging from one to fifteen minutes. It can be seen that for arrival rates higher than one transmission per node every eleven minutes, the expanded model proposed by Van Staden is able to more accurately predict the behaviour of the network, however, at the original arrival rate, the model presented in 7.4.2.3 is still the most accurate. The possibility that at these higher duty cycles, the congestion on the network causes it t experience traffic similar to that of a network with a

star topology, this is likely the reason behind Van Staden's model performing so well.

7.6 Summary

This chapter began by introducing some important concepts of queuing theory before explaining two approaches to the applications of these concepts. The Poisson process, in which a closed form expression is obtained which describes the behaviour of the queue, and the Markov process, which models the queue as a Markov chain.

A number of models were developed using both the Poisson and Markov processes. The Poisson process models were deemed to be unacceptable and it is not recommended to attempt to model a multi-hop network in such a way in the future. The Markov process models were much more successful, with the final presented model being able to predict the effective throughput of the network with acceptable accuracy at the original duty cycle, for the proposed network.

Chapter 8

Summary and Conclusion

8.1 Project Outcomes

In the first chapter of this thesis, an overview of LPWAN implementations was presented in order to gain a greater understanding of the challenges faced by LPWANs, as well as how they have been modelled and simulated in previous work.

Following on from this, a number of LPWAN technologies were researched and compared across various criteria, including, but not limited to, modulation technique, operating frequency, range, and data rate. LoRa was chosen as the LPWAN technology for this project as it provides a good range, generally consumes little power, has no limit to the number of transmissions in a day, besides the duty cycle restrictions applied to any radio broadcasting done in the ISM bands and is cost effective. The specific radio module which would be used to parameterise the models in later chapters, was selected as the RFM98W, as it is readily available, reasonably priced, consumes little power, and operates at the chosen frequency of 433MHz.

A study into the Data Link Layer of networks was then conducted. The challenges faced by wireless sensor networks were discussed before the investigation of a number of MAC layer protocols, namely ALOHA, CSMA, TDMA, and MACAW. MACAW was selected as the MAC Layer protocol to be modelled in both the simulation and queuing theory.

Similarly, the Network Layer was studied. Important concepts of routing algorithms were evaluated before investigating a number of routing algorithms of interest to this project. Subsequently the AODV algorithm was selected.

Chapter 6 details the use of Discrete Event Simulation and more specifically, the DES platform DESMO-J in modelling and simulating the wireless

sensor network using the aforementioned MAC layer protocol and routing algorithm. The process oriented approach was combined with the various black box components of the DESMO-J package to model the network. The MACAW MAC layer protocol and the AODV routing protocol were modelled. Using the LoRa time on air calculator program, in conjunction with the parameters of the radio module selected in Chapter 3, the time on air for all the various transmissions involved in both the routing and MAC layer protocols were calculated and applied to the model. Using the Radio Mobile platform, a number of hypothetical network topologies were created with these configurations also fed into the simulation. The frequency at which transmissions were generated was also varied, in increments of one minute ranging from one minute to fifteen minutes. The simulation was run across all configurations and all transmission arrival times. As could be expected, with shorter periods in between transmission arrivals, the network experienced more collisions and a lower effective throughput. The highest effective throughput being 242 Bytes/s while the lowest was 65 Bytes/s.

In Chapter 7, the basic concepts of queuing theory were briefly explained before discussing two methods of applying these concepts, namely the Poisson and Markov processes. Several applications of both methods were then considered. Any approach consisting of a "network" of queues was deemed unacceptable, as any results would be specific to that network's topology and not adequately robust. The first approach consisted merely of altering the arrival rate by accounting for failed transmissions having to re-attempt as a new transmission entering the system. Next, the flow of the queue system was altered to include a separate queue for transmissions that had failed and entered into the backoff state. This instance, however, did not account for the possibility of a transmission failing more than once, and so the next iteration connected the output of the backoff queue to the input of the initial transmissions queue. An alternative to this was to move the split between failed and successful transmissions to the arrival rates, and have separate queues for transmissions that were going to be successful, and those that were going to fail. The final approach was to attempt to model the multi-hop nature of the network by introducing another fork. After a transmission was deemed to have been successful, there is a probability that it might require retransmission anyway as the node where it has just successfully arrived, is not a gateway. Ultimately, none of these closed-form solutions were acceptable. The complexity of including all parameters accurately in these solutions is significant.

The models produced using the Markov process gave much better results. Initially, the expanded state space model put forward by Van Staden was altered for use with the MACAW MAC layer protocol. However, this model was designed to represent a network with a star topology. At the original arrival rate of one transmission at each node every fifteen minutes, it was not able to

accurately track the performance of the network, with the resulting waiting period being within 47% of the latency achieved in the simulation. Building on this approach, the probability that each message would require a certain number of hops before reaching a gateway was used in a weighted average to determine the service rate. The resulting waiting period, however, was an unsatisfactory 72.6% of that achieved in the simulation. Altering the flow of Van Staden's expanded model so that the system is able to go from the state of transmitting an ACK message, directly back to sending an RTS message, without the queue length changing, produced much better results. The final approach was to alter the service rate again, but this time altering it based on the length of the queue, i.e. how many transmissions were already in the system. This resulted in a waiting period within 13.3% of the simulated results, at the original arrival rate. This is a much better result and well within the tolerances to be reasonably expected for this type of system performance.

The results of both the simulation and all Markov process theoretical predictive models were compared at different arrival rates, ranging from one minute to fifteen minutes. This comparison showed that for networks in which the frequency at which transmissions are generated at each node is lower than one transmission at each node every eleven minutes, Van Staden's expanded model provided a closer prediction than the model presented in 7.4.2.3, but at the original arrival rate the model presented in 7.4.2.3 is much more accurate.

8.2 Contributions

The work conducted as set out in this thesis, contain the following contributions:

- The performance of multihop networks is not simple to predict, particularly due to the great variety in topologies encountered. Past attempts at simulation have proved to be reasonably successful, but documented work tends to be very focused on a particular configuration. In this case the simulation framework developed, is reasonably flexible and fairly easy to adapt to different network layouts. Being DES based, it is also capable to provide useful results independent of simulation hardware capabilities. As such a useful tool has been created to estimate network performance prior to committing financial and other resources for deployment.

- Accurate theoretical modelling and analysis of this type of network is not very common. The approach followed in this work built on earlier attempts also based on queueing networks, but the present one is again somewhat more general. Results correspond with the simulations to an acceptable degree and presents a parallel confirmatory option. The type of approach followed has proved to be interesting, versatile and

quite powerful. There is certainly more potential to be explored by way of extended investigation. The work documented, could however be a useful point of departure.

- Ultimately, two tools were developed in this project. Namely the DESMO-J simulation of the multi-hop network using the MACAW communications protocol and AODV routing protocol, and the queue length dependent Markov process theoretical predictive model. Both of these tools are useful aids for the network design process.

8.3 Conclusions

Itemized below are the conclusions drawn from the work presented in this thesis.

- DESMO-J, whilst most popularly used in for the modelling and simulation of logistical problems, is a very suitable toolkit for the purposes of performance prediction of a LPWAN.
- Even with transmissions generated at every node at intervals of one minute, the duty cycle of each node in the network is still within the 10% limit. However, the simulation was not consistently able to find routes for every node in each possible configuration with this much traffic in the network. The routing protocol for this scenario could be investigated for further improvement.
- For the MACAW MAC layer protocol, the adjusted service rate model in section 7.4.2.3 was able to predict the performance of the network with sufficient accuracy at lower arrival rates, i.e. transmissions arriving at intervals of 11 minutes or more per node.
- At all inter-arrival times greater than eleven minutes per node, the expanded model put forward by Van Staden, adjusted to fit the MACAW protocol, proved to be more accurate than the model presented in section 7.4.2.3.
- The type of network under investigation at the time, will dictate which of the above is the better option.

8.4 Further Work

- The AODV protocol used in the simulation should be improved to allow for more stable simulation at a faster rate of transmission generation. The stability of the routing protocol was not of great importance to the

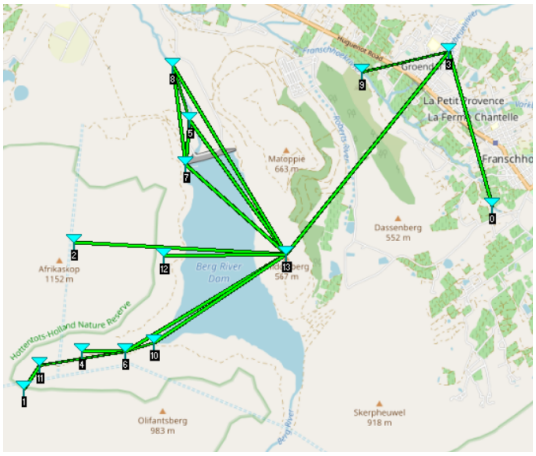
work presented in this thesis, as all results gained from the simulation needed to be obtained when the network was operating in a steady state in order to be compared to those of the theoretical predictive model, i.e. the results had to be obtained after all nodes had already discovered routes.

- The simulation should be expanded to include larger networks to investigate the performance at a larger scale. This would also allow the models presented to be tested for more nodes and would provide greater insight into the accuracy of these models for use in the general case.
- The simulation should be expanded to be configurable to the end user, allowing for alteration of radio hardware parameters, for example, to see how the network would perform using different hardware.
- The simulation should be expanded to model the battery performance of the nodes, as this is an important part of the overall success of a wireless sensor network.
- The application of queuing theory for this case should be further investigated in order to find which of the network parameters has the biggest influence in the accuracy of the Poisson and Markov approaches.
- The network should be practically built and deployed, as such real world data would serve to further validate the performance of both the simulated and theoretical models proposed in this thesis.

Appendices

Appendix A

Network topologies created using the Radio Mobile platform.

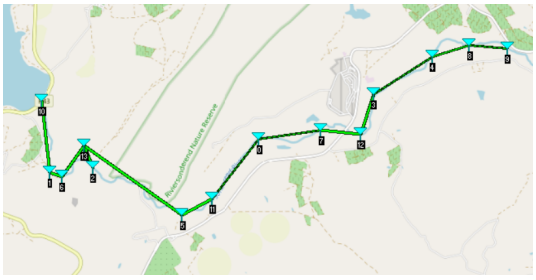


(a) Topology of network configuration 2

```
Net members:            #  01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
0                       01    00 00 50 00 00 00 00 00 00 00 00 00 00 Node
1                       02 00    00 00 00 00 00 00 00 00 00 50 00 00 Node
2                       03 00 00    00 00 00 00 00 00 00 00 00 00 50 Node
3                       04 50 00 00    00 00 00 00 00 00 50 00 00 00 50 Node
4                       05 00 00 00 00    00 50 00 00 00 00 00 00 00 00 Node
5                       06 00 00 00 00 00    00 50 50 00 00 00 00 50 Node
6                       07 00 00 00 00 50 00    00 00 00 50 50 00 50 Node
7                       08 00 00 00 00 00 50 00    50 00 00 00 00 50 Node
8                       09 00 00 00 00 00 50 00 50    00 00 00 00 50 Node
9                       10 00 00 00 50 00 00 00 00 00    00 00 00 00 Node
10                      11 00 00 00 00 00 00 50 00 00 00    00 00 50 Node
11                      12 00 50 00 00 00 00 50 00 00 00 00    00 00 Node
12                      13 00 00 00 00 00 00 00 00 00 00 00 00    50 Node
13                      14 00 00 50 50 00 50 50 50 00 50 00 50    Terminal

                        Quality = 50 - number of resend
```

(b) Signal strength between nodes

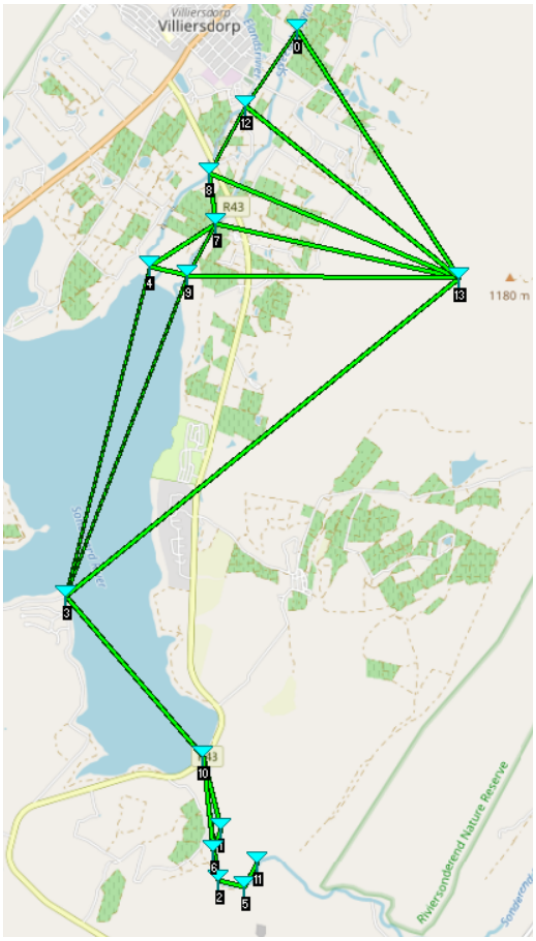


(a) Topology of network configuration 3

```
Net members:            #  01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
0                       01    00 00 00 00 50 00 50 00 00 00 50 00 00 Node
1                       02 00    00 00 00 00 50 00 00 00 50 00 00 00 Node
2                       03 00 00    00 00 00 00 00 00 00 00 00 00 50 Node
3                       04 00 00 00    50 00 00 00 00 00 00 00 50 00 Node
4                       05 00 00 00 50    00 00 00 50 00 00 00 00 00 Node
5                       06 50 00 00 00 00    00 00 00 00 00 50 00 50 Node
6                       07 00 50 00 00 00 00    00 00 00 00 00 00 50 Node
7                       08 50 00 00 00 00 00 00    00 00 00 00 50 00 Node
8                       09 00 00 00 00 50 00 00 00    50 00 00 00 00 Node
9                       10 00 00 00 00 00 00 00 00 50    00 00 00 00 Node
10                      11 00 50 00 00 00 00 00 00 00 00    00 00 00 Node
11                      12 50 00 00 00 00 50 00 00 00 00 00    00 00 Node
12                      13 00 00 00 50 00 00 00 50 00 00 00 00    00 Node
13                      14 00 00 50 00 00 50 50 00 00 00 00 00 00    Terminal

                        Quality = 50 - number of resend
```

(b) Signal strength between nodes

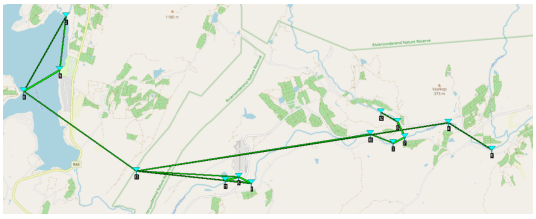


(a) Topology of network configuration 4

```
Net members:            #  01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
0                       01    02 03 46 47 01 02 46 49 43 08 01 57 61 Node
1                       02 02    48 33 14 18 72 06 06 07 64 10 03 39 Node
2                       03 03 48    23 18 62 69 08 08 09 58 17 05 25 Node
3                       04 46 33 23    53 03 32 46 45 53 60 01 47 59 Node
4                       05 47 14 18 53    01 17 61 52 72 22 01 48 50 Node
5                       06 01 18 62 03 01    25 01 01 01 15 59 01 14 Node
6                       07 02 72 69 32 17 25    08 08 09 62 11 04 47 Node
7                       08 46 06 08 46 61 01 08    61 58 13 01 49 67 Node
8                       09 49 06 08 45 52 01 08 61    52 14 01 56 63 Node
9                       10 43 07 09 53 72 01 09 58 52    14 01 46 64 Node
10                      11 08 64 58 60 22 15 62 13 14 14    04 11 45 Node
11                      12 01 10 17 01 01 59 11 01 01 01 04    01 01 Node
12                      13 57 03 05 47 48 01 04 49 56 46 11 01    61 Node
13                      14 61 39 25 59 50 14 47 67 63 64 45 01 61    Terminal

                        Quality = 50 + signal margin in dB
```

(b) Signal strength between nodes

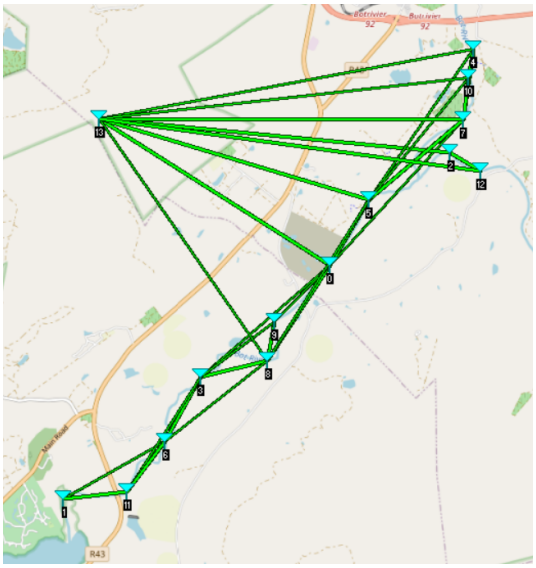


(a) Topology of network configuration 5

```
Net members:            #  01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
LoRa Node1              01    01 54 01 01 01 01 01 01 61 01 01 01 55 Node
LoRa Node2              02 01    01 10 08 01 68 10 01 01 14 60 01 56 Node
LoRa Node3              03 54 01    01 01 01 01 01 01 56 01 01 01 23 Node
LoRa Node4              04 01 10 01    43 15 17 64 44 01 55 13 16 30 Node
LoRa Node5              05 01 08 01 43    55 14 46 36 01 19 17 31 53 Node
LoRa Node6              06 01 01 01 15 55    01 22 28 01 17 01 33 08 Node
LoRa Node7              07 01 68 01 17 14 01    19 06 01 22 71 01 56 Node
LoRa Node8              08 01 10 01 64 46 22 19    59 01 33 17 29 49 Node
LoRa Node9              09 01 01 01 44 36 28 06 59    01 17 03 54 03 Node
LoRa Node10             10 61 01 56 01 01 01 01 01 01    01 01 01 26 Node
LoRa Node11             11 01 14 01 55 19 17 22 33 17 01    25 12 56 Node
LoRa Node12             12 01 60 01 13 17 01 71 17 03 01 25    01 54 Node
LoRa Node13             13 01 01 01 16 31 33 01 29 54 01 12 01    01 Node
LoRa Node14             14 55 56 23 30 53 08 56 49 03 26 56 54 01    Terminal

                        Quality = 50 + signal margin in dB
```

(b) Signal strength between nodes

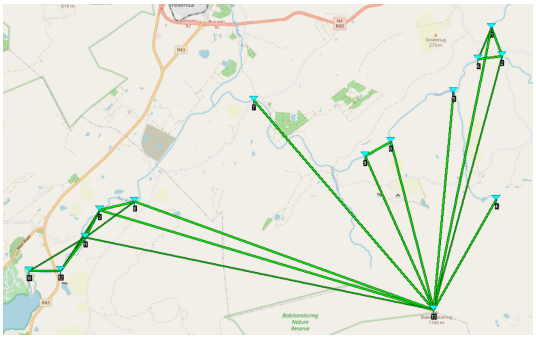


(a) Topology of network configuration 6

Net members:	#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	Role:
LoRa Node1	01		16	25	32	20	52	25	34	52	45	13	22	16	52	Node
LoRa Node2	02	16		07	26	09	16	30	13	22	18	01	51	01	10	Node
LoRa Node3	03	25	07		12	36	44	09	39	23	19	37	08	57	56	Node
LoRa Node4	04	32	26	12		14	21	51	19	62	33	01	46	08	10	Node
LoRa Node5	05	20	09	36	14		28	10	55	15	13	68	12	25	47	Node
LoRa Node6	06	52	16	44	21	28		22	50	41	34	26	23	27	56	Node
LoRa Node7	07	25	30	09	51	10	22		19	27	27	01	54	04	12	Node
LoRa Node8	08	34	13	39	19	55	50	19		27	29	54	18	29	57	Node
LoRa Node9	09	52	22	23	62	15	41	27	27		60	05	21	14	29	Node
LoRa Node10	10	45	18	19	33	13	34	27	29	60		12	23	11	31	Node
LoRa Node11	11	13	01	37	01	68	26	01	54	05	12		01	27	26	Node
LoRa Node12	12	22	51	08	46	12	23	54	18	21	23	01		01	11	Node
LoRa Node13	13	16	01	57	08	25	27	04	29	14	11	27	01		59	Node
LoRa Node14	14	52	10	56	10	47	56	12	57	29	31	26	11	59		Terminal

Quality = 50 + signal margin in dB

(b) Signal strength between nodes

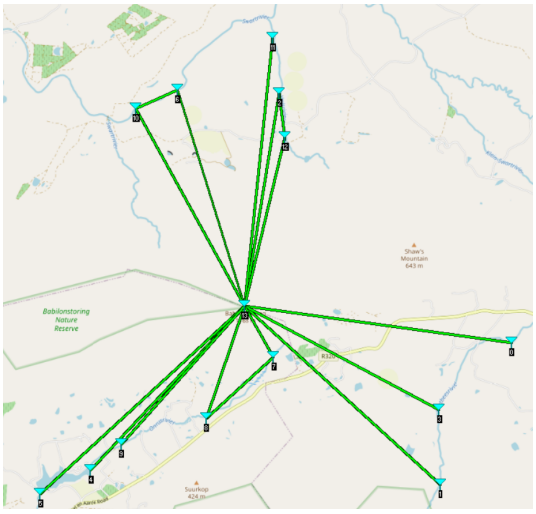


(a) Topology of network configuration 7

Net members:	#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	Role:
LoRa Node1	01		09	11	12	12	19	16	01	11	53	10	11	11	52	Node
LoRa Node2	02	09		58	01	28	22	25	01	01	06	01	01	01	55	Node
LoRa Node3	03	11	58		01	10	22	57	05	04	13	01	01	01	20	Node
LoRa Node4	04	12	01	01		06	01	01	11	62	05	26	55	45	51	Node
LoRa Node5	05	12	28	10	06		43	13	03	01	09	02	06	01	65	Node
LoRa Node6	06	19	22	22	01	43		47	03	01	16	01	01	01	53	Node
LoRa Node7	07	16	25	57	01	13	47		03	02	20	01	01	01	36	Node
LoRa Node8	08	01	01	05	11	03	03	03		17	01	06	11	05	59	Node
LoRa Node9	09	11	01	04	62	01	01	02	17		06	23	33	21	58	Node
LoRa Node10	10	53	06	13	05	09	16	20	01	06		06	07	06	56	Node
LoRa Node11	11	10	01	01	26	02	01	01	06	23	06		31	56	48	Node
LoRa Node12	12	11	01	01	55	06	01	01	11	33	07	31		53	39	Node
LoRa Node13	13	11	01	01	45	01	01	01	05	21	06	56	53		38	Node
LoRa Node14	14	52	55	20	51	65	53	36	59	58	56	48	39	38		Terminal

Quality = 50 + signal margin in dB

(b) Signal strength between nodes

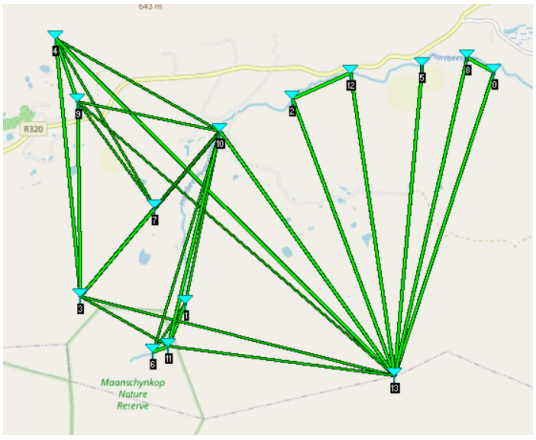


(a) Topology of network configuration 8

Net members:	#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	Role:
LoRa Node1	01		14	01	21	01	01	01	01	01	01	01	01	01	54	Node
LoRa Node2	02	14		01	42	01	01	01	08	01	01	01	01	01	61	Node
LoRa Node3	03	01	01		01	01	01	15	01	01	01	12	45	69	53	Node
LoRa Node4	04	21	42	01		01	01	01	07	01	01	01	01	01	58	Node
LoRa Node5	05	01	01	01	01		40	01	24	24	43	01	01	01	62	Node
LoRa Node6	06	01	01	01	01	40		01	24	24	33	01	01	01	56	Node
LoRa Node7	07	01	01	15	01	01	01		01	01	01	60	17	15	35	Node
LoRa Node8	08	01	08	01	07	24	24	01		53	45	01	01	01	43	Node
LoRa Node9	09	01	01	01	01	24	24	01	53		33	01	01	01	62	Node
LoRa Node10	10	01	01	01	01	43	33	01	45	33		01	01	01	59	Node
LoRa Node11	11	01	01	12	01	01	01	60	01	01	01		11	16	53	Node
LoRa Node12	12	01	01	45	01	01	01	17	01	01	01	11		46	53	Node
LoRa Node13	13	01	01	69	01	01	01	15	01	01	01	16	46		54	Node
LoRa Node14	14	54	61	53	58	62	56	35	43	62	59	53	53	54		Terminal

Quality = 50 + signal margin in dB

(b) Signal strength between nodes



(a) Topology of network configuration 9

Net members:	#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	Role:
LoRa Node1	01		06	28	28	13	33	12	16	60	03	22	23	28	53	Node
LoRa Node2	02	06		17	26	33	15	18	18	06	23	32	61	15	23	Node
LoRa Node3	03	28	17		39	32	37	04	30	33	13	41	35	50	57	Node
LoRa Node4	04	28	26	39		52	41	05	62	26	52	63	05	40	50	Node
LoRa Node5	05	13	33	32	52		13	01	23	11	56	42	38	18	54	Node
LoRa Node6	06	33	15	37	41	13		01	22	48	08	31	41	34	56	Node
LoRa Node7	07	12	18	04	05	01	01		02	06	01	01	61	01	38	Node
LoRa Node8	08	16	18	30	62	23	22	02		17	23	37	29	21	45	Node
LoRa Node9	09	60	06	33	26	11	48	06	17		08	24	23	27	52	Node
LoRa Node10	10	03	23	13	52	56	08	01	23	08		17	28	13	35	Node
LoRa Node11	11	22	32	41	63	42	31	01	37	24	17		62	36	57	Node
LoRa Node12	12	23	61	35	05	38	41	61	29	23	28	62		39	13	Node
LoRa Node13	13	28	15	50	40	18	34	01	21	27	13	36	39		57	Node
LoRa Node14	14	53	23	57	50	54	56	38	45	52	35	57	13	57		Terminal

Quality = 50 + signal margin in dB

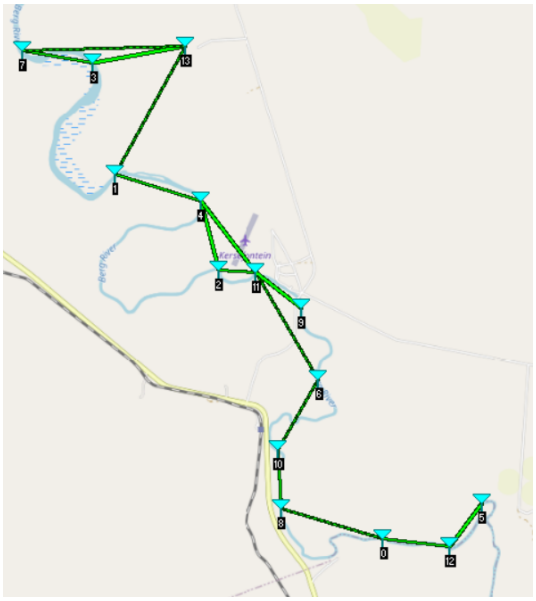
(b) Signal strength between nodes

(a) Topology of network configuration 10

```
Net members:          #  01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
LoRa Node1            01    37 50 48 40 42 40 32 32 37 51 55 46 39 Node
LoRa Node2            02 37    47 43 59 67 46 39 49 48 51 44 59 45 Node
LoRa Node3            03 50 47    55 49 47 36 35 40 42 55 35 53 43 Node
LoRa Node4            04 48 43 55    50 45 32 45 42 46 52 42 42 44 Node
LoRa Node5            05 40 59 49 50    55 29 44 53 54 50 39 53 50 Node
LoRa Node6            06 42 67 47 45 55    69 43 49 31 49 40 54 25 Node
LoRa Node7            07 40 46 36 32 29 69    14 19 20 47 38 38 14 Node
LoRa Node8            08 32 39 35 45 44 43 14    55 45 41 33 40 51 Node
LoRa Node9            09 32 49 40 42 53 49 19 55    49 41 33 42 49 Node
LoRa Node10           10 37 48 42 46 54 31 20 45 49    46 42 48 57 Node
LoRa Node11           11 51 51 55 52 50 49 47 41 41 46    57 53 44 Node
LoRa Node12           12 55 44 35 42 39 40 38 33 33 42 57    39 42 Node
LoRa Node13           13 46 59 53 42 53 54 38 40 42 48 53 39    47 Node
LoRa Node14           14 39 45 43 44 50 25 14 51 49 57 44 42 47    Terminal

Quality = 50 + signal margin in dB
```

(b) Signal strength between nodes

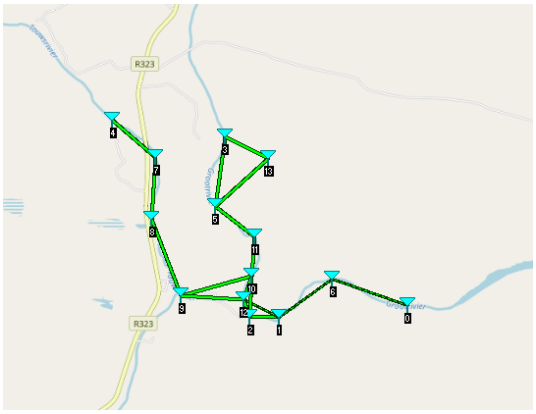


(a) Topology of network configuration 11

Net members:	# 01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
LoRa Node1	01 22 27 16 23 33 36 21 53 29 40 27 56 26 Node
LoRa Node2	02 22 51 51 57 04 44 46 40 39 39 49 22 55 Node
LoRa Node3	03 27 51 41 59 16 50 39 44 49 35 57 27 51 Node
LoRa Node4	04 16 51 41 43 05 38 57 34 30 23 41 19 59 Node
LoRa Node5	05 23 57 59 43 12 45 42 38 43 29 58 25 49 Node
LoRa Node6	06 33 04 16 05 12 27 01 31 20 30 19 60 13 Node
LoRa Node7	07 36 44 50 38 45 27 36 46 52 53 53 38 44 Node
LoRa Node8	08 21 46 39 57 42 01 36 31 35 35 39 19 54 Node
LoRa Node9	09 53 40 44 34 38 31 46 31 32 58 33 50 41 Node
LoRa Node10	10 29 39 49 30 43 20 52 35 32 39 60 30 36 Node
LoRa Node11	11 40 39 35 23 29 30 53 35 58 39 36 42 29 Node
LoRa Node12	12 27 49 57 41 58 19 53 39 33 60 36 30 51 Node
LoRa Node13	13 56 22 27 19 25 60 38 19 50 30 42 30 27 Node
LoRa Node14	14 26 55 51 59 49 13 44 54 41 36 29 51 27 Terminal

Quality = 50 + signal margin in dB

(b) Signal strength between nodes

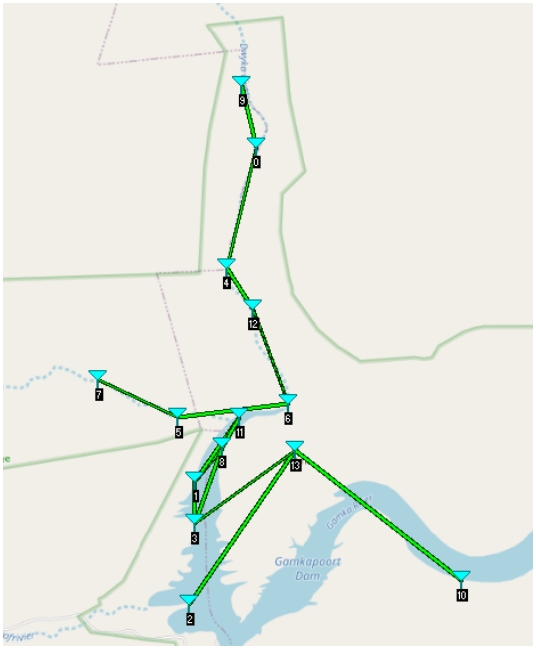


(a) Topology of network configuration 12

Net members:	# 01 02 03 04 05 06 07 08 09 10 11 12 13 14 Role:
0	01 23 24 01 04 11 54 07 15 29 03 05 26 05 Node
1	02 23 75 22 09 30 53 18 25 37 36 25 55 19 Node
2	03 24 75 34 24 24 52 18 30 23 66 49 80 32 Node
3	04 01 22 34 34 60 01 39 22 18 35 39 33 69 Node
4	05 04 09 24 34 26 01 59 36 37 16 23 18 37 Node
5	06 11 30 24 60 26 12 27 16 15 41 69 20 64 Node
6	07 54 53 52 01 01 12 08 17 35 20 14 36 10 Node
7	08 07 18 18 39 59 27 08 62 34 19 24 14 45 Node
8	09 15 25 30 22 36 16 17 62 61 30 32 28 28 Node
9	10 29 37 23 18 37 15 35 34 61 59 42 61 36 Node
10	11 03 36 66 35 16 41 20 19 30 59 69 74 41 Node
11	12 05 25 49 39 23 69 14 24 32 42 69 49 43 Node
12	13 26 55 80 33 18 20 36 14 28 61 74 49 32 Node
13	14 05 19 32 69 37 64 10 45 28 36 41 43 32 Terminal

Quality = 50 + signal margin in dB

(b) Signal strength between nodes



(a) Topology of network configuration 13

Net members:	#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	Role:
0	01		00	00	00	50	00	00	00	00	50	00	00	00	00	Node
1	02	00		00	50	00	00	00	00	50	00	00	50	00	00	Node
2	03	00	00		00	00	00	00	00	00	00	00	00	00	50	Node
3	04	00	50	00		00	00	00	00	50	00	00	00	00	50	Node
4	05	50	00	00	00		00	00	00	00	00	00	00	50	00	Node
5	06	00	00	00	00	00		50	50	00	00	00	00	00	00	Node
6	07	00	00	00	00	00	50		00	00	00	00	50	50	00	Node
7	08	00	00	00	00	00	50	00		00	00	00	00	00	00	Node
8	09	00	50	00	50	00	00	00	00		00	00	50	00	00	Node
9	10	50	00	00	00	00	00	00	00	00		00	00	00	00	Node
10	11	00	00	00	00	00	00	00	00	00	00		00	00	50	Node
11	12	00	50	00	00	00	00	50	00	50	00	00		00	00	Node
12	13	00	00	00	00	50	00	50	00	00	00	00	00		00	Node
13	14	00	00	50	50	00	00	00	00	00	00	50	00	00		Terminal

Quality = 50 - number of resend

(b) Signal strength between nodes

Bibliography

[1] Beck, A.: Simulation: the practice of model development and use. *Journal of Simulation*, vol. 2, no. 1, pp. 67–67, 2008. ISSN 1747-7778.

[2] Belding-Royer, E.M. and Perkins, C.E.: Evolution and future directions of the ad hoc on-demand distance-vector routing protocol. *Ad hoc networks*, vol. 1, no. 1, pp. 125–150, 2003. ISSN 1570-8705.

[3] Bembe, M., Abu-Mahfouz, A., Masonta, M. and Ngqondi, T.: A survey on low-power wide area networks for iot applications. *Telecommunication Systems*, vol. 71, no. 2, pp. 249–274, 2019. ISSN 1018-4864.

[4] Jagdale, B., Patil, P., Lahane, P. and Javale, D.: Analysis and comparison of distance vector, dsdv and aodv protocol of manet. *International Journal of Distributed and Parallel Systems*, vol. 3, 04 2012.

[5] Chan, L.: Hierarchical routing protocols for wireless sensor network: a compressive survey. *Wireless Networks (10220038)*, vol. 26, no. 5, pp. 3291–3315, July 2020. ISSN 10220038.

[6] Fatang, C., Shuqi, Z. and Hui, Z.: Analysis and implementation of nb-iot random access process. *Dianzi Jishu Yingyong*, vol. 44, no. 2, pp. 75–79, 2018. ISSN 0258-7998.
Available at: `https://doaj.org/article/a5e0632d04ae474b9407a2b841cfe02f`

[7] Barrett, J., Jayaraman, B., Patel, D. and Skolnik, J.: A sas-based solution to evaluate study design efficiency of phase i pediatric oncology trials via discrete event simulation. *Computer methods and programs in biomedicine*, vol. 90, pp. 240–50, 07 2008.

[8] Reynders, B. and Pollin, S.: Chirp spread spectrum as a modulation technique for long range communication. In: *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, pp. 1–5. Nov 2016. ISSN null.

[9] Clausen, T.H., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A. and Viennot, L.: Optimized link state routing protocol (olsr). 01 2003.

[10] Page B., K.W.: *The Java Simulation Handbook*. 1st edn. Shaker Verlag, D-52018 Aachen, 2005. ISBN 3-8322-3771-2.

[11] Dhanalakshmi, N.: Efficient energy conservation in manet using energy conserving advanced optimised link state routing model. *International Journal of Parallel, Emergent & Distributed Systems*, vol. 31, no. 5, pp. 469–481, September 2016. ISSN 17445760.

[12] Durand, T.G.: Evaluation of next-generation low-power communication technologies to replace gsm in iot-applications. 2018.

[13] Commission, F.C.: Wireless operations in the 3650-3700 mhz band; rules for wireless broadband. *DCC 05-56*, 2005.

[14] Perkins, C. and Royer, E.: Ad-hoc on-demand distance vector routing. In: *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100. IEEE, 1999. ISBN 0769500250.

[15] Garrido, J.M.: Practical process simulation using object-oriented techniques and c++. 1998.

[16] Goebel, J., Krzesinski, A. and Page, B.: The discrete event stimulation framework desmo-j and its application to the java-based simulation of mobile ad hoc networks. 2012.

[17] M, M.: Stability in routing: Networks and protocols. 2001.

[18] Gueguen, C.: Link state opportunistic routing for multihop wireless networks. *Wireless Networks (10220038)*, vol. 25, no. 7, pp. 3983–3999, October 2019. ISSN 10220038.

[19] Cuomo, A., Rak, M. and Umberto, V.: Process-oriented discrete-event simulation in java with continuations:quantitative performance evaluation. pp. 87–96. 01 2012.

[20] Karnon, J., Stahl, J., Brennan, A., Caro, J.J., Mar, J. and MÃ¶ller, J.: Modeling using discrete event simulation: A report of the ispor-smdm modeling good research practices task force 4. *Medical Decision Making*, vol. 32, no. 5, pp. 701–711, 2012. ISSN 0272-989X.

[21] Sinha, R.S., Wei, Y. and Hwang, S.-H.: A survey on lpwa technology: Lora and nb-iot. *ICT Express*, vol. 3, no. 1, pp. 14–21, 2017. ISSN 2405-9595. Available at: `https://doaj.org/article/178ccad4644b4027914f02fb9f49b824`

[22] Sundaram, J.P.S., Du, W. and Zhao, Z.: A survey on lora networking: Research problems, current solutions and open issues. *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2019. ISSN 1553-877X.

[23] Matloff, N.: Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science*, vol. 2, 01 2008.

[24] Ngo, C.T.: Tlsr: A tree link state routing protocol using message aggregation based on a skewed wait time assignment for infrastructure-based mobile ad hoc networks. *Computer Communications*, vol. 74, pp. 87–101, January 2016. ISSN 01403664.

[25] Perkins, C.E.: Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM Computer Communication Review (ACM Digital Library)*, vol. 24, no. 4, pp. 234–245, October 1994. ISSN 01464833.

[26] Raza, U., Kulkarni, P. and Sooriyabandara, M.: Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. ISSN 1553-877X.

[27] Lozano, A., Caridad, J., Paz, J.D. and Bajo, J.: Smart waste collection system with low consumption lorawan nodes and route optimization. *Sensors*, vol. 18, no. 5, p. 1465, 2018. ISSN 14248220.
Available at: `http://search.proquest.com/docview/2108721493/`

[28] Mekki, K., Bajic, E., Chaxel, F. and Meyer, F.: A comparative study of lpwan technologies for large-scale iot deployment. *ICT Express*, vol. 5, no. 1, pp. 1 – 7, 2019. ISSN 2405-9595.
Available at: `http://www.sciencedirect.com/science/article/pii/S2405959517302953`

[29] Huo, P., Yang, F., Luo, H., Zhou, M. and Zhang, Y.: Distributed monitoring system for precision management of household biogas appliances. *Computers and Electronics in Agriculture*, vol. 157, pp. 359–370, 2019. ISSN 0168-1699.

[30] Gu, F., Niu, J., Jiang, L., Liu, X. and Atiquzzaman, M.: Survey of the low power wide area network technologies. *Journal of Network and Computer Applications*, vol. 149, 2020. ISSN 1084-8045.

[31] Tanenbaum, A.S.: *Computer networks*. 4th edn. Prentice Hall/PTR, Upper Saddle River, N.J., 2003. ISBN 0130384887.

[32] Uthra, R. and Raja, S.V.K.: Qos routing in wireless sensor networks-a survey. *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–12, 2012. ISSN 0360-0300.

[33] Stallings, W.: A practical guide to queuing analysis. *Byte*, vol. 16, no. 2, pp. 309–, 1991. ISSN 0360-5280.

[34] Van Staden, T.: Investigation into the optimization of low speed communication protocols for narrow band networks. 2012-12.

[35] Kleinrock, L.: Queuing systems: L. kleinrock, wiley, 1975, 416 pp. *Advances in Mathematics*, vol. 22, no. 2, pp. 267–267, 1976. ISSN 0001-8708.

[36] Wolhuter, R.: The determining of optimum protocol strategies for half-duplex telemetry communication links. 2002.

[37] Spreeth, G.: Design of a low power wireless sensor network for environment monitoring. 2008-12.

[38] Hua, M., Wang, Y., Zhang, Z., Li, C., Huang, Y. and Yang, L.: Power-efficient communication in uav-aided wireless sensor networks. *IEEE communications letters*, vol. 22, no. 6, pp. 1264–1267, 2018. ISSN 1089-7798.

[39] Adamo, F., Attivissimo, F., Carducci, C.G.C. and Lanzolla, A.M.L.: A smart sensor network for sea water quality monitoring. *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2514–2522, 2015.

[40] Thorstensen, B., Syversen, T., BjÃ¸rnvold, T.-A. and Walseth, T.: Electronic shepherd - a low-cost, low-bandwidth, wireless network system. MobiSys '04, pp. 245–255. ACM, 2004. ISBN 9781581137934.

[41] Sevcik, P., ZAK, S. and Hodon, M.: Wireless sensor network for smart power metering. *Concurrency and Computation: Practice and Experience*, vol. 29, no. 23, p. e4247, 2017. E4247 cpe.4247.

[42] Wotherspoon, J.B.: Design of a multi-hop ad-hoc outdoor wireless sensor network. 2019.