Virtualisation of media access control protocol forlarge scale internet of things

Prosper Sotenga

VIRTUALISATION OF MEDIA ACCESS CONTROL PROTOCOL FOR LARGE SCALE INTERNET OF THINGS

by

Prosper Zanu Sotenga

Submitted in partial fulfillment of the requirements for the Degree

DOCTOR OF ENGINEERING (DEng): ELECTRICAL ENGINEERING

Department of Electrical Engineering

Tshwane University of Technology

&

DOCTOR OF PHILOSOPHY (PhD): COMPUTER SCIENCE

École Doctorale

Mathématiques, Sciences et Techniques de l'Information et de la Communication

Université Paris-Est Créteil Val de Marne

Defended on: 8 December 2022

Jury

Samia BOUZEFRANE	Conservatoire National des Arts et Métiers, France	Chairperson
Jalel BEN OTHMAN	L2S, CentraleSupélec, France	Examiner
Yanxia SUN	University of Johannesburg, South Africa	Examiner
Eric MATSON	Purdue University, USA	Examiner
Nadjib AITSAADI	Université de Versailles Saint-Quentin-en-Yvelines, France	Rapporteur
Jules-Raymond TAPAMO	University of KwaZulu-Natal, South Africa	Rapporteur
Karim DJOUANI	Université Paris-Est Créteil, France	Supervisor
Anish KURIEN	Tshwane University of Technology, South Africa	Co-supervisor

DECLARATION

I hereby declare that this thesis submitted for the degree DEng in Electrical Engineering, at the Tshwane University of Technology, Pretoria, South Africa, and the PhD in Computer Science at Université Paris-Est Créteil Val de Marne, Paris, France, is my own original work and has not previously been submitted to any other institution of higher education for any degree or examination. I further declare that all sources cited or quoted are indicated and acknowledged by means of a comprehensive list of references.

Prosper Zanu Sotenga

15 December 2022

DEDICATION

This work is dedicated to my wonderful family for their support, prayers, and encouragement: To my supportive parents Mr. John Sotenga and Mrs. Susana Sotenga and my adorable siblings Timinu, Awah and Alena, I thank you all.

ACKNOWLEDGEMENTS

My sincere acknowledgement goes to the Lord almighty for his guidance throughout this journey and to my family for the moral support they have provided.

I thank and acknowledge my supervisors for the technical guidance and general support in all valuable areas of this work. Prof. Karim Djouani has been very helpful in providing guidance for the successful completion of this study. Prof. Anish Kurien, whose role has also made an impact on the success of this thesis.

I would also like to thank the Tshwane University of Technology (TUT), Université Paris-Est Créteil Val de Marne (UPEC), the French South African Institute of Technology (F'SATI), the National Research Foundation (NRF) and the Telkom Centre of Excellence programme for the opportunity, facilities and financial support in carrying out this study.

ABSTRACT

To support the massive deployment of heterogeneous devices in the Large-Scale Internet of Things (LS-IoT), several novel enhancements have been proposed to improve the scalability of the network with an emphasis on concurrent transmissions. Some of these enhancements include spatial re-usability with directional antennas, sectorisation, frame aggregation and orthogonal multi-channels. The Gateway (GW) device in the network plays an important role in managing media access to support such improvements. Without a scalable and robust GW Media Access Control (MAC) component, the underlying improvements may be fruitless due to functional and resource constraints in using a dedicated GW MAC as in existing LS-IoT GWs. In this study, the aim is to improve the scalability of the GW MAC component to support the massive deployment of heterogeneous devices by employing virtualisation techniques to augment the GW MAC as virtual instances deployed on an edge data centre (eDC). To achieve this, firstly the impact of a dedicated GW MAC component on the media access with multiple concurrent transmissions referred to as virtual Access Groups (vAG) is modelled and evaluated. The results suggest that the throughput using a dedicated GW MAC is degraded with multiple vAGs. Secondly, a suitable virtualisation network with Software Defined Networking (SDN) capabilities for instantiating and deploying the GW MAC component over a set of distributed eDCs is proposed and modelled. The results suggest that the throughput can be improved when the GW MAC is virtualised using the proposed framework as opposed to using a dedicated localised GW MAC component. The third objective is to propose an effective strategy to minimise the virtualisation delay and further improve the robustness for dynamic virtual GW MAC requests (dynamic load conditions) to ensure that the timing deadlines are met for an improved throughput. The results show that the virtualisation delay is minimised and the MAC throughput versus the number of devices is improved especially for high IoT traffic rates when compared to suboptimal approaches and the dedicated localised GW MAC approach.

RESUMÉ

Pour prendre en charge le déploiement massif d'appareils hétérogènes dans l'Internet des objets à grande échelle (LS-IoT), plusieurs nouvelles améliorations ont été proposées pour améliorer l'évolutivité du réseau en mettant l'accent sur les transmissions simultanées. Certaines de ces améliorations incluent la réutilisation spatiale avec des antennes directionnelles, la sectorisation, l'agrégation de trames et les multicanaux orthogonaux. Le dispositif de passerelle (GW) du réseau joue un rôle important dans la gestion de l'accès aux médias pour prendre en charge de telles améliorations. Sans un composant évolutif et robuste de contrôle d'accès au support (MAC) GW, les améliorations sous-jacentes peuvent tre vaines en raison de contraintes fonctionnelles et de ressources liées à l'utilisation d'un MAC GW dédié comme dans les G-W LS-IoT existants. Dans cette étude, l'objectif est d'améliorer l'évolutivité du composant GW MAC pour prendre en charge le déploiement massif d'appareils hétérogènes en utilisant des techniques de virtualisation pour augmenter le GW MAC en tant qu'instances virtuelles déployées sur un centre de données Edge (eDC). Pour ce faire, tout d'abord l'impact d'un composant GW MAC dédié sur l'accès au média avec plusieurs transmissions simultanées appelées groupes d'accès virtuels (vAG) est modélisé et évalué. Les résultats suggèrent que le débit utilisant un MAC GW dédié est dégradé avec plusieurs vAG. Deuxièmement, un réseau de virtualisation approprié avec des capacités de mise en réseau définie par logiciel (SDN) pour instancier et déployer le composant GW MAC sur un ensemble d'eDC distribués est proposé et modélisé. Les résultats suggèrent que le débit peut tre amélioré lorsque le GW MAC est virtualisé à l'aide du cadre proposé par opposition à l'utilisation d'un composant GW MAC localisé dédié. Le troisième objectif est de proposer une stratégie efficace pour minimiser le délai de virtualisation et améliorer encore la robustesse des requtes dynamiques virtuelles GW MAC (conditions de charge dynamiques) afin de garantir le respect des délais de synchronisation pour un débit amélioré. Les résultats montrent que le délai de virtualisation est minimisé

et que le débit MAC par rapport au nombre d'appareils est amélioré, en particulier pour les taux de trafic IoT élevés par rapport aux approches sous-optimales et à l'approche MAC GW localisée dédiée.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS AND PSEUDO CODES

GLOSSARY

3GPP 3rd Generation Partnership Project

ACB Access Class Baring

API Application Programming Interface

BI Beacon Interval

BO Back-Off

BS Base Station

CCC Common Control Channel

CCRP Cloud Computing Resource Pool

COP Contention Only Period

CPRI Common Public Radio Interface

CPRIoE Common Public Radio Interface over Ethernet

CRA Collision Resolution Algorithm

CRAN Cloud Radio Access Network

CRC Cyclic Redundancy Check

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

DCF Distributed Control Function

DIFS DCF Inter-frame Spacing

DQ Distributed Queuing

DQRAP Distributed Queuing Random Access Procedure

EAB Enhanced Access Barring

eDC Edge Data Center

ETSI European Telecommunications Standards Institute

FDM Frequency Division Multiplexing

GW Gateway

IIoT Industrial Internet of Things

IoT Internet of Things

IP Internet Protocol

LPWAN Low Power Wide Area Networks

LS-IoT Large Scale Internet of Things

LTE Long-Term Evolution

LTE-M Long-Term Evolution Machine Type Communication

M2M Machine-to-Machine

MAC Media Access Control

MEC Multi-access Edge Computing

MIPS Million Instructions Per Second

NAM Network Activity Monitor

NB-IoT Narrowband Internet of Things

NFV Network Function Virtualisation

NFV-MANO NFV Management and Orchestration

NFVI Network Function Virtualisation Infrastructure

NP Notification Period

NV Network Virtualisation

OFDMA Orthogonal Frequency-Division Multiple Access

PDM Placement Decision Maker

pGW Physical Gateway

PHY Physical Layer

PRACH Physical Random Access Channel

PS-POLL Power Saving Poll

QNA Queuing Network Analyser

QoS Quality of Service

RA Random Access

RAM Random Access Memory

RAN Radio Access Network

RAO Random Access Opportunity

RAR Random Access Response

RAW Restricted Access Window

RF Radio Frequency

RFTDMA Random Frequency and Time Division Multiple Access

RRU Remote Radio Unit

S1G Sub 1 GHz

SDN Software Defined Networking

SDR Software Defined Radio

SFSR Single-feature Single-Request

SIFS Short Inter-frame Spacing

TDM Time Division Multiplexing

TDMA Time Division Multiple Access

TIM Traffic Identification Map

TOP Transmission Only Period

TXOP Transmit Opportunity

UNB Ultra Narrow-Band

VIM Virtualised Infrastructure Manager

vMAC Virtual Media Access Control

VNF Virtual Network Function

VNO Virtual Network Operator

WLAN Wireless Local Area Network

WRR Weighted Round Robin

WSRP Wireless Spectrum Resource Pool

CHAPTER 1. INTRODUCTION

This chapter presents the background of the research topic and motivates the study. The main research problem to be addressed and the associated sub-problems are also indicated followed by the methodology employed in the study. The delimitations, the major contributions of the work and the overview of the subsequent chapters are also highlighted in this chapter.

1.1 Background and Motivation

1.1.1 The Growth of Machine-to-Machine Devices in Internet of Things

Machine-to-Machine (M2M) devices need to be capable of being deployed on large scale. An example use-case that characterises high growth in M2M communication is the use of up to 30000 M2M devices in a smart metering infrastructure for periodic remote logging of energy consumption [3]. Moreover, several studies have found that the total number of M2M devices connected to the internet will reach 50 to 100 billion by the year 2020 [4]. As evident from Figure 1.1, as of the year 2008, the number of connected devices surpassed the number of active users per global population [1]. Such growth in M2M devices connected to the internet is driven by several factors. In recent times, the tremendous advancement in sensing capabilities, control systems, cloud computing, processing and storage has led to a more evolved world, where our day-to-day activities have become more autonomous, seamless and intelligent. The application of these technologies has also grown staggeringly with examples of innovative use cases being; agriculture monitoring and control, building automation, assisted living for the aged and disabled, supply chain management, vehicle monitoring, smart cities, infrastructure maintenance and many more. These applications are made possible through M2M communication supported by the Internet. As a result, more ubiquitous IoT devices that form M2M communication

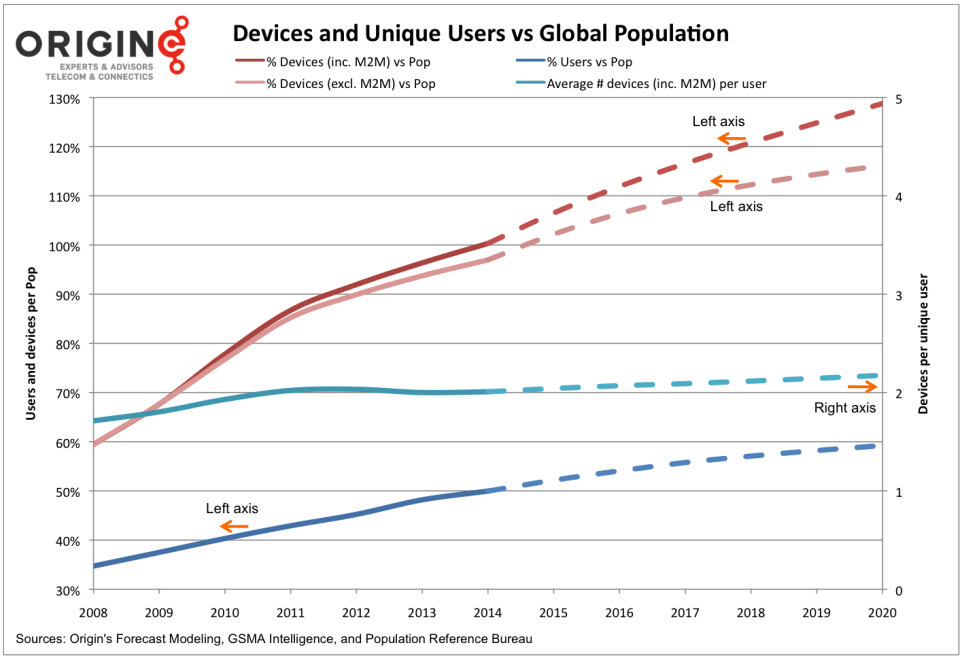

Figure 1.1: A graphical representation of the number of devices connected to the internet and the population of people in the world [1].

will be deployed in the future which will stimulate the increase of the co-existence of M2M devices with the current population of active Internet users. Additionally, the growth in M2M communication is triggered by the continuous reduction in the cost of sensors, microcontrollers, microprocessors, actuators and radio frequency chips that are used for M2M communication [5]. In essence, there is a rapid growth in M2M devices in the Internet of Things (IoT) network infrastructure which makes it critical to characterise IoT as a massive or large-scale network to adequately address the issues associated with it.

1.1.2 Effects of Exponential Growth of M2M Devices on the IoT Network

The exponential growth of M2M devices results in a densely populated network of devices which primarily affects the Access Network component of the IoT network. In general, the Physical (PHY) and the Media Access Control (MAC) layers of the wireless access network are critical in managing the effects of a large number of devices in an IoT network. As a result, the PHY and MAC layers have been improved, leading to the development of various Large-Scale IoT (LS-IoT) technologies. The massive interconnection of M2M devices poses a significant chal-

lenge to the radio resources at the PHY layer due to constraints on bandwidth and antennas for the transmission of data generated by a massive number of devices. Using the Ultra Narrow-Band (UNB) technique as with the SigFox technology enables multiple transmissions using a single carrier frequency over a limited bandwidth to accommodate several device transmissions at the PHY layer [6]. Whereas the use of directional antennas for spatial re-usability coupled with improved modulation techniques increases the probability of concurrent multiplexing [7] at the PHY layer as with the IEEE 802.11ah technology. However, the UNB limits the network to devices with very low data rate or small uplink traffic [8] which raises concerns about the heterogeneity of the network in terms of supporting dynamic high data rate applications in conjunction with low data rate LS-IoT applications. Moreover, the concurrent multiplexing of signals based on spatial re-usability for a massive number of devices could lead to excessive traffic which can cause congestion at the IoT Gateway (GW) and hence reduced reliability in terms of packet drops as a result of buffer overflows. At the MAC layer, a large number of M2M devices associated with an IoT GW leads to increased contention for the wireless medium in a contention-based channel access scheme. This, therefore, increases the probability of collisions hence high transmission errors. Likewise, in a non-contention based channel access scheme, the combined polling interval increases as the number of M2M devices scale up which could cause starvation of services to other devices hence increasing the network latency in the entire IoT network. The use of MAC enhancements such as the group synchronised media access protocols as with the IEEE 802.11ah technology [2] as well as the Access Class Baring (ACB) Random Access (RA) procedure for Narrow Band IoT (NB-IoT) and Long Term Evolution Machine Type Communication (LTE-M) [9] helps to tackle congestion of the MAC. However, these approaches may increase the network latency and packet drops for a large number of devices since they require high signalling and processing at the MAC layer of the GW device. Although the PHY and MAC enhancements maintain QoS requirements to some extent, industries and standardization bodies still emphasise that media access for scenarios with a very large number of devices has the potential to become a bottleneck [5]. This kind of phenomenon could potentially affect Quality of Service (QoS) requirements in the large-scale IoT network. Therefore this work focuses on the MAC component of the GW node given the extremely high

processing, signalling and scheduling by the GW MAC to maintain an effective MAC protocol for a very large number of M2M devices.

1.1.3 The Need for Large-Scale IoT Gateway MAC Enhancements

The PHY layer scalability and throughput gains of the different LS-IoT technologies become fruitless if the GW MAC can not scale according to the increased M2M transmissions associated with the increased number of M2M devices. In essence, the effectiveness of the GW MAC of the large-scale IoT network may be affected by relatively constrained resources (bandwidth/time, Buffer size, Processing). This may cause MAC frame drops hence more retransmission and increased probability of collision or polling delays. To support the MAC enhancements and as well the heterogeneous nature of the technology, the GW MAC needs to be robust due to the increased signalling and management of device transmissions by the GW MAC component. The GW MAC needs to support concurrent multiplexing and scheduling for scalability to reduce the bottleneck. Thus, concurrent processing and transmission of GW MAC data frames and Control frames are imperative for enhancing massive M2M device scheduling for both distributed and point scheduling. Therefore, this study argues that augmenting the GW MAC component has the potential to support multiple concurrent transmissions over a media access control scheme supported by an underlying PHY layer capable of equally guaranteeing high concurrency through concepts such as multichannel concurrent transmissions, beamforming, sectorisation etc. The ultimate result of external virtualisation is the optimisation of resource utilisation and an increase in the whole systems performance [10]. The authors in [11], indicate how virtualisation of the media access network in future mobile networks could deal with the capacity demand. As such, this study looks into the external virtualisation concept by harnessing the benefits of Network Virtualisation (NV) and Software Defined Networking (SDN) to augment the execution of the GW MAC and minimise associated delays, congestion and frame drop to support media access scalability.

1.2 Problem Statement

The GW processes frame transmissions to support media access based on a set of MAC functions which are imperative for the throughput and scalability performance of the network. While several studies focus on enhancing the MAC protocol design and improving the PHY layer concurrency, the ability of the GW MAC component to support a massive number of simultaneously and heterogeneously connected devices is a challenge due to the functional and resource constraints required for executing the MAC protocol functions to support the throughput enhancements. Therefore, there is a need to provide an effective approach to support the scalability of the media access in the LS-IoT network through the augmentation of the GW MAC component based on the emerging network virtualisation approach to further deal with resource constraints for executing the related GW MAC functions.

1.3 Sub-Problems

The following sub-problems are deduced based on the problem statement presented.

1.3.1 Sub-Problem 1

The scalability enhancements of the LS-IoT network which are achieved by managing concurrent media access using various techniques proposed in literature such as Radio Frequency (RF) virtualisation and sectorisation impose a significant load on the GW MAC component. As a result, the classical approach of having a dedicated GW MAC component for managing multiple concurrent transmissions poses a challenge due to the buffer and processing constraints of the classical GW MAC component. The impact thereof needs to be adequately established to propose a suitable approach for addressing the GW MAC constraints to support the media access scalability enhancements.

1.3.2 Sub-Problem 2

To address the potential impact of the GW MAC constraints, the creation and remote deployment of virtual GW MAC instances for a LS-IoT GW is proposed in this study. However, this requires a virtual network infrastructure made up of different nodes and elements running different processes. These virtual network elements may lead to overheads and increased latency in the transmission and processing of MAC frames virtually. However, it is a challenge to estimate the latency imposed by the virtualisation processes due to the complexity of the virtual network infrastructure. Therefore, there is a need to propose and develop a suitable model to evaluate the effects of offloading the MAC frames through a virtual network. This can be achieved by employing an appropriate queuing network model which synthesises the performance factors of each virtualisation network element and process.

1.3.3 Sub-Problem 3

When virtualising the GW MAC component, the requests are made dynamically and the edge Data Centre (eDC) nodes are characterised by varying resource levels and response times. These characteristics have a direct impact on achieving the scalability needed when the GW MAC instances are to be deployed virtually. Thus, if the eDC is not selected efficiently, it could affect the timing deadlines of various MAC functions and frame transmissions. Therefore, there is a need to propose an effective placement strategy to effectively and dynamically find the most suitable eDC for running the virtual MAC (vMAC) instances such that the total virtualisation latency due to the eDC resource levels and response time is minimised. This may be achieved by proposing a vMAC placement strategy using a suitable heuristic optimisation approach to dynamically select the best host for running the MAC instances based on the vMAC instance resource requirements and the eDC resource availability.

1.4 Research Methodology

The following summarised methodology is used to conduct this research study;

a. **Phase 1:** A comprehensive literature survey is conducted on existing studies regarding existing MAC scheduling protocols in LS-IoT, the GW MAC constraints in LS-IoT, Virtual network models and edge/cloud-based virtualisation for solving the scalability problems. This phase spans through to the conclusion of this study.

b. **Phase 2:** The second phase of this work is to study the throughput performance of a classical contention-based MAC scheduling of M2M devices and evaluate the potential impact of the GW MAC constraints on the scheduling of M2M devices in a large scale scenario. This is done by studying a MAC scheduling model based on the 802.11ah standard for an unsaturated network, which is modified to introduce the GW MAC buffer and processing constraints. A software-based numerical computing tool is used to evaluate the model and generate the results for analysis. The results obtained from the experimental evaluation are observed and analysed to ascertain the behavioural dynamics and provide verification.

c. **Phase 3:** The third phase of the study is to propose a conceptual framework and develop an analytical model which appropriately represents the external virtualisation network nodes and processes to establish the GW MAC virtualisation delay. This is done by using the Queuing Network Analyser method to synthesise the different virtual network components and processes to arrive at the average virtualisation delay. A software-based numerical computing tool is used to evaluate the model and generate the results for analysis. The results obtained for the experimental evaluation are analysed and compared with a simulation model for validation and verification.

d. **Phase 4:** The fourth phase of the study is to develop a vMAC placement strategy to minimise the latency and meet the GW MAC virtualisation resource requirements. This is achieved by designing a placement framework for the SDN controller and formulating a minimisation problem for the virtualisation delay and solving the problem using a heuristic algorithm. A software-based numerical computing tool is used to evaluate the solution and generate the results for analysis. The results obtained from the experimental evaluation are observed and analysed.

1.5 Delimitations

The main delimitations of this research are outlined and motivated below:

a. This study is limited to the IEEE 802.11ah MAC protocol. The technology provides a suitable functionality for the concept proposed in this work in comparison with the LoRa, SigFox, NB-IoT and LTE-M technologies. The IEEE 802.11ah represents a more heterogeneous LS-IoT network as it supports relatively high and low data rates for a large number of devices to access the channel over a much larger bandwidth than the LoRa, SigFox, NB-IoT and LTE-M technologies. It also allows for concurrent transmission over antenna sectorisation. A larger bandwidth over sectorisation allows for the creation of several concurrent virtual access groups using RF virtualisation. This makes it possible and practical for the protocol to be modelled based on the creation of concurrent media access groups over RF virtualisation mapped to several vMAC instances. In addition, the sectorisation and restriction of access to a group of devices in the IEEE 802.11ah MAC help alleviate the hidden terminal problem for a large number of devices. Hence it simplifies the model.

b. The study is limited to an unsaturated network with four different LS-IoT traffic rates characterising different LS-IoT applications. The unsaturated network allows for the modelling of the MAC throughput according to the different device traffic rates which is essential for considering the heterogeneity of the LS-IoT network.

c. This study is limited to external virtualisation within the eDC based on the SDN and Network Function Virtualisation (NFV) framework. Since the aim is to ease congestion in the GW MAC, external virtualisation helps to offload the GW MAC related function from the GW. Using the edge instead of the cloud helps with supporting the time sensitive GW MAC signalling that is required to ensure the strict GW MAC timing and QoS requirements in LS-IoT are met. In addition, edge computing allows for improved security and reliable transmission for the virtualisation of the GW MAC than the cloud platform.

d. The proposed vMAC placement strategy is limited to a heuristic approach as it provides a faster and more practical solution for ensuring that the MAC timing deadlines are met to guarantee QoS.

e. The main performance factors analysed in this work are limited to the virtualisation delays (including the node response time and virtual network response time), the media access throughput and transmission probabilities. This is because latency and throughput are classical performance factors for assessing the scalability of the network relative to the number of devices. Moreover, the MAC technology of the IEEE 802.11ah technology is based on random channel access and therefore its performance is easily characterised by the MAC throughput and transmission failure probabilities.

f. This study is executed using a combination of analytical models and simulation models for validation and verification. The modelling considers the key network and virtualisation components essential to mimic the practical performance factors of concern in this work and a simulation model is used to validate the analytical model.

Other specific delimitations and assumptions on the different sub-problems are highlighted in the subsequent related chapters.

1.6 Contributions and Outputs

The main contributions of this study are presented below:

a. A throughput model for an unsaturated network based on the 802.11ah Restricted Access Window (RAW) and the Power Saving Polling (PS-POLL) mechanism for large-scale IoT networks is studied and modified to incorporate the blocking probability of a GW MAC frame and the processing delay of the GW MAC component. The model also incorporates the MAC scheduling scheme based on virtual access groups. To the best of the author's knowledge, no work has been conducted in literature on modelling the

802.11ah MAC throughput performance based on multiple virtual access groups, GW MAC buffer constraints and the GW MAC processing capacity.

b. A virtual network framework is proposed and developed based on the Common Public Radio Interface (CPRI) encapsulation process, SDN Priority-based traffic classification and switching and the eDC resource provisioning. A queueing network approach (Queuing Network Analyser (QNA)) is used to model the virtual network node utilisation, node response time, network response time and the virtualisation delay for the GW MAC virtualisation process. The virtual network model performance is incorporated in the media access throughput model to evaluate the impact of the virtualisation model. To the best of the author's knowledge, no work has been conducted in literature on modelling the 802.11ah MAC throughput performance based on a QNA, incorporating the CPRI process, traffic classification process, priority switching and the eDC resource provisioning.

c. A SDN and NFV-based application is proposed which incorporates a strategy for network activity monitoring and vMAC placement decisions. The placement decision making is further enhanced using a heuristic algorithm which short-lists the eDC nodes, sorts the eDC nodes and selects the best eDC node for the vMAC placement, taking into consideration the eDC and virtual network constraints. To the best of the author's knowledge, no work has been conducted in literature on the optimisation of the virtual network for virtualising the GW MAC functions of the 802.11ah standard based on the virtual access groups to support the scalability of the LS-IoT network.

A list of research outputs generated from the results of this work during this study is presented in Appendix D.

1.7 Outline of Thesis

The remainder of the thesis is outlined as follows: Chapter 2 focuses on the literature review conducted on MAC in LS-IoT network, related works on GW MAC constraints, Virtual Network models and Virtual Network placement strategies. In Chapter 3, the Virtual Media Access

throughput performance with GW MAC constraints is modelled and analysed. Chapter 4 presents the proposed virtual network model and the results are analysed. In Chapter 5, the proposed vMAC placement approach is presented and analysed. Chapter 6 concludes the study by summarising the main points and the recommendations for future work are also proposed.

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

In this chapter, an overview of the MAC protocol and a review of the MAC protocols used in the various LS-IoT networks are presented. The associated challenges in LS-IoT networks that necessitate the need to further enhance the GW MAC scalability are also presented. Based on the main aspects of this study, a review of some related work around the GW MAC constraints, the virtualisation network models and Virtual Network Function (VNF) placement strategies are also presented. An extended literature study on LS-IoT network characterisation and the most current approaches aimed at enhancing the MAC are presented in Appendix C.

2.1.1 Overview of Media Access Control Protocols

One of the core operations of a wireless access network is the management of the transmission medium for multiple devices by employing multiple access techniques to ensure that the constrained transmission resources are effectively utilised for a successful transmission [12], [13]. The multiple access procedure also called the MAC scheduling protocol [14] resides in the MAC layer which is based on defined rules that indicate how the physical transmission resources are shared amongst the devices in the network. The MAC relies on the PHY layer's multiplexing mechanism where multiple signals from different devices are multiplexed onto a constrained spectrum or multiple spectra. The MAC protocol can be seen as the scheduling of frames from multiple M2M devices implemented by a scheduler which performs the assignment of transmission time and channels for a frame transmission from all devices distributively or centrally [15].

As depicted in Figure 2.1, the MAC protocols can be divided into contention-based protocol-

s, contention-free based protocols, and hybrid-based protocols [16], [17]. In contention-based protocols, devices randomly attempt media access which exposes the devices to possible simultaneous media access. However, contention-based protocols employ a Collision Resolution Algorithm (CRA) to ensure simultaneous media access is resolved. CRAs are classically employed distributively whereby devices execute randomised re-transmissions for media access schedules or reservations. This concept is illustrated in Figure 2.1 (a) whereby device 1 has some data ready to be transmitted. However, the device senses a busy medium until a time when the medium is free. When the medium is free, the device then waits for a fixed duration called an Inter-Frame Space (IFS) plus a random back-off delay time. This process randomly schedules device 1 for transmission. However, if the transmission is unsuccessful, the random back-off delay is computed again until a successful transmission occurs. The same process is executed by device 2. The contention-based scheduling is widely known for its ease and flexibility of implementation as well as reduced overhead [18]. However, it cannot scale due to the high collision probability with many devices.

In contention-free based protocols, a scheduler is distributively or centrally employed to allocate the communication resource. In contention free-based MAC scheme, a device may be permanently or dynamically allocated some communication resource. Devices only use their allocated communication resource to transmit, which may be coordinated by a coordinating device such as a GW. The basic operation of the contention-free based approach is depicted in Figure 2.1 (b). Device 1 only transmits within one slot whereas device 2 only transmits during a different slot. During the assigned slot duration, the medium is busy for the entire transmission period. How time slots are allocated may differ depending on the network performance metrics. Contention-free based protocols avoid collisions in time and frequency. However, these schemes cannot scale since more resources are needed in proportion to the number of devices. Thus, more time slots and more channels are required.

In a typical hybrid MAC protocol, a coordinator provides synchronised time slots over a time frame for groups of devices. Each group of devices then employs a contention-based approach

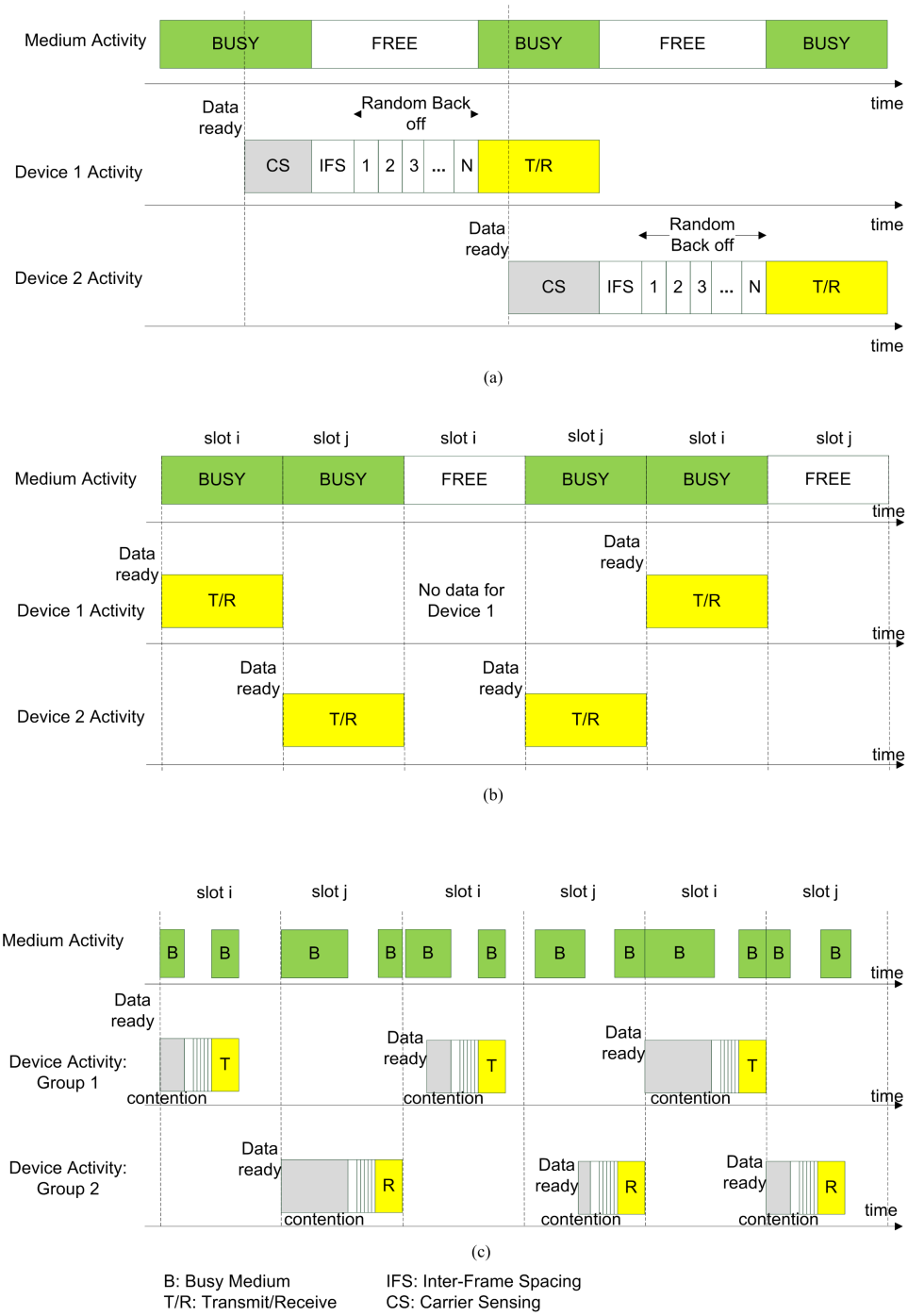

Figure 2.1: (a) Contention-based MAC Protocol. (b) Contention-free based MAC protocol. (c) Hybrid MAC protocol.

to access the media within a given time slot. This process is illustrated in Figure 2.1 (c). Device 1 which belongs to a group of devices that can contend for media access in one slot contends for media access using the contention-based approach. The busy medium depicts a transmission or reception activity either by device 1 or other devices within that slot. Likewise, device 2 also competes for channel access with a group of stations in a different slot and the slots are repeated periodically. Such hybrid approaches are sometimes called synchronous contention-based protocols [19] and are the basis for many MAC scheduling protocols in existing wireless communication protocols and standards for IoT and M2M communication.

2.1.2 Media Access Control Protocols used in LS-IoT Technologies

Wireless communication has evolved as the ultimate way to connect a large number of devices in an IoT network. Several classical and legacy wireless technologies have been enhanced by standardisation bodies and wireless communication companies such as the 3GPP, IEEE and Semtech to accommodate M2M connectivity within the IoT context. In general, these technologies are known as Low Power Wide Area Networks (LPWAN) as it specifies a number of design considerations targeted at M2M communication for a scalable IoT network of devices using low power over long-range communication. The most promising LPWAN technologies currently that define the MAC protocols for LS-IoT are the LoRa, SigFox, NB-IoT, LTE-M and IEEE 802.11ah technologies. The MAC protocols of these technologies are derived from some classical MAC protocols. In this section, the MAC protocols associated with the LPWAN technologies are discussed and a summary of the MAC of these technologies is provided in table 2.1.

The LoRaWAN protocol defines the MAC protocol of the LoRa technology which uses the un-slotted ALOHA protocol for channel access in the uplink direction. Thus, a device with data to be transmitted will transmit its data irrespective of whether the channel is idle or busy. When there is a simultaneous transmission and the devices do not receive an acknowledgement, the frame is discarded and a retransmission is initiated a certain number of times until a successful transmission occurs. However, specific to the LoRa MAC protocol, the devices choose

a random frequency channel before invoking the un-slotted ALOHA protocol and transmit frames based on a restricted transmission duty cycle [20]. A dedicated downlink window is used for transmitting downlink data upon a successful uplink transmission for class A devices whereas a scheduled downlink window using beacon frames is sent by the GW for Class B devices. Class C LoRa devices continuously listen for downlink data. Similarly, the MAC protocol of the SigFox technology also uses the un-slotted ALOHA protocol for media access. However, it operates based on the Random Frequency and Time Division Multiple Access (RFTDMA) method whereby the devices randomly select a time and UNB channel for transmitting a frame using the ALOHA protocol [6]. The devices send three messages within a given period. The LoRa and SigFox MAC protocols are proven to only be reliable with moderate traffic loads [21].

The NB-IoT and the LTE-M MAC protocols are both generally based on the existing random access procedure (RA) depicted in Figure 2.2 as used for LTE over Orthogonal Frequency-Division Multiple Access (OFDMA) channels. The main random access procedure of the NB-IoT and LTE-M technologies used is based on channel contention. This approach can be referred to as a multi-band multichannel slotted ALOHA since devices are assigned a frequency band within which the devices randomly choose a sub-channel for transmission in a synchronised time slot [22]. Contention-free-based random access is only used during handover. The contention-based random access procedure is executed when a device either wants to establish a connection, transmit uplink data, re-establish a connection or synchronise with the GW. This protocol involves a four-way negotiation process whereby a device first transmits a random preamble to the GW known as the eNodeB in LTE during a random access opportunity (RAO). The GW transmits a Random Access Response (RAR) to the device within a time frame indicating the resource block assigned to the device for the transmission of the next message [9]. If more than one device selects the same preamble during the RAO, the devices will all receive the same RAR from the GW. Hence the next message to be transmitted will collide since both devices have the same assigned resource block. Therefore, the GW executes a contention resolution algorithm for all the devices assigned to the same resource block and responds to the device

that is successful in the contention resolution. If a device does not receive a response from the GW after transmitting a preamble, it retransmits a preamble after a random back-off time. Specific to the NB-IoT and LTE technologies, the MAC protocol assigns a resource block to multiple devices for uplink transmission as opposed to the traditional LTE where a resource block is allocated to only one device.

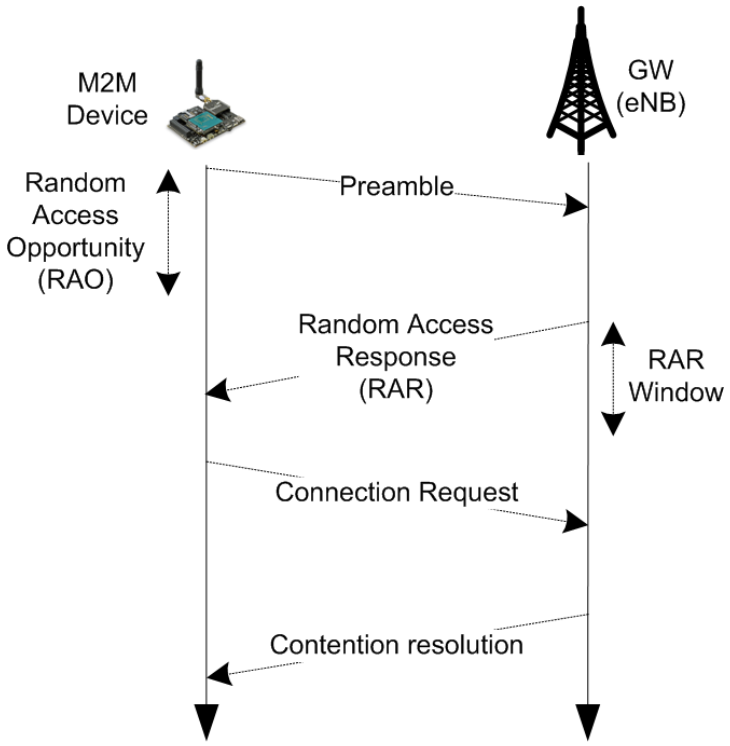

Figure 2.2: Contention-based Random Access procedure

Furthermore, the main difference between the NB-IoT and the LTE-M exists mainly in the PHY layer where the preambles of the NB-IoT are single-tone frequency hopping patterns whereas the LTE-M preambles are repeated for a certain number of times [23]. Moreover, as opposed to NB-IoT which uses a physical resource block either within a GSM carrier or a dedicated LTE carrier or a vacant resource block within the guard-band, the LTE-M uses resource blocks that are multiplexed between LTE and LTE-M devices using Frequency Division Multiplexing (FDM). In that case, LTE-M can coexist with already existing LTE infrastructure with minimal changes [24]. With both these technologies, although the GW performs contention resolution, it leads to high signalling in the case of a large number of devices. The use of ACB in NB-IoT

and LTE-M MAC protocol prevents M2M devices from triggering a random access procedure to control the load on the GW MAC. However, this approach may affect transmission latency for mission-critical IoT applications.

The IEEE 802.11ah protocol on the other hand is an amendment of the IEEE 802.11 protocol to support a large number of devices and provide an extended range to support LS-IoT. The MAC protocol of the IEEE 802.11ah uses the classical contention-based Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [2] process. Thus, the devices sense the medium within a given period and if the medium is free it sends a frame. If an acknowledgement is not received, the device enters an exponential back-off period before attempting to retransmit. A major concern when a large number of devices are to be scheduled by the MAC protocol using CSMA/CA is the increased contention for channel access. This drawback is addressed in the IEEE 802.11ah technology using the novel RAW approach shown in Figure 2.3 which restricts channel access to a predefined group of devices over a given period defined by two successive beacons. Within the restricted period, a group of devices are further allocated time slots during which devices may contend for uplink channel access using the CSMA/CA. Besides the RAW, the IEEE 802.11ah technology proposes several other MAC features to accommodate a large number of devices such as relay access to extend the transmission coverage, target wake times for reduced contention and power consumption, negotiated transmission for devices with high-speed data and sectorisation/partitioning of devices into groups supported by spatial multiplexing for concurrent transmission with reduced contention.

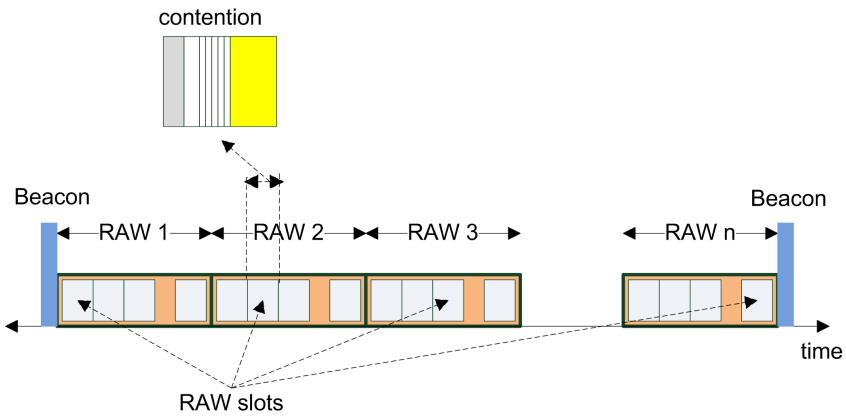

Figure 2.3: Beacon interval representation of the Restricted Access Window MAC approach.

2.1.3 LS-IoT Media Access Control Scalability Challenges

Studies such as [25] show that the very large number of M2M devices contribute to the most demanding challenges in IoT networks in general. The LS-IoT's peculiar characteristics associated with the pervasive and ubiquitous nature of M2M device deployment lead to serious media access challenges [26]. The ability of the MAC layer to effectively accommodate an increasing number of devices and resource allocation is essential. However, there are some challenges faced with achieving such MAC scalability within the context of LS-IoT. These fundamental challenges are discussed below.

2.1.3.1 Collisions

It is evident from the MAC protocols used in the LS-IoT technologies that LS-IoT technologies mostly make use of random channel access which can lead to frame collisions in a dense network. The scalability of the MAC in LS-IoT networks is significantly affected by the high collision probability in contention-based MAC schemes. The collision probability is established based on the likelihood that there will be at least one or more devices that may randomly select or compute the same back-off schedule following the detection of a free medium for a transmission attempt [27]. This likelihood certainly increases for LS-IoT networks because many devices lead to a high likelihood of transmission or reception of frames [28]. In [29], the number

Table 2.1: A summary of Media Access Control protocols used in LS-IoT Technologies

IoT Technology	**MAC Protocol**	**MAC Modifications**	**Comments. (Strengths and/or Limitations**
LoRa	Un-slotted ALOHA	Class A: Dedicated downlink window. Class B: Scheduled downlink window. Class C: Continuous listening for downlink. Restricted transmission duty cycle.	Moderate latency. Low to moderate data rate (50 kbps uplink).
SigFox	Un-slotted ALOHA	Added Frequency and Time Division Multiple Access. Restricted transmission duty cycle.	High latency. Low data rate (100 bps uplink)
NB-IoT	Slotted ALOHA	Multi-band and Multichannel access. Access class barring. Resource block sharing between M2M devices. Single-tone frequency hopping preambles.	Moderate to high latency. Moderate data rate. High signalling. Potential GW Overload.
LTE-M	Slotted ALOHA	Multi-band and Multichannel access. Access class barring. Resource block multiplexing between LTE and M2M devices using FDM	Scalable. Low latency. Moderate data rate. High signalling and Potential GW overload and high GW MAC processing.
IEEE 802.11ah	CSMA/CA	RAW, TWT, Sectorisation, GW MAC relays, High-speed data negotiation, Reduced frame formats	Low collisions, power saving. Low to high data rate (heterogeneous traffic). High signalling and high GW MAC processing.

of contending devices is modelled as a function of the probability of a collision based on a classical contention-based scheduling method, Distributed Coordination Function (DCF). The relationship between the number of contending devices and the conditional collision probability is illustrated in Figure 2.4 which shows the rapid increase in the conditional collision probability as the number of contending devices increases for the different minimum contention window CW_{min} values. When collisions become highly probable, delays emanate, and to some extent, the transmission fails inevitably after many retransmission attempts. This affects the effective throughput significantly while trying to accommodate many devices. Thus, the scalability of the MAC protocol is greatly affected by the high rate of collision probability in LS-IoT.

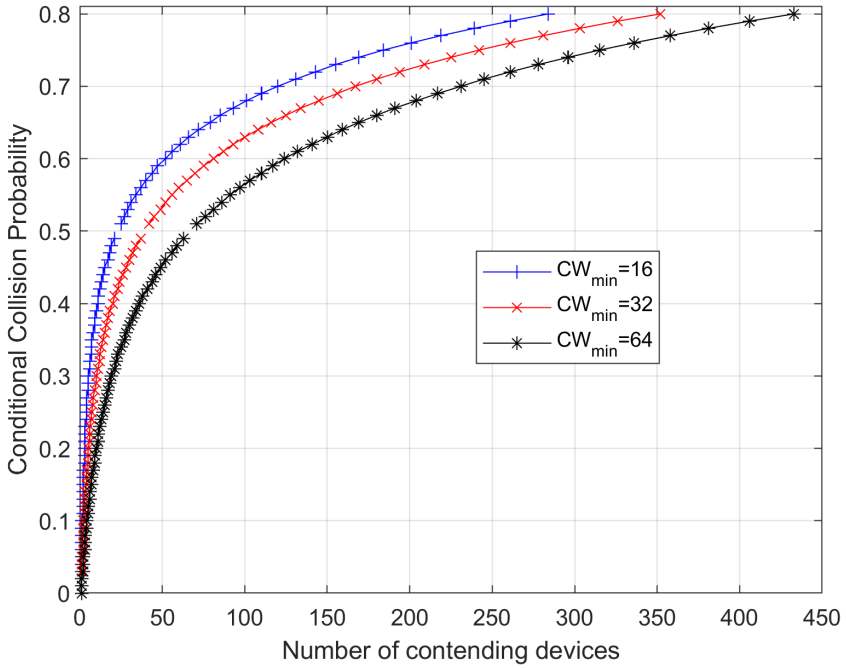

Figure 2.4: The effect of a rising number of contending devices on the probability of a collision using DCF contention-based MAC scheduling approach as modelled in [29].

2.1.3.2 Control Overheads

Generally, wireless MAC necessitates the exchange of control frames for signalling to make the protocol practically possible [30]. These control overheard frames have an impact on the MAC. In the widely adopted hybrid-based MAC protocols and contention-free based protocols,

synchronisation between devices is necessary. The synchronisation of devices implies that the duration of the signalling information needs to be compensated for within the time frame duration that is to be allocated for transmission. As a result, there is a consequence in terms of reduced available time for contention and frame transmission. In terms of LS-IoT networks, the need for the coexistence of dynamic traffic patterns and heterogeneous device types makes large control overheads inevitable [31]. Some LS-IoT networks that aim to achieve coexistence at the expense of increased control overheads can be found in [32]. A classic case of the impact of LS-IoT on control overhead issues is the Physical Random Access Channel (PRACH) approach which uses 64 preambles for contention-free and contention-based media scheduling [33]. The 3GPP group has identified the RACH procedure as a major issue for the existence of massive heterogeneous devices due to the surge in signalling overhead caused by the increasing number of M2M devices [34].

2.1.3.3 Spectrum Constraints

The design of a scalable MAC may require the need for adequate bandwidth. Several traditional wireless communication technologies still rely on the licensed spectrum due to the signal fidelity and minimal spectrum interferences that come with using the licensed band. However, MAC protocols that rely on the licensed band are highly constrained due to the limited availability of spectrum. On the other hand, many IoT standards are moving to the unlicensed spectrum band to provide connectivity for M2M devices since it is free of monetary cost. The fact that the unlicensed band may be used without formal allocation means that there could be high interference. In the case of LS-IoT, the existence of a multi-spectral access network is a reality. LS-IoT access networks are meant to provide a converged multi-standard connectivity of devices. The MAC could be constrained as a result of channel bandwidth and channel interference. LS-IoT networks with a constrained spectrum bandwidth, a large number of device connections and a high level of interference will result in a reduction in the maximum data rate as modelled in [35] and shown in Figure 2.5. Figure 2.5 indicates that the LS-IoT networks with a constrained spectrum bandwidth and a large number of device connections will result in a reduction in the maximum data rate.

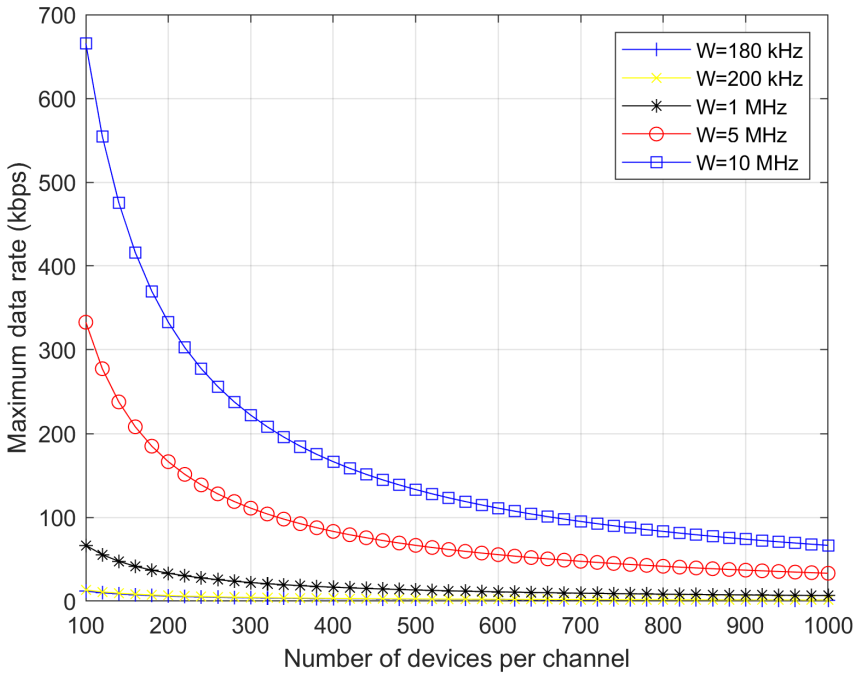

Figure 2.5: The effect of a large number of devices and spectrum bandwidth constraints on the maximum data rate as modelled in [35]

2.1.3.4 Timing Constraints

The MAC service time is a critical component of the resource scheduling process. The service time indicates how fast, how effective, and how scalable transmission and reception may be achieved. For a typical contention-based MAC protocol, the MAC service time is the interval between the start of contention and the end of a successful transmission (thus, including acknowledgement) or until the frames are discarded due to a failed transmission [36]. For contention-free based protocols, the duration of the contention is eliminated since there is no contention. From a general perspective, the MAC protocol may require guard time to accommodate any synchronisation offset. There is also some delay introduced by the computational processing of frames that are necessary to establish critical parameters for resource provisioning. This includes time slot computation, contention windows, synchronisation times, error detection, Acknowledgement generation and other complex mandatory or optional computational functions. Some analysis of the delays due to the processing of frames in some wireless

technologies can be found in [37],[38] and [39]. In [40] some simulations and analyses are conducted to establish the relationship between frame inter-arrival time and the collision probability for contention-based MAC scheduling. It shows that when the average inter-arrival time increases, the collision probability drops exponentially and so does the average delay. In LS-IoT, industrial-based M2M connectivity contributes significantly to the large-scale nature of IoT networks and is sometimes referred to as Industrial IoT (IIoT) [41]. Automation and control applications in mega-factories, aviation, road transportation and shipping, collectively require millions of sensors. Such applications contribute to the greatest number of sensors in a confined place at a given time with a typical example being a commercial aircraft embedded with about 6000 sensors. Such LS-IoT networks rely on real-time data and strict latency requirements due to strict regulations on safety and other factors. In such a case the MAC service time must be strictly bounded. To enable scalability, the MAC service delay must be reduced. Also, the underlying hardware and firmware factors such as interrupts, multi-thread processing, and switching affect the ability to reduce the time delays [42]. This hinders the possibility of achieving a scalable MAC protocol.

2.1.3.5 Hardware Constraints

A scalable MAC depends on the interfacing between all the components of the hardware platform. The interconnection between the different PHY processing units affects the timing of the MAC. Delays may exist in the path between the different PHY processes and the MAC functions [43]. The MAC may require the radio to switch between sleep, transmit and receive modes or between channels frequently. This process introduces some amount of delay which affects the ability to provide timely MAC functions to support scalability [44]. The Random Access Memory (RAM) Processor combination is usually a resource shared amongst different layers of the communication protocol stack. For LS-IoT networks, the massive frame exchanges result in the culmination of all the processing path delays and queuing which causes a slowdown of the MAC scheduling of frames and affects the time constraint requirements. Some LS-IoT technologies such as NB-IoT, LTE-M and IEEE 802.11ah employ complex and lengthy procedures for single device access at the device or GW to manage a large number of devices. Therefore,

if the processing of frames during the execution of the MAC protocol procedures is not fast enough due to the microprocessor's incapability, a bottleneck situation may occur which in turn affects scalability. Such hardware constraints are very critical in LS-IoT networks due to the increasing number of connected devices to be managed. Moreover, the execution of complex algorithms for fidelity and MAC enhancements to support LS-IoT requires a substantial amount of processing and memory capabilities.

2.1.4 Summary

In summary, considering the complex computations, high level of concurrency using multi-channel approaches and various resource constraints, the GW MAC may experience congestion which imposes limits on the media access scalability. Moreover, very little emphasis is placed on the augmentation of the GW MAC component to further support the scalability of the media access in terms of managing simultaneous transmissions.

2.2 Related work on GW MAC constraints in LS-IoT

In [45], the authors modelled and analysed the throughput effects of buffer overflows in an IoT access point for traffic flows based on the infrastructure mode. The authors use the Markov regenerative processes in conjunction with a fluid model to analyse the response times and the fairness of the IoT traffic. They also proposed an adaptive admission control mechanism to improve the response time and throughput of the access point which is modelled based on the blocking probability of an M/D/1 queue.

In [46], the authors examined the funnelling effect of a GW used for industrial wireless communications which is one of the applications of LS-IoT. They analyse the congestion at the GW based on the hop-by-hop and many-to-one network structure. In their study, they also proposed an on-demand Time Division Multiple Access (TDMA) approach in the MAC layer super-frame of the GW as well as a priority base-based communication scheduling algorithm. The analysis presented in their work suggests that the congestion in the GW can be solved by providing extra communication opportunities and control within the GW.

In [47], the author discussed the effects of congestion in the GW and how it leads to congestion at the MAC level. The author argues that in an ad hoc network the number of relayed packets to a GW can cause congestion at the GW MAC, hence high packet delays. As such, the author proposes an approach based on load balancing between GWs and congestion alerts. In essence, an intermediate GW can distribute its load to other gateways by path sharing based on their routing packet traffic to ease congestion.

In [48], [49] and [50] the authors evaluated the MAC scheduling protocol's throughput performance model based on the CSMA/CA scheme for IEEE 802.11ah networks for the Sub 1 GHz (S1G) devices while considering the effects of sectorisation within a RAW, the time constraints of the RAW slot concerning the traffic load of the devices and the duty cycle limitations of the GW and devices respectively. The results presented in these studies demonstrate classical behaviours of the IEEE 802.11ah standard which are based on the throughput performance of the MAC scheduling scheme.

In [51] the authors suggested that the buffer constraints at the GW node do not directly influence the behaviour of the MAC scheduling scheme because the channel contention executed by the devices occurs before the queue overflow. However, their argument was based on the queue overflow of the upper MAC layer of the GW node since the upper MAC layer of the GW does not directly influence the PHY layer where the GW needs to send ACK and other synchronisation frames as part of the channel access transmission procedure. However, their work does not explicitly present the results of the impact on the throughput of the MAC scheduling algorithm.

2.2.1 Summary

In summary, the related works do not consider the large-scale nature and heterogeneous nature of an IoT network to establish the extent of the degradation in throughput performance. The evaluation of the media access throughput performance for LS-IoT based on the GW MAC congestion and resource constraints requires further studies. It is fundamental to evaluate the

Table 2.2: A summary of some related work on GW MAC Constraints in LS-IoT

Related Work	**Summary of work**	**Comments and limitations**
Pokherl et al. [45]	Analyses the effects of buffer overflow in the GW using a throughput model. Addresses GW congestion using an adaptive admission control mechanism based on blocking probability.	Based on TCP flows rather than MAC frames. Does not consider a large number of devices, GW MAC buffer and processing constraints.
Zhong et al. [46]	Analyses the funnelling effect of the GW MAC based on bandwidth, slot time, and collisions. Addresses GW MAC congestion using on-demand TDMA	Does not consider the GW MAC buffer and processing constraints.
Paliwal. [47]	Analyses the GW MAC congestion based on relayed packets.	No model was used for analysis. Does not consider the GW MAC processing constraints. Large-scale devices are not considered.
Bhandari et al. [48], Lei et al. [49] and Qutab et al. [50]	Analyses the IEEE 802.11ah throughput model based on effects of sectorisation, RAW, device traffic load and duty cycle	Does not consider the GW MAC buffer and processing constraints.
Xu et al. [51]	Analyses the MAC scheduling scheme based on the GW MAC buffer constraints.	Does not explicitly show the impact on the throughput due to the MAC scheduling algorithm

effects of the classical GW MAC in supporting media access scalability enhancements.

2.3 Related work on Virtual Network models

Network virtualisation involves the abstraction of hardware and software network resources and various network functions into a single unit whereby the resources and functions can be shared [52]. The combination of different network components to form a single entity for sharing of resources is referred to as a virtual network. In recent times, virtual networks have been used to decouple a range of network functions that are traditionally fixed on a single dedicated hardware, to provide differentiated services for improved utilisation [53]. However, the distinctive nature and constraints of the media access control functions such as those highlighted in section 2.1.3 make the design and modelling of a suitable virtual network very complex [52]. Several existing studies have proposed various virtual network models for different access technologies. To propose a suitable virtual network model for augmenting and decoupling the GW MAC functions in a LS-IoT network, some related work on virtual network models is surveyed and presented.

The authors in [54] proposed a massive IoT Protocol Stack Virtualisation (PSV) framework for virtualising the IoT gateways as well as the devices and the applications for addressing computational constraints and resource limitations. A controller for the protocol stack is deployed in the core network to support the IEEE 802.15.4 access network and a mathematical model based on packet loss probability is used to evaluate the number of devices that can be accommodated using the proposed PSV framework. Their approach does not consider the transmission delays between the gateway and the remotely deployed controller. The approach also does not fully virtualise the media access component of the gateway since it is only the controller that is virtualised.

In [55] the authors proposed a virtualisation mechanism for access control in a 5G-based IoT Gateway for a massive number of devices. The access control and authentication are handled as a virtual Gateway function deployed on a third-party core network or cloud for IoT devices.

The impact of the proposed approach on the network load is evaluated regarding signalling overheads. Although the access control functions are virtualised, the routing of traffic to the network slice is not isolated from the GW which could lead to a bottleneck under higher load conditions.

The authors in [56] proposed an IoT GW virtualisation approach called the Edge Massive Machine-type Communications (mMTC) slicing. In their work, a two-tier IoT GW structure is proposed for virtualisation. One tier provides direct access between IoT devices and the physical IoT-GW whereas another tier is made up of virtualised GWs at the network edge which provides a dedicated IoT application platform. The proposed concept was evaluated on an Open Source Management and Orchestration (MANO) using a WiFi access point. Their proposed approach may not be suitable for large-scale devices given that the physical GW still manages device access, and another mediation access layer is added virtually which reduces the media access throughput when more tenants are created. In addition, the virtual link between the physical IoT GW and virtual GW is not explicitly defined and examined.

In [58] the authors presented an NFV approach for virtualising the GW to split the application logic of the GW from the physical GW onto the cloud based on non-time-critical and time-critical applications. They also proposed the use of SDN to dynamically schedule network functions between data centres. Although the work presented explains the impact on latency and network traffic, the work was not evaluated to examine the impact on latency and the media access throughput for a large number of devices. In addition, the virtual link is not explicitly presented.

The authors in [59] proposed an approach to partially decouple some Access Point MAC processing functions of the IEEE 802.11 protocol to the cloud. In their work, the aim was to achieve network flexibility and reconfigurability by leveraging the benefits of SDN. The virtual network handles the processing of the MAC layer frames as well as the generation of management frames. An SDN controller based on OpenFlow is used to manage the movement of virtual traffic between the virtual access point and the physical access points. The results

obtained from this work indicate similar performance with the legacy IEEE 802.11 protocol. However, their proposed approach may not be suitable for large networks given that the results obtained are based on a single device connected to the access point and the MAC is augmented into multiple instances.

In [60], a generic approach is proposed for virtualising the MAC in heterogeneous networks with different radio access technologies. The author uses a novel approach to offload resources for mobility and device scheduling. The results obtained in their work suggest that resource utilisation can be enhanced by virtualising the MAC function of the radio access component. However, their work does not provide a model for characterising the virtual network performance. It only defines the signal flow and resource allocation.

In [61] the authors conducted an experiment based on the deployment of virtual MAC protocols in a shared network. They used a software program called MAClets to provide real-time virtualisation of the CSMA/CA based MAC scheduling of devices in an IEEE 802.11 network. The results obtained in the study merely indicate a saturation of the throughput level of the virtual MAC as additional devices join the network. However, the proposed approach is implemented on the access point devices hence it cannot scale effectively due to constrained resources.

In [62] the authors proposed a virtualisation model for aggregating the functions of several Wireless Local Area Network (WLAN) access points to reduce interference of coexisting access points and improve the throughput. A physical access point architecture is proposed for hosting the virtual access points and running the virtual access point functions concurrently. A layer 2 flat tunnelling virtual network model is proposed to establish communication between the virtual access points and the network domain through a router. A virtual access point migration and management component are also proposed. Their work was implemented as a testbed and the results show a slight decrease in throughput. By using a physical access point as the host of the virtual access point, the proposed strategy may not scale effectively due to constrained resources.

In [63] the authors proposed a framework called Virtual Network Embedding in Wireless Systems (VNEWS) based on the ProtoGENI V2 format RSpec resource specification language. The proposed approach creates the coexistence of multiple virtual transmission topologies through spectrum slicing and a heuristic algorithm is used to provision resources for transmission. Although the proposed concept was implemented based on a testbed, this study does not evaluate the performance of the virtual network architecture.

In [64], the concept of SDN over network virtualisation is used to develop an architecture called OpenRAN with aim of achieving complete virtualisation and programmability of the Radio Access Network (RAN). The proposed architecture is based on three main parts namely, the Wireless Spectrum Resource Pool (WSRP), Cloud Computing Resource Pool (CCRP) and SDN controller. The WSRP translates a physical Remote Radio Unit (RRU) into several virtual RRUs. The CCRP provides a high-speed cloud computing network that hosts the virtual access elements. The SDN controller provides the control layer for integrating the control functions of the virtual access elements based on an SDN agent. The effectiveness of the proposed approach can not be verified since the approach was not modelled nor tested.

In [65], [66] and [67] the Cloud RAN topology based on the CPRI for virtualising the RAN function elements was modelled using G/G/1 queues to ascertain the performance metrics for 5G networks. These works suggest high accuracy in the use of G/G/1 queuing models for aggregated virtual flows. However, the virtual network topology used in these studies does not capture all the virtual network components and the performance metric studied is focused on the CPRI specific node rather than based on the entire virtual network components.

In [68], the authors presented an analytical model for virtualised network functions based on an open queuing network of G/G/m queues. The model can estimate the mean response time of a virtual network function. However, their model does not factor in the virtual network nodes. It only focuses on the chain of virtualised network function components executed in the data centre.

Table 2.3: A summary of some related work on virtual network models

Related Work	**Proposed Strategy**	**Comments and possible drawbacks**
Herrero [54]	Massive IoT Protocol Stack Virtualisation (PSV) for IoT GW deployed in the core network	Delay between the GW and the remote controller not considered.
Behrad et al. [55]	Virtualises the access control for 5G-based IoT GW deployed in a third-party core network.	Impact of signalling flows considered. Network slice traffic is not isolated from the GW. There is a possibility of bottlenecks.
Theodorou et al. [56]	Two-tier IoT GW structure for virtualising IoT GW using MANO	Fast instantiation of slices. The virtual link is not explicitly defined.
Miladinovic et al. [58]	Dynamic scheduling of network functions based on time-critical and non-time critical applications.	Impact on the virtualisation latency and MAC throughput for a large number of devices is not considered.
Vestin et al. [59]	Partial decoupling of GW MAC processing to the cloud using OpenFlow SDN controller.	Results based on a single device. Hence not be suitable for large networks.
Fan et al. [60]	Offloading of GW MAC in the heterogeneous network for mobility and device scheduling.	Lacks a model for characterising the virtual network performance
Garlisi et al. [61]	Deployment of virtual CSMA/CA MAC protocol in a shared network using MAClets tool	Internally virtualised MAC. Hence, may not scale effectively due to constrained resources.
Nagai et al. [62]	Virtualises aggregated functions of multiple GW nodes using layer 2 flat tunnelling	Internally virtualised MAC. Hence, may not scale effectively due to constrained resources
Leivadeas et al. [63]	Coexistence of multiple virtual transmission topologies using spectrum slicing and a heuristic algorithm	Does not evaluate the performance of the virtual network architecture
Yang et al.[64]	RAN virtualisation based on SDN, Wireless Spectrum and Cloud Computing resource pools	No analytical model nor physical experimentation.
Perez et al.[65, 66], Gowda et al.[67], Prados et al.[68]	Virtual network based on G/G/1 queuing for aggregated virtual flows	Few virtual network components considered. Models are based on the CPRI node only and not the entire network.

2.3.1 Summary

Based on the literature survey conducted, several virtual network frameworks or models exist. However many of these studies do not provide an adequate model that shows the performance metrics of each virtual network node as well as the entire virtual network. This is fundamental for virtualising the GW MAC where timing deadlines are imposed. Also, the virtual network node's performance enables the identification and incorporation of the critical nodes in the virtual network management strategy to optimise performance. Although the studies presented provide some virtual network models or test-bed strategies, they may not provide a complete approach for GW MAC virtualisation for LS-IoT networks due to the lack of protocol translation strategies, routing strategies and resource allocation strategies as a whole. Therefore, it is fundamental to propose a virtual network model that fully captures the key contributing nodes or components and to also provide a suitable analytical model that can be used to assess the performance of the virtual network nodes, the entire network and the impact of the media access throughput.

2.4 Related work on Virtual Network Function placement strategies on the edge

There are various approaches for placing VNF functions on the edge. A number of these approaches are reviewed for their suitability in placing VNF functions related to the GW MAC functions in heterogeneous LS-IoT networks.

In [69] the authors proposed a heuristic algorithm for the management of radio access technologies onto a set of differentiated virtual base stations. The proposed algorithm tries to provision virtual resources to meet the capacity demand of the virtual base stations based on different service requirements, usage and isolation and the changes in the wireless channel. The results obtained indicate that the average serving data rate is always above the minimum required data rate for guaranteeing the required service. However, their work does not explicitly present the algorithm.

In [70] an online placement algorithm for VNFs in a Multi-access Edge Computing (MEC) environment is proposed to support the 5G IoT architecture. The algorithm takes into consideration the latency of the placement mechanism where the VNF is first allocated an appropriate edge node and the VNF instance is instantiated and orchestrated based on the traffic demand. Although the proposed approach improves the capacity of the network this approach is based on user data traffic only. Hence it may not be applied for media access control related traffic due to the different service level agreements required. Hence this approach can not be applied in the context of this study.

In [71], an optimisation strategy based on a heuristic algorithm is proposed to increase the number of VNFs deployed by searching for nodes with abundant resources and reserving the resources based on priority. Although the proposed approach improves the performance, this approach does not characterise each request based on the traffic rate. This makes it difficult to be applied for GW MAC placements where the traffic load is imperative for reducing the virtualisation delay.

Kim et al. proposed a VNF placement method in [72] which is based on the knowledge of the number of resources and a priori periodic information about resource levels. This approach does not consider the virtualisation delay which is imperative for GW MAC related functions where timing deadlines are to be respected.

In [73] an approach is proposed to enhance the placement of VNFs in a MEC for accommodating mission-critical and delay-sensitive traffic to minimise the end-to-end communication delay based on a Tabu Search algorithm. Although the approach reduces the end-to-end delay, it is based on only one traffic type. Therefore this approach may not be suitable in a heterogeneous network where different traffic types are to be considered as is the case in this study.

Table 2.4: A summary of some related work on Virtual Network Function placement strategies on the edge

Related Work	**Proposed Strategy**	**Comments and possible drawbacks**
Caeiro et al.[69]	Virtual resource provisioning heuristic algorithm based on different service requirements, the usage, isolation and the changes in the wireless channel	The Algorithm is not explicitly presented
Sarrigiannis et al.[70]	Online placement algorithm is based on VNF placement latency and traffic demand	Improves the capacity of the network. Not suitable for MAC traffic due to different service level agreements required.
Morin et al. [71]	Heuristic optimisation algorithm based on abundant resources and reservation of resources based on priority	Improves the performance but does not characterise the traffic rate of each request.
Kim et al. [72]	Placement based on the number of resources and a priori periodic information about resource levels	Does not consider the virtualisation delay
Leivadeas et al.[73]	Minimises the end-to-end communication delay based on a Tabu Search algorithm	Based on only one traffic type. Not be suitable for a heterogeneous network

2.4.1 Summary

In summary, most of the VNF placement strategies presented in literature do not support the virtualisation of MAC related functions for IoT networks because of the relaxation or inadequate consideration of the virtualisation delays in the placement process as well as the sensitivity and heterogeneous nature of the traffic as a whole. These approaches are not suitable for the GW MAC in heterogeneous LS-IoT network placements due to the strict timing deadlines imposed by the 802.11ah standard to achieve a successful transmission. Therefore it is imperative to design and develop a placement strategy that can be employed within the context of GW MAC virtualisation using a virtual network model that estimates the overall virtualisation delay for the different traffic types of GW MAC related frames.

2.5 Conclusion

In this chapter, a literature review was conducted and presented. A review of the MAC protocols, the associated LS-IoT technologies and the GW MAC related challenges were presented. Based on the review conducted, the GW MAC may experience congestion which imposes limits on the media access scalability. Moreover, very little emphasis is placed on the augmentation of the GW MAC component to further support the scalability of the media access. Some related works on the GW MAC constraints in LS-IoT were also presented. Based on this review, it is fundamental to evaluate the effects of the classical GW MAC in supporting media access scalability enhancements. Related works on virtual network models were also presented where it is argued that many of the studies conducted do not provide an adequate model that shows the performance metrics of each virtual network node as well as the entire virtual network. Lastly, a review of related works on virtual network function placements in the edge was presented where it can be concluded that existing approaches are not suitable for the virtualisation of the GW MAC in heterogeneous LS-IoT network placements due to the strict timing deadlines imposed by the 802.11ah standard to achieve a successful transmission.

CHAPTER 3. VIRTUAL MEDIA ACCESS WITH GATEWAY MEDIA ACCESS CONTROL CONSTRAINTS IN LARGE-SCALE IOT

3.1 Introduction

Recent approaches, technologies and standards have been developed to address scalability in very large IoT networks with an emphasis on the MAC component. One of the best strategies to manage media access for a large number of devices is to provide concurrent media access. This can be achieved through the concept of RF virtualisation whereby co-located devices can be grouped such that each group contend for channel access using a dedicated spectrum or a beam-formed signal [74]. These strategies together with other proposed enhancements at the PHY layer such as in [75], [76], [77] and [78] aim to support concurrent independent transmissions in a LS-IoT network. However, to support such strategies in a LS-IoT network, the GW MAC needs to equally manage transmissions concurrently otherwise congestion is imminent due to the resource constraints of the GW node in a typical infrastructure mode setting [79]. This is because of the high amount of traffic arrivals from independent seemingly concurrent PHY layers. In this chapter, concurrent virtual media access for supporting scalability using a dedicated GW MAC is studied based on evaluating the MAC throughput performance for LS-IoT networks and the impact of the GW MAC functions thereof. Thus, the aim is to examine the throughput performance of the MAC scheduling scheme based on the 802.11ah standard with virtual access groups for large IoT networks considering the constraints of the GW MAC functions. To achieve this aim, the following objectives are achieved in this chapter:

a. To establish and study the throughput model for an unsaturated network based on the 802.11ah RAW and PS-Poll mechanism for LS-IoT networks using a CSMA/CA probabilistic model.

b. To evaluate the impact of a constrained/dedicated GW MAC (non-virtualised) component on the throughput of the 802.11ah MAC scheduling scheme based on concurrent media access (virtual access groups) in an infrastructure mode by incorporating the GW MAC buffer and processing resource constraints in the throughput model.

3.2 The IEEE 802.11ah Media Access Control

In this study, the IEEE 802.11ah media access scheme is used as it is intended to support LS-IoT based applications with power constraints and heterogeneous packet transmissions. This section further elaborates on the IEEE 802.11ah MAC protocol. The devices use the PS-Poll mechanism for channel access and power efficiency. The 802.11ah standard proposes an improvement to the contention-based CSMA/CA protocol used in the legacy 802.11 standard which is not suitable for large-scale M2M devices. The improvement is such that a high collision probability and a severe hidden terminal problem in large size networks are minimised using a RAW approach as illustrated in Figure 3.1. The RAW concept divides the transmission resources into restricted time frames during a Beacon Interval (BI), which then allows a group of devices to contend for channel access in an assigned RAW using the classical CSMA/CA or DCF. The remaining devices that do not belong to that RAW group can then go to sleep until the beginning of the group's RAW slot to save power. The GW, referred to as the Access Point in the standard, broadcasts the initiated RAW parameters (RAW Start time, RAW duration, Device ID) using short beacon frames [80]. In essence, the grouping of the devices may be done by the GW device. The process minimises the number of devices that contend for channel access at a given time to minimise the probability of collisions. It is important to highlight that, when there is a large number of devices, the number of RAW groups will be large, therefore the BI will be long which can affect the transmission duration and in turn affect the scalability of the network.

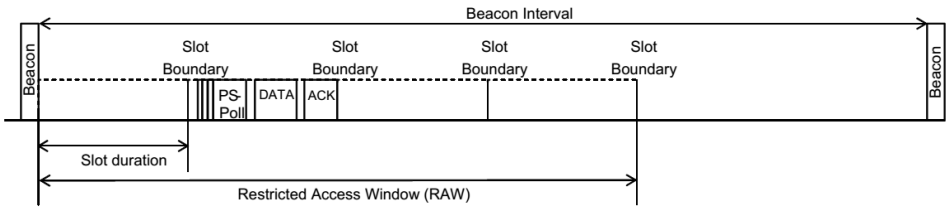

Figure 3.1: A Restricted Access Window (RAW) with CSMA/CA based channel access in a RAW slot [2].

3.2.1 Channel Access Procedure

For any device to be granted a transmit opportunity within its RAW slot, the channel access procedure must be executed using the CSMA/CA contention process. In the CSMA/CA scheme, a device that has a frame in its MAC buffer to be transmitted, first senses the communication channel to detect if the channel is free from any ongoing transmission. If the device senses that the channel is idle within a period called a DCF Inter-frame Spacing (DIFS), the device then initiates the transmission of the data packet. On the other hand, if the device senses that there is an ongoing transmission in the channel during the DIFS period, the device continues to monitor the channel until the channel is idle for the duration of the DIFS. When the channel is idle, the device enters a random back-off (BO) procedure before transmitting the data packet. The random BO procedure in this case serves as a collision avoidance mechanism. The random BO procedure does not completely avoid collisions. However, it reduces the probability of having multiple transmissions in one channel. The random BO is selected over a range of 0 to CW, where CW is the contention window, starting with the minimum contention window CW_{min}. After randomly selecting a BO time, the device then initiates a down counter until it reaches zero before transmitting a packet. If the transmission is unsuccessful after the BO timer reaches zero, possibly due to a collision, then the contention window is increased to $2^i CW_{min}$ upon each failed transmission attempt until the maximum contention window value CW_{max}. The parameter i represents the current BO stage. The BO procedure is executed so long as the channel is idle. However, it is frozen when a transmission is detected and resumed when the channel is idle again for a DIFS period. The time duration for transmission after the DIFS period is divided into time slots, t_{slot}. Devices therefore may only transmit at the begin-

ning of a timeslot. The duration of the time slot is therefore very critical in the performance of the LS-IoT network. This is because the time slots are shared amongst all other devices contending for channel access in that RAW slot.

3.2.2 Transmission Procedure

A complete transmission cycle is made up of several slot activities which are illustrated in figures 3.2 and 3.3. There are two possibilities in a transmission cycle. There is either a possibility of successful transmission or an unsuccessful transmission. However, in general, the transmission cycle is made up of a DIFS, a BO (channel access), frame transmission, Short Inter Frame Spacing (SIFS), ACK and signal propagation.

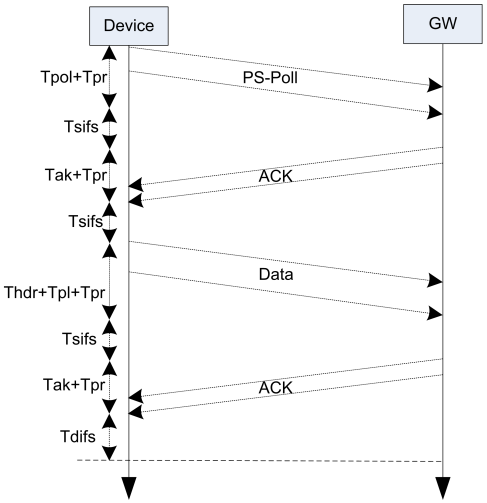

Figure 3.2: The MAC transmission procedure of a successful frame in a RAW timeslot for S1G IoT devices using PS-Poll.

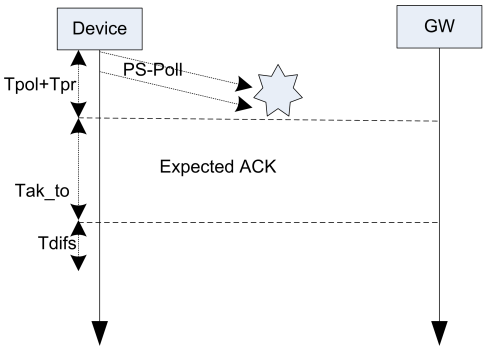

Figure 3.3: The MAC transmission procedure for a collided frame in a RAW timeslot for S1G IoT devices using PS-Poll.

The uplink transmission is explained since most IoT applications transmit data in the uplink direction and this study focuses on the uplink direction. When a device has data to transmit, it performs the random CSMA/CA channel contention procedure before it transmits a PS-Poll frame to the GW to reserve a RAW slot for data transmission on the condition that the transmission access was successful. This takes a duration equivalent to the PS-Poll PHY layer transmission time T_{pol} plus the prorogation delay T_{pr}. The GW device receives the PS-Poll frame which processes the frame and responds after a T_{sifs} duration to the device with downlink data or an ACK frame notifying the device and all the other devices that successfully transmitted a PS-Poll to contend for channel access using the random channel access procedure based on CSMA/CA for data transmission. The ACK frame contains a bit field which notifies the devices of any downlink buffered traffic intended for it. This exchange takes the duration equal to T_{pr} plus the ACK frame's PHY layer transmission time T_{ak} to be received at the device. After the device receives the ACK frame, it then transmits the data frame which contains the header frame and payload frame after waiting for a T_{sifs} period. This takes a total duration equivalent to the header transmission time T_{hdr} and the payload frame transmission time T_{pl} plus T_{pr} to be received at the GW. The GW then processes the header and data frames and waits for a T_{sifs} period before transmitting an ACK frame back which takes $T_{pr} + T_{ak}$ period before it is received by the device. After receiving the ACK, the device waits for another T_{difs} period before it attempts to transmit the next PS-Poll based on contention. If there is an unsuccessful transmission due to congestion, there is no frame received at the GW which implies that there is no ACK transmitted back to the device after a PS-Poll frame is transmitted. However, the device waits for a $T_{ak_{to}}$ duration until it considers that the PS-Poll was unsuccessful. Therefore, the duration of the collision is defined by the ACK time-out value, the duration of the PS-Poll frame, the T_{difs} period for the next transmission attempt and the propagation delay T_{pr}.

3.3 Throughput Model

The 802.11ah MAC scheduling scheme with the scheduled RAW and PS-Poll mechanism based on the CSMA/CA process is first studied and modelled in the context of a heterogeneous LS-IoT network before introducing the GW MAC constraints. The unsaturated DCF model presented

in [81] is studied and extended to achieve this. The model in [81] which is an extension of the saturated model presented in [82] has been widely applied in many works due to its accuracy. The idea is to model the performance probabilities and estimate the throughput of the MAC protocol for an unsaturated network. The unsaturated condition adequately represents the heterogeneous nature of traffic in a typical IoT network. An unsaturated IoT network implies that M2M devices transmit frames at defined relatively short and long intervals. The unsaturated model enables the modelling of the IoT network performance based on different average traffic generation rates for various LS-IoT applications. In essence, the model considers the different traffic load conditions attributed to a heterogeneous LS-IoT network.

To model the throughput with the effects of the GW MAC constraints for supporting concurrent transmission, it is assumed that the PHY layer multiplexing guarantees concurrent independent channel access using group or transmit opportunity (TXOP) sectorisation over orthogonal multichannel. The 802.11ah proposes such concurrent transmission [83]. This needs to be supported by a concurrent PHY layer transmission strategy. Some of these strategies have already been proposed and studied in literature. Based on this, each PHY layer grouping is referred to as a Virtual Access Group (vAG) in this study. The proposed grouping of devices into vAGs takes the form of RF virtualisation over Software Define Radio (SDR) at the PHY layer. The RF front-end of the GW is abstracted and sliced into independent virtual RF front-ends. This concept is proposed for IoT networks in [84]. Based on this, each virtual RF front-end can be used by different vAGs to achieve concurrent transmission. The management of the media access concurrently at the GW based on the vAGs is the focus of this study. The RF virtualisation aspect is beyond the scope of this work. It is also assumed that the characteristics of each vAG are the same in terms of the number of devices, the PHY channel data rate and the channel conditions. It is also considered that the channel condition is ideal (absence of noise in the channel) and no hidden terminals exist in each vAG. It is also assumed that all the data transmissions are initiated in the uplink direction (device to GW) and the traffic load of all devices is unsaturated. However, since the GW may respond to the PS-Poll with downlink data or an ACK frame, the transmission procedure inherently captures the

downlink data transmission also.

3.3.1 Throughput without GW MAC Constraints

Considering D as the overall number of devices associated with a single GW node in infrastructure mode, the number of devices D is divided into n_{ag} number of vAGs to give D_{ag} number of devices per vAG. In each vAG, devices contend for channel access using the RAW mechanism independent of other vAGs. This implies that D_{ag} number of devices in a vAG are further grouped into n_{rw} RAW groups whereby d is the actual number of devices that contend for channel access in a given RAW time slot. The number of contending devices in a RAW slot d can be estimated using 3.1 as suggested in [50] which is extended here to include the groupings according to vAGs. The upper limit is also taken to obtain an integer value and ensure that all D_{ag} devices are taken into account.

$$d = ceil\left\{\frac{D}{n_{ag} \times n_{rw}}\right\} \tag{3.1}$$

The RAW and PS-Poll channel access and transmission occur in a single vAG, therefore the throughput for a single vAG is considered based on the illustration of the network in Figure 3.4. All the devices that contend for a transmission opportunity in a single vAG and the devices in a vAG are further grouped into multiple RAW groups based on the 802.11ah standard. This means that the throughput can be obtained based on the amount of successfully transmitted payload in a given RAW slot of a vAG.

The CSMA/CA MAC protocol processes can be described based on statistical information (probabilities). The conditional collision probability P_c and the probability that a device initiates a transmission in a randomly scheduled time slot τ^{us} are obtained first. The conditional collision probability P_c is defined in 3.2 and is based on the probability that, in a given time slot at least one of the remaining $d-1$ devices may transmit some packet in a time slot with a probability τ^{us}.

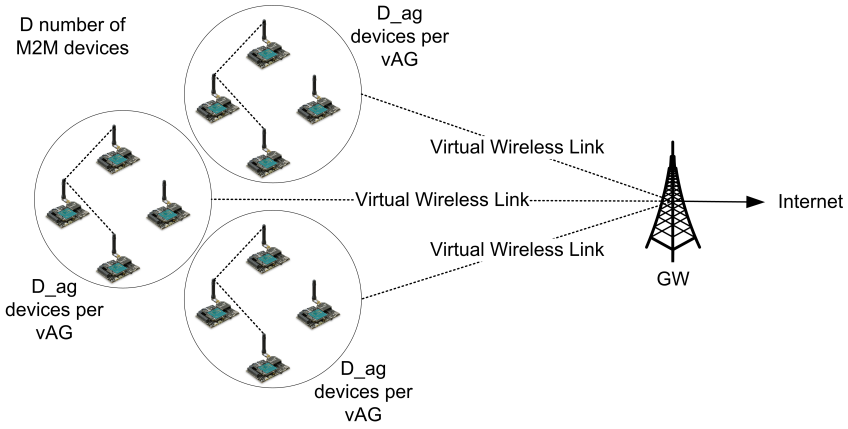

Figure 3.4: M2M devices connected to a GW node in infrastructure mode based on virtual access groups for concurrent media access.

$$P_c = 1 - (1 - \tau^{us})^{d-1} \tag{3.2}$$

The probability τ^{us} for an unsaturated network is the fundamental part of the model. This is because τ^{us} describes the contention process which occurs before the attainment of a transmit opportunity. This process contributes to the throughput performance. In an unsaturated network, the probability that a device initiates a transmission in a time slot depends on the behaviour of the packet arrivals from the upper MAC layer (the part of the MAC that interacts with the Logical Link Control sublayer) of the device for transmission. Therefore, to obtain τ^{us} a Markov model that defines the CSMA/CA contention process based on the rate at which packets arrive at the MAC buffer for transmission must be used. There are numerous Markov models proposed in literature for unsaturated networks such as [85] and [86]. However, for accuracy and ease of computation, the Markov model proposed by [81] is adopted which yields the equation for τ^{us} as depicted in 3.3. The parameter CW_0 is the contention window of the initial BO stage, CW_i is the i^{th} contention window represented as $2^i CW_{min}$ and m is the number of BO stages. In 3.3 the model considers the probability P_{ba} where the channel is busy when a packet arrives for transmission or during the DIFS and the probability P_{na} where no packet arrives for transmission. Also, the probability P_{np} where a new packet arrives for

transmission during a time slot is considered.

$$\tau^{us} = \frac{1/(1-P_c)}{\frac{CW_0+1}{2} + \frac{P_{na}}{P_{np}} + 1 + \frac{CW_0-1}{2}P_{ba}P_{na} + \sum_{i=1}^{m-1}(P_c)^i\frac{CW_i+1}{2} + \frac{(P_c)^m}{1-P_c}\frac{CW_m+1}{2}} \tag{3.3}$$

The probability P_{ba} is given in 3.4 which is based on the probability that there is at least one busy instance in a DIFS period. The busy instance is described by the probability of a collision and a successful transmission during the average duration of a generic slot t^*_{slot}. The parameter t^*_{slot} here represents the average slot duration covering the slot activities including idleness, collisions and successful transmissions.

$$P_{ba} = 1 - e^{-\frac{1-(1-\tau^{us})^d \times T_{difs}}{t^*_{slot}}}. \tag{3.4}$$

The probability P_{na} is given in 3.5 which is simply derived based on the utilisation (λ_d/μ_d) of the MAC queue in the device. Thus, when the utilisation of the queue increases, the lower the probability of having an empty queue which indicates no packet arrivals. λ_d is the rate of packet arrivals for transmission and μ_d is the service rate of the packets in the device.

$$P_{na} = \begin{cases} 1 - \lambda_d/\mu_d & \lambda_d/\mu_d < 1 \\ 0 & \lambda_d/\mu_d \geq 1 \end{cases} \tag{3.5}$$

The service rate μ_d of the packets that arrive for transmission in the device is defined as $1/(n_{slot} \times t^*_{slot})$, where n_{slot} is the number slots required for the transmission of a packet which is derived in 3.6.

$$n_{slot} = \frac{(2 - 4P_c + 2(P_c)^2 - 2^m(P_c)^{m+1})CW_0}{2 - 6P_c + 4(P_c)^2} \tag{3.6}$$

The probability P_{np} that at least one packet arrives during t^*_{slot} can be obtained in 3.7. Considering a given number of devices in a vAG for an unsaturated network, the equations 3.2, 3.3, 3.4, 3.5 and 3.7 represent a system of non-linear equations which can be solved numerically to obtain the P_c, τ^{us}, P_{na}, P_{ba} and P_{np}.

$$P_{np} = 1 - e^{-\lambda_d \times t^*_{slot}}. \tag{3.7}$$

Based on the probabilities solved numerically, the probabilities that describe the fraction of the transmission time or payload for obtaining the throughput performance can be derived. The first probability that characterises the MAC throughput performance is the probability that there is at least one active or busy transmission in a time slot denoted as P_{at}. This is defined in 3.8 which is based on the fact that there is d number of devices that contend for a transmit opportunity with at least one device having τ^{us} chances of transmitting data.

$$P_{at} = 1 - (1 - \tau^{us})^d \tag{3.8}$$

The probability P_{at} is used to derive the probability that there is a successful transmission in a RAW time slot denoted as P_{su}. The probability P_{su} is defined in 3.9 which is based on the fact that there is at least one device and exactly one device transmitting in the channel.

$$P_{su} = \frac{d\tau^{us}(1 - \tau^{us})^{d-1}}{P_{at}} = \frac{d\tau^{us}(1 - \tau^{us})^{d-1}}{1 - (1 - \tau^{us})^d} \tag{3.9}$$

The throughput for an average generic slot transmission S_{tr} which is also the throughput of a single vAG in bits per second based on the probabilities is presented in 3.10. In 3.10, L_{pl} is the average length of the packet payload in bits, T_{su} is the duration of a busy channel for the transmission of all frames successfully and T_{cl} is the duration of a busy channel due to a collision of a frame, all in seconds. The parameter t_{slot} represents the duration of an empty slot

which corresponds to the duration of the BO slot. The denominator of 3.10 defines the average generic time slot duration t^*_{slot}. It is important to highlight that 3.10 is similar to Dao et.al's model with a modification to the number of devices per slot which is based on the number of RAWs and T_{su} as defined based on the PS-Poll channel access mechanism.

$$S_{tr} = \frac{P_{su}P_{at}L_{pl}}{(1-P_{at})t_{slot} + P_{su}P_{at}T_{su} + P_{at}(1-P_{su})T_{cl}} \tag{3.10}$$

The throughput model in 3.10 is generic and therefore can be defined according to a specific media access transmission procedure. Therefore, S_{tr} is defined by the media access transmission mode for S1G IoT devices in the 802.11ah standard described prior. To achieve this, the duration of a successful transmission T_{su} and the duration of a collision T_{cl} are defined in 3.11 and 3.12 respectively and are based on the non-TIM channel access transmission procedure illustrated in Figures 3.2 and 3.3, where T_{pol} is the PS-Poll frame transmission time, T_{hdr} is the header transmission time, T_{pl} is the payload frame transmission time, T_{ak} is the ACK transmission time, T_{sifs} is the SIFS period, T_{pr} is the prorogation delay, T_{difs} is the DIFS period and $T_{ak_{to}}$ is the acknowledgement time out time.

$$T_{su} = T_{pol} + T_{hdr} + T_{pl} + 2T_{ak} + 3T_{sifs} + 4T_{pr} + T_{difs} \tag{3.11}$$

$$T_{cl} = T_{pol} + T_{pr} + T_{ak_{to}} + T_{difs} \tag{3.12}$$

In a given RAW slot, a device has to pause its transmission for a given period before the end of the RAW slot boundary to avoid the transmission from crossing over to the next RAW slot. This period is referred to as the RAW slot holding time [87]. The RAW holding time means that the RAW slot duration is reduced which in turn reduces the throughput. To factor in the effect of the holding time, the parameter η_{rw} is used to represent the fraction of the RAW slot that is used as the fixed holding period. In addition, the RF virtualisation also imposes some

delays due to the multiplexing of signals from the different vAGs in the PHY layer. Therefore this effect reduces the throughput by a factor η_{ag} and this factor increases with two or more vAGs. Therefore, the RAW slot throughput for a given vAG taking into consideration the holding time can be defined using 3.13 as proposed in [50] which is extended to include the effects of the vAGs. It is worth noting that, when n_{ag} is equal to 1, 3.1 and 3.13 conforms with the model for the classical RAW throughput (without vAGs) in [50].

$$S_{rw} = S_{tr}(1 - \eta_{rw} - \eta_{ag}(n_{ag} - 1)) \tag{3.13}$$

3.3.2 Throughput with Gateway MAC Constraints

In this section, the GW MAC functions are considered in the throughput model whereby the GW MAC processing and buffer resource constraints are modelled. As depicted in Figure 3.5, it is considered that a seemingly parallel stream of PHY layer data each associated with a vAG is handled by the GW MAC. To model the effects of the GW MAC constraints on the MAC protocol's throughput performance, a typical GW MAC functionality is described and used based on the high-level structure in Figure 3.6. In a given time slot, a scheduler schedules the GW MAC Receive (Rx) controller to read a frame of data from a vAG with available frames in the PHY layer. This could either be a PS-Poll frame or a Data frame. The frame is then stored in a memory buffer called the Rx buffer. The GW MAC core is then notified by the Rx controller that there is a frame available in the receiver buffer. The GW MAC core then fetches the data from the RX buffer and executes an algorithm for error detection such as Cyclic Redundancy Checks (CRC). The GW MAC core also checks if the frame received is intended for the GW and then invokes any other necessary or proprietary process attributed to the reception of frames. The core then generates an ACK frame which is placed in the transmit buffer and the transmit controller then sends the frame to the PHY for transmission.

The GW MAC process described is executed for one vAG in a single timeslot. Therefore, only one frame can be read and processed from the PHY layer at a time during an active slot serving

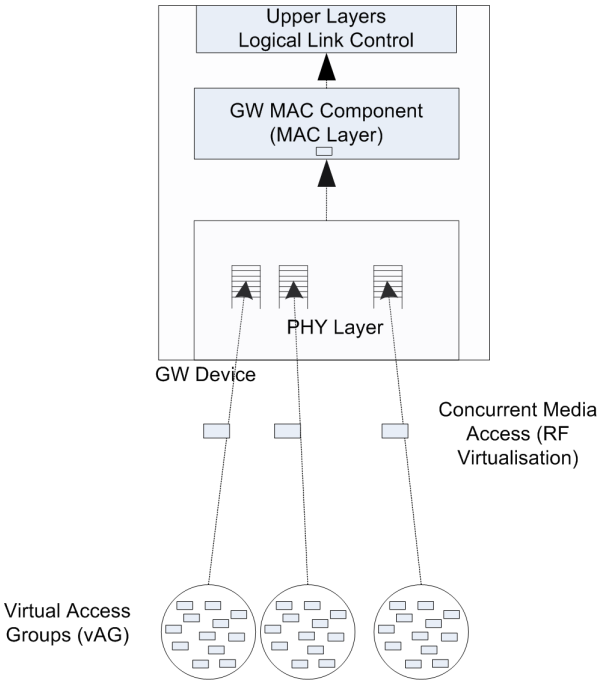

Figure 3.5: GW MAC component with concurrent media access.

multiple independent vAG transmissions. If only one vAG is considered, this may not be a problem. However, with increased PHY channel concurrency (the frames from multiple vAG PHY layer streams arriving concurrently in a timeslot), the traffic arrival rate at the GW MAC is bound to be very high. This is coupled with limited resources in the GW MAC component which can lead to increased congestion or bottlenecks and other performance-related issues. To introduce the GW MAC constraints, the probability of a frame being blocked or dropped in the GW MAC component due to the constrained GW MAC buffer capacity and the response time of the entire GW MAC component is modelled.

To model the buffer constraints and total response time, the M/M/1/c queuing model is used to represent the GW MAC component. The M/M/1/c queue represents frames that arrive exponentially at the GW MAC buffer (from multiple vAGs) with a finite queue capacity of c_g, and a single server (GW MAC core) with an exponentially distributed GW MAC processing rate. The probability that a frame will be blocked due to a constrained GW MAC buffer can be denoted as P_{bk}^{bf} and computed using 3.14, which is the blocking probability of an M/M/1/c queue [88]. The parameter ρ_{gm} is the utilisation of the GW MAC queue.

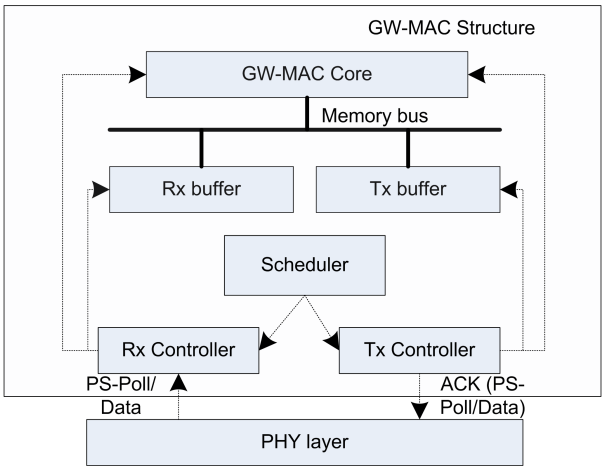

Figure 3.6: High-level representation of the GW-MAC functionality of a GW

$$P_{bk}^{bf} = \rho_{gm} \frac{1 - \rho_{gm}}{1 - (\rho_{gm})^{c_g+1}} \tag{3.14}$$

The utilisation ρ_{gm} can be derived as λ_{gm}/μ_{gm} where λ_{gm} is computed using 3.15 and μ_{gm} is the GW MAC processing rate. In 3.15, the frame arrival rate is the superposition of the arrival rates from the different vAG (the rate at which packets are generated in each device of a specific vAG) conditioned around the probability that there is at least one busy transmission (successful contention) in the r^{th} vAG.

$$\lambda_{gm} = \sum_{r=1}^{n_{ag}} P_{at}^{r} \lambda_d \tag{3.15}$$

Based on the normal operation of a processor architecture [89] the average frame processing rate can be obtained in 3.16. By considering that one instruction is executed per CPU clock cycle for a l_{cg}-bit architecture, the total number of instructions required to execute a frame is equivalent to the average GW MAC frame length processed by the GW MAC core denoted as L_{mc} divided by l_{cg} bits. By dividing the number of instructions per frame by the processing capacity C_{gm} in Million Instructions Per Second (MIPS) and considering the added delay T_{gm} due to memory access, interrupts, schedulers and other events, the frame processing rate in the

GW can be derived as in 3.16. The parameter T_{gm} depends on the design architecture of the GW MAC and underlying hardware.

$$\mu_{gm} = \frac{1}{\frac{L_{mc}}{l_{cg}C_{gm}} + T_{gm}} = \frac{C_{gm}}{L_{mc}/l_{cg} + C_{gm}T_{gm}} \tag{3.16}$$

The overall response time of the GW MAC denoted as T_{rp}^{gm} given the average GW MAC frame length can be obtained using the expression for the response time of a $M/M/1/c$ queue [88] as presented in 3.17.

$$T_{rp}^{gm} = \frac{1}{\mu_{gm} - \lambda_{gm}} \frac{1 - (c_g + 1)(\rho_{gm})^{c_g} + c_g(\rho_{gm})^{c_g+1}}{1 - (\rho_{gm})^{c_g}} \tag{3.17}$$

Taking the response time into consideration, it is possible to define another probability based on the GW MAC core response time. A transmission is considered a failed transmission if the time it takes to process a frame and transmit back an ACK frame is more than $T_{ak_{to}}$. Therefore, the probability that a transmission will fail due to the GW MAC response time can be defined in 3.18 as the ratio of the GW MAC response time and the ACK deadline less the amount of time already used up by other transmission activities between the device and the GW in a slot.

$$P_{bk}^{ak} = \begin{cases} \frac{T_{rp}^{gm}}{T_{ak_{to}} - (T_{sifs} + 2T_{pr})}, & T_{rp}^{gm} \leq (T_{ak_{to}} - (T_{sifs} + 2T_{pr})) \\ 1, & otherwise \end{cases}, \tag{3.18}$$

From the probabilities derived so far, the probability of a successful transmission in a given time slot in any of the vAGs depends on three events. These are a collision-free transmission, a frame that is not dropped due to the GW MAC buffer constraints and the response time of the frame that is less than the required time out. Therefore, 3.9 can be modified to give P_{su}^{gm} in 3.19. This applies to either the PS-Poll or the Data frames transmitted by the device

since the average GW MAC frame length is used to derive the probabilities of the GW MAC constraints.

$$P_{su}^{gm} = P_{su}(1 - P_{bk}^{bf})(1 - P_{bk}^{ak}) \tag{3.19}$$

Having obtained the probability of a successful transmission based on the GW MAC constraints, the duration of a successful transmission in a given time slot in 3.11 can be modified to incorporate the GW MAC delay T_{rp}^{gm} as expressed in 3.20. The processing delay is doubled to represent the average delay for both the PS-Poll and Data frames.

$$T_{su}^{gm} = T_{pol} + T_{hdr} + T_{pl} + 2T_{ak} + 3T_{sifs} + 4T_{pr} + T_{difs} + 2T_{rp}^{gm} \tag{3.20}$$

The throughput model per vAG defined in 3.10 can be modified to give S_{gm} in 3.21 which incorporates P_{su}^{gm} and T_{su}^{gm}. It is important to highlight that the probability of a successful transmission and the probability of a failed transmission is conditioned around the probability that there is at least one transmission during a time slot. The 802.11ah channel access mechanism cannot explicitly detect the type of transmission failure (thus, whether it's a collision, corrupted packet, blocked packet etc.) hence any failed frame transmission is regarded as a collision. Therefore the portion of the slot duration that is wasted when a transmission is unsuccessful due to a collision or GW MAC constraints can be taken as T_{cl}. The RAW throughput based on the GW MAC constraints can simply be computed using 3.13 when S_{gm} is obtained. This is the RAW slot throughput in each vAG.

$$S_{gm} = \frac{P_{su}^{gm} P_{at} L_{pl}}{(1 - P_{at})t_{slot} + P_{su}^{gm} P_{at} T_{su}^{gm} + P_{at}(1 - P_{su}^{gm})T_{cl}} \tag{3.21}$$

3.4 Numerical Experiment, Results and Analysis

In this section, the computation of the models studied is presented based on a numerical experiment. The results obtained from the computation of the models are presented and analysed to validate and verify the behaviour of the model. The computation of the model was performed using MATLAB R2022a.

3.4.1 Setup and Parameters

The computation and evaluation are conducted based on four different traffic rates representing different LS-IoT applications or use cases such as in [90]. In table 3.1 the four different LS-IoT traffic types labelled as traffic A, B, C and D are presented based on the average length of the IoT/M2M payload and average time interval at which the payload is generated. The inverse of the arrival interval gives the traffic arrival rate of a device (λ_d) for each IoT application. As depicted in table 3.1, traffic A can be attributed to applications such as Industrial Automation (IA) because it typically has more data to be transmitted at relatively short intervals on average and such data may be highly critical in controlling machines, robots, cameras for certain operations. Traffic B can be attributed to applications such as Logistics and Transportation (LT), which typically have shorter data payload and slightly longer transmission intervals on average than IA because data may require short images and numerous sensor data about the cargo condition among others not very critical but important. Traffic C may be attributed to Smart Environmental or Agricultural monitoring (AM) applications which may also have a much shorter average data payload and longer transmission interval than IA and LT. Lastly, traffic D may represent Smart Metering (SM) applications which can be attributed to very short data payload but very long transmission intervals on average than the other applications and are not critical. The length of the header is constant for each payload based on the 802.11ah standard.

A summary of the rest of the parameters used for the computation of the model is presented in table 3.2. Some of the parameters not presented in the table are varied to evaluate the effects

Table 3.1: Experimental parameters for the mean traffic rate for the different LS-IoT applications.

LS-IoT type	**Traffic**	**Payload (bytes)**	**Arrival Interval**
Traffic A		120	10 ms
Traffic B		80	100 ms
Traffic C		50	1 s
Traffic D		20	15 s

on the models' behaviour. In table 3.2 the channel rate is chosen to represent the PHY layer characteristics of the IEEE 802.11ah Sub 1 Gigahertz (S1G) channel for a 1 MHz bandwidth using the modulation and coding scheme (MCS) index 4 based on a 16-QAM modulation type as provided in [91]. The length of the PS-Poll frame and ACK frame are based on the maximum values proposed by the 802.11ah standard. The empty time slot, DIFS, SIFS and ACK time-out durations also correspond to the proposed values in the 802.11ah standard. The propagation delay value is chosen as a typical average value used for characterising IEEE 802.11 network outdoor environments [92], including the 802.11ah standard which targets a range of up to 1 km in the S1G frequency channel [93]. The minimum and maximum contention window values are chosen to correspond to 5 maximum BO stages. The GW MAC processing rate, buffer capacity and the additional delay are arbitrarily chosen.

3.4.2 Analysis of Results

3.4.2.1 Throughput Results without Gateway MAC Constraints

In Figure 3.7 the media access throughput without GW MAC constraints of 4 vAGs with RAW is plotted against the total number of devices connected to the GW for the different LS-IoT traffic types in each subplot. The results are compared with the Bianchi saturated model and the Dao et al. unsaturated model both without the RAW mechanism. The number of devices dictates the probability of having at least 1 transmission for each vAG. It can be observed for all the traffic types that while the Bianchi model initially peaks at 5 devices and decreases, the Dao et al. model increases linearly until the peak then decreases and converges with the

Table 3.2: Experimental parameters for evaluating the MAC Throughput Model

Parameter	**Value**
Channel Rate (R_{ch})	2.0 Mb/s
MAC Header length (L_{hdr})	16 Bytes
PS-Poll frame length (L_{pol})	20 Bytes
ACK frame length (L_{ak})	14 Bytes
DIFS duration (T_{difs})	264 μs
SIFS duration (T_{sifs})	160 μs
Empty Slot duration (t_{slot})	52 μs
ACK time out ($T_{ak_{to}}$)	$T_{sifs} + (2 \times t_{slot})$
Propagation delay (T_{pr})	1 μs
Min. Contention Window (CW_0)	32
Max. Contention Window (CW_m)	1024
RAW holding time delay factor (η_{rw})	0.05
RF virtualisation delay factor (η_{ag})	0.01
GW MAC Processing Capacity (C_{gm})	1000 MIPS
Number of bit per Instruction (l_{cg})	64
GW MAC buffer Capacity (c_g)	20 MAC frames
GW MAC additional delay (T_{gm})	5 μs

Bianchi model. For traffic D, the peak occurs at a much higher number of devices than the other traffic types. This behaviour is consistent with the saturated and unsaturated conditions of the Bianchi and Dao et. al. model respectively because, for a saturated condition, the devices continuously have data to be transmitted hence the collision probability increases with only a few devices irrespective of the traffic type. However, for the unsaturated case, the linear increase in throughput implies a reduced probability of collision due to devices contending for channel access based on the varying traffic rates or interarrival times.

In comparing the reference models with the 802.11ah RAW model with vAGs, the 802.11ah LS-IoT model based on the RAW with 4 vAGs shows a relatively slow rate in the decline of the throughput as the number of devices increases compared to all traffic types. Also, for the lower traffic rates (Traffic C and D), the throughput peaks when the number of devices is relatively high before decreasing. The rate of decline in throughput is much slower when the number of RAWs is increased. This behaviour is consistent with the scaling effect of the RAW and vAG

groupings. This, therefore, reduces the probability of a collision and improves the scalability within a slot. The LS-IoT throughput with vAGs represents the ideal throughput if multiple concurrent transmission using vAGs is considered with ideally no GW MAC constraints.

3.4.2.2 Failure Probability with and without Gateway MAC Constraints

In Figure 3.8 the results of the probability that a transmission will fail when the GW MAC constraint is considered and when the GW MAC constraint is not considered are plotted against the number of devices for the different traffic types. The number of vAGs is varied between 2 and 6 to observe the behaviour. The results were obtained based on 4 RAWs.

For the GW MAC without any constraints using 6 vAGs, the probability of a failed transmission versus the number of devices is relatively low. This is because the GW MAC constraints are not taken into account hence the only effect on the transmission failure is the probability of collisions which is also reduced significantly due to the grouping of devices into vAGs.

For the GW MAC constrained case using 6 vAGs, it can be observed that the probability of a failed transmission versus the number of devices is higher than the unconstrained case for traffics A and B because of the effects of the GW MAC constraints on high aggregated traffic rates which increases the blocking of frames. For traffic C, the probability only increases significantly when there are more devices based on 6 vAGs whereas, for traffic D, the probability remains constant at 0 for the number of devices based on 6 vAGs. This is because of the reduced effects of the GW MAC constraints due to much lower aggregated arrivals of frames attributed to traffics C and D.

It can also be observed that when the number of vAGs is low (2 vAGs), the probability of a failed transmission versus the number of devices increases when compared to 6 vAGs for all the traffic types. This is because there are more devices per vAG when using 2 vAGs than using 6 vAGs, which therefore increases the probability of having concurrent transmission for all vAGs and hence a failed transmission.

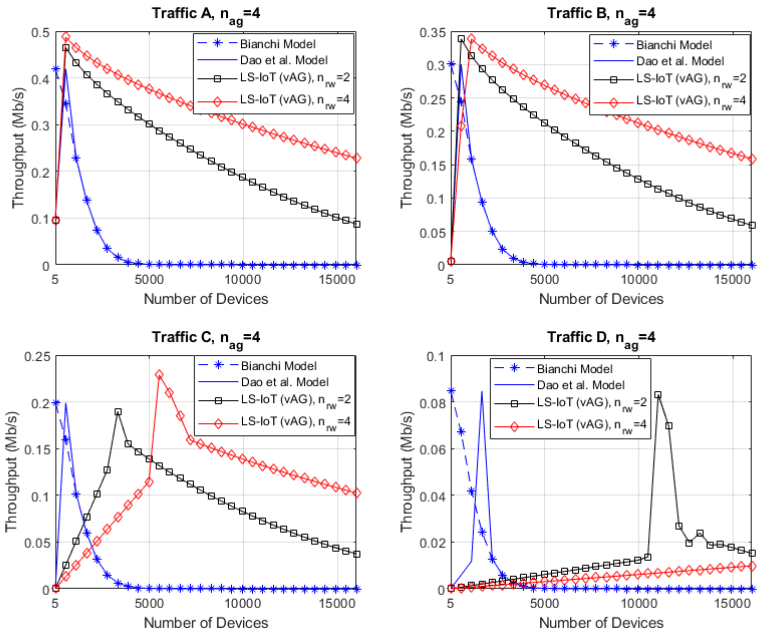

Figure 3.7: Media access throughput based on four (4) vAGs with RAW versus the number of devices for the different traffic types, compared with reference models.

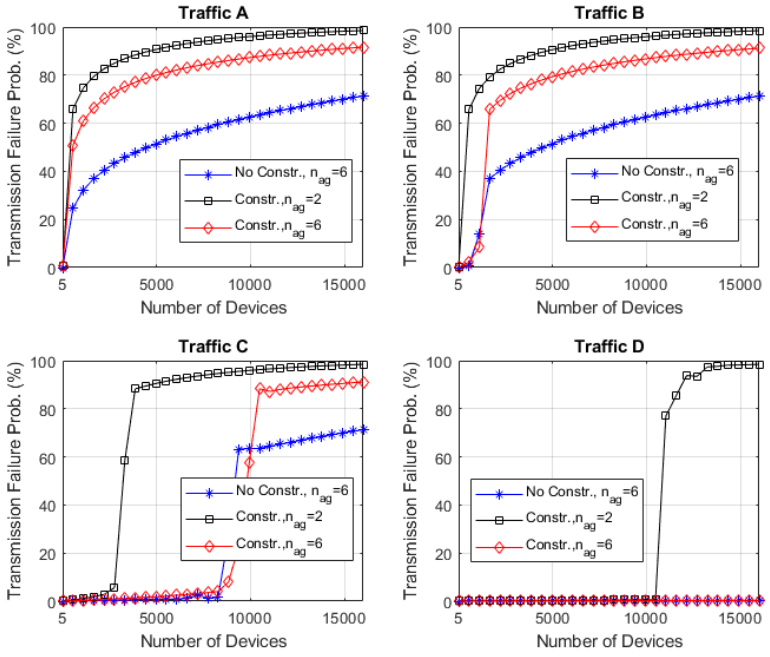

Figure 3.8: Transmission failure Probability with and without the GW MAC constraints versus the number of devices for the different traffic types.

3.4.2.3 Throughput Results with and without Gateway MAC Constraints

In Figure 3.9 the results of the MAC throughput with the constrained GW MAC versus the number of devices in the network are presented for the different traffic types in each subplot. In each subplot, the constrained case is compared with the throughput without GW MAC constraints. The results were obtained based on 4 RAWs.

For the GW MAC without any constraints using 6 vAGs, the throughput for traffic A and B drops with fewer devices because of the high traffic rate attributed to traffic A and B. However, for traffic C the throughput drops when the number of devices is more and for traffic D, the throughput continues to increase linearly and does not reach the peak for the plotted number of devices. This is because of the low traffic rate attributed to traffic C and D.

For the GW MAC constrained case using 6 vAGs, it can be observed that the throughput performance for traffics A and B is poor (throughput drops rapidly with fewer devices) when compared with the unconstrained case also with 6 vAGs. This is due to the effects of the GW MAC constraints on the aggregated traffic arrivals which increase the transmission failure probability as well as the duration of a successful transmission with fewer devices. For traffic C, the throughput with 6 vAGs drops when the number of devices is more (the same as the unconstrained case), However, the throughput drops far below that of the unconstrained case. For traffic D the throughput increases linearly and the peak throughput is not reached with 6 vAGs. This is because of the low traffic rate attributed to traffic C and D which results in the aggregated traffic arrivals being high only when there is a large number of devices.

It can also be observed that when the number of vAGs is low (2 vAGs), the throughput performance is poor (throughput drops with fewer devices) when compared to 6 vAGs for all the traffic types. This is because there are more devices per vAG when using 2 vAGs than using 6 vAGs, which therefore increases the probability of a collision as well as a frame being blocked due to GW MAC constraints and hence a failed transmission.

In Figure 3.10, the GW MAC core processing capacity is varied to further observe the behaviour of the throughput using 6 vAGs. It is observed that for traffic A, B and C, when the GW MAC core processing capabilities are relatively low (1 MIPS), the throughput performance versus the number of devices drops when compared to 1000 MIPS. This is because of the increased response time and the frame blocking (buffer constraints) attributed to a low processing capacity for the aggregated traffic rate. For traffic D, the throughput increases linearly without reaching the peak for constrained and unconstrained cases irrespective of the GW MAC core processing capacity. This is due to the very low traffic rate attributed to traffic D. In general, for both 1000 MIPS and 1 MIPS the throughput performance is still degraded in comparison to the ideal case of having no GW MAC constraints.

3.5 Conclusion

In this chapter, the 802.11ah media access scheme for an unsaturated LS-IoT network was modelled based on the RAW without vAGs (classical case) and the RAW based on vAGs with and without GW MAC constraints. The results obtained in the study conform with the classical throughput scalability behaviour of a media access scheme and in particular the 802.11ah RAW MAC scheme as compared with a reference model. Overall, the results suggest that, although increasing the number of vAGs (high concurrent transmissions) improves the MAC throughput, the GW MAC constraints degrade the throughput performance, especially for high IoT traffic rates. Therefore, the classical approach of having a single dedicated GW MAC with resource constraints can not support the scalability of the media access in terms of provisioning concurrent vAGs. Also, it can be concluded that a high GW MAC processing capacity can increase the throughput slightly. A suitable approach must be proposed to support the GW MAC scalability based on virtual access groupings. A GW MAC virtualisation framework is proposed and modelled to evaluate the possibility of supporting the scalability of the LS-IoT network with multiple vAG in the next chapters.

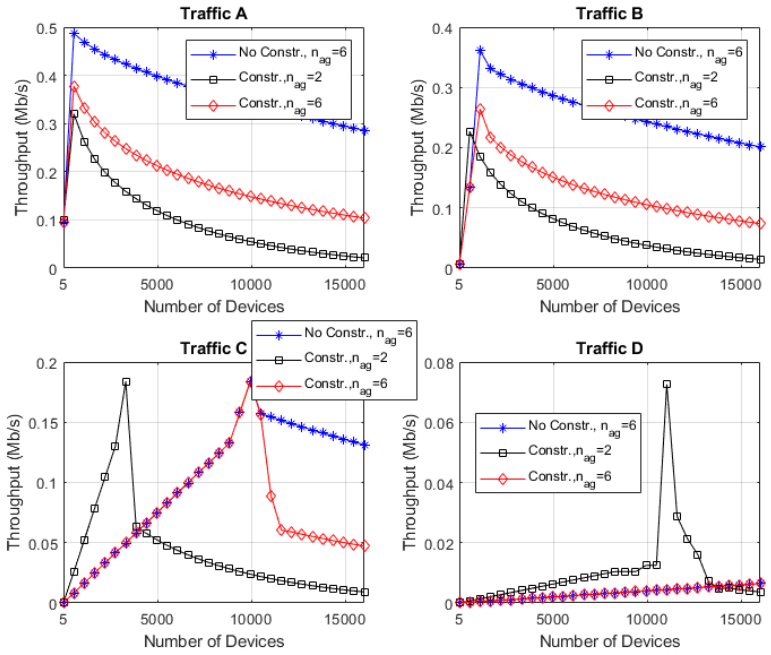

Figure 3.9: Media access throughput with the GW MAC constraints versus the number of devices for the different traffic types, compared with the unconstrained case.

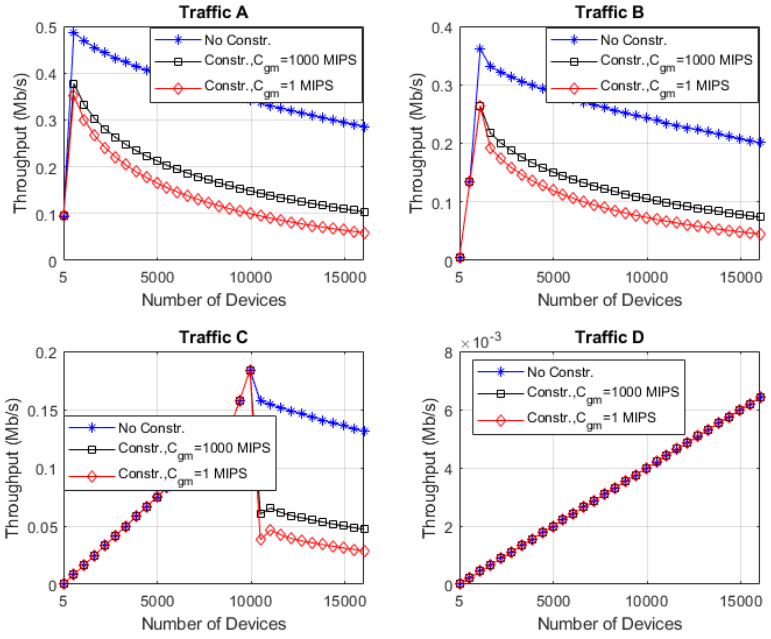

Figure 3.10: Media access throughput with different GW MAC core processing capacities based on the GW MAC constraints versus the number of devices for the different traffic types, compared with the unconstrained case. ($n_{ag} = 6$)

CHAPTER 4. A VIRTUAL NETWORK MODEL FOR GW MAC VIRTUALISATION IN LARGE-SCALE IOT

4.1 Introduction

A virtual network brings together different network components to form a single entity for sharing of resources. Therefore, using a virtual network to virtualise the GW MAC functions may address the GW MAC constraints and improve the scalability. However, the virtual network components and their associated functions introduce some performance degradation which is very critical to consider when virtualising the GW MAC functions. Each virtual network node or entity exhibits some level of performance such as resource availability, utilisation and response time. An unsuitable virtual network design can drastically affect the performance of the entire virtualisation process. In the context of this study, this can counter the objective of scaling the GW MAC functions to support multiple virtual Access Groups. Existing models like in [94], omit the MAC related functions like ACK transmissions from the virtualisation process due to the strict timing deadlines that cannot be met over the virtual network. Also, some virtualisation components such as protocol translation or encapsulation, switching and resource provisioning are omitted from the virtual network model to simplify the models. These omissions do not provide a true representation of the virtualisation process. Therefore a virtual network based on the fundamental nodes and functions necessary to achieve full GW MAC virtualisation for LS-IoT is imperative. It is important to develop an analytical model that can estimate the performance metrics at each virtual network node and the whole network. However, this is a key challenge due to the complexity of the GW MAC virtualisation process. In this chapter, the main aim is to develop a model which appropriately represents the virtual network nodes and processes necessary to establish the GW MAC virtualisation performance metrics and evaluate the impact on the media throughput. To achieve this aim, the following

objectives are achieved in this chapter:

a. To develop a virtual network framework which encompasses the relevant components and processes for virtualising the GW MAC scheduling function based on the CPRI over Ethernet Encapsulation process, SDN Priority-based traffic classification and Switching and the edge data centre resource provisioning capabilities.

b. To employ a queuing network approach to analytically model the virtual network framework in terms of the utilisation and the response time of each node and the entire virtual network as well as the overall virtualisation delay.

c. To evaluate the impact of a virtualised GW MAC component on the throughput of the 802.11ah MAC scheduling mechanism by modifying the throughput model to incorporate the virtualisation delay.

4.2 Proposed Virtual Network Architecture

A simple illustration of the GW MAC virtualisation process based on one GW node and one eDC node is represented in Figure 4.1. The virtualisation of the GW MAC is based on the creation of multiple instances which are deployed in the eDC where there are enough resources to handle multiple GW MAC instances. In this chapter, the focus is on the virtual network link design based on multiple eDCs for supporting multiple GW nodes with virtualised GW MAC instances.

The proposed virtual network architecture for virtualising the GW MAC is depicted in 4.2. The virtual network infrastructure is typically owned and managed by a Virtual Network Operator (VNO). The M2M devices are connected to a physical gateway node (pGW) over a wireless link that supports heterogeneous applications. More than one pGW node is serving different sectors which may belong to different IoT service providers. The pGW nodes are equipped with a radio and only perform the PHY layer functions with the assumption that RF virtualisation based on spectrum slicing is supported for multiple concurrent transmissions for multiple vAGs.

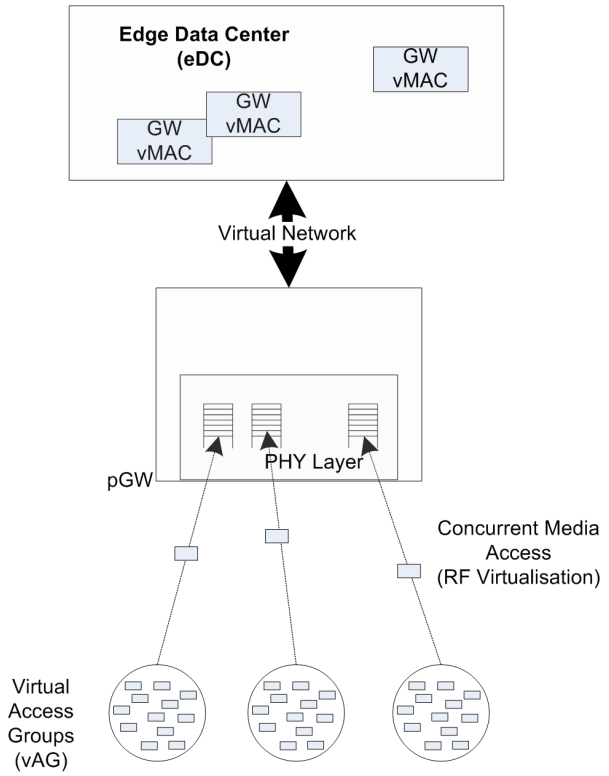

Figure 4.1: A simple Virtualised GW MAC Scenario

The GW MAC functions are decoupled and deployed on one of the eDCs. For the decoupling to be possible, the pGW node must first encapsulate the PHY layer data into virtual frames supported by the virtual network. It is important that this process is lightweight to minimize delays. Therefore the Common Public Radio Interface over Ethernet (CPRIoE) is proposed. The CPRIoE defines virtual connections between endpoints of a front-haul communication link for Cloud Radio Access Network (CRAN) technologies through multiplexing different data streams onto an optical fibre using the Time Division Multiplexing (TDM). It is capable of supporting modulation data and other real-time baseband signal processing functions. This makes it highly suitable for GW MAC virtualisation in terms of guaranteeing timing deadlines since the MAC functions are not as time-sensitive as baseband signal processing. Therefore, the pGW node encapsulates the PHY data into CPRIoE frames to create GW virtual MAC (vMAC) frames. The vMAC frames are transmitted over optical fibre to an SDN switch managed and owned by the VNO. The VNO may have more than one pGWs that is interfaced with an SDN switch serving as an aggregation node for multiple IoT service providers. The

SDN switch comprises a store-forward buffer, a classifier for grouping the frames according to the LS-IoT traffic type and a scheduler for routing vMAC frames to the intended eDC node. The scheduler component is controlled by the SDN control plane which is managed by a remote SDN application. The SDN application abstracts the entire virtual network components and may employ heuristics to guarantee QoS, however, this aspect is not studied in this chapter. The vMAC frames are transmitted to the eDC over an optical fibre link. The eDC creates VNF instances that define the functions of each vMAC instance. The vMAC instances, therefore, perform the GW MAC functions and forward the vMAC frame to the IP-based back-haul network.

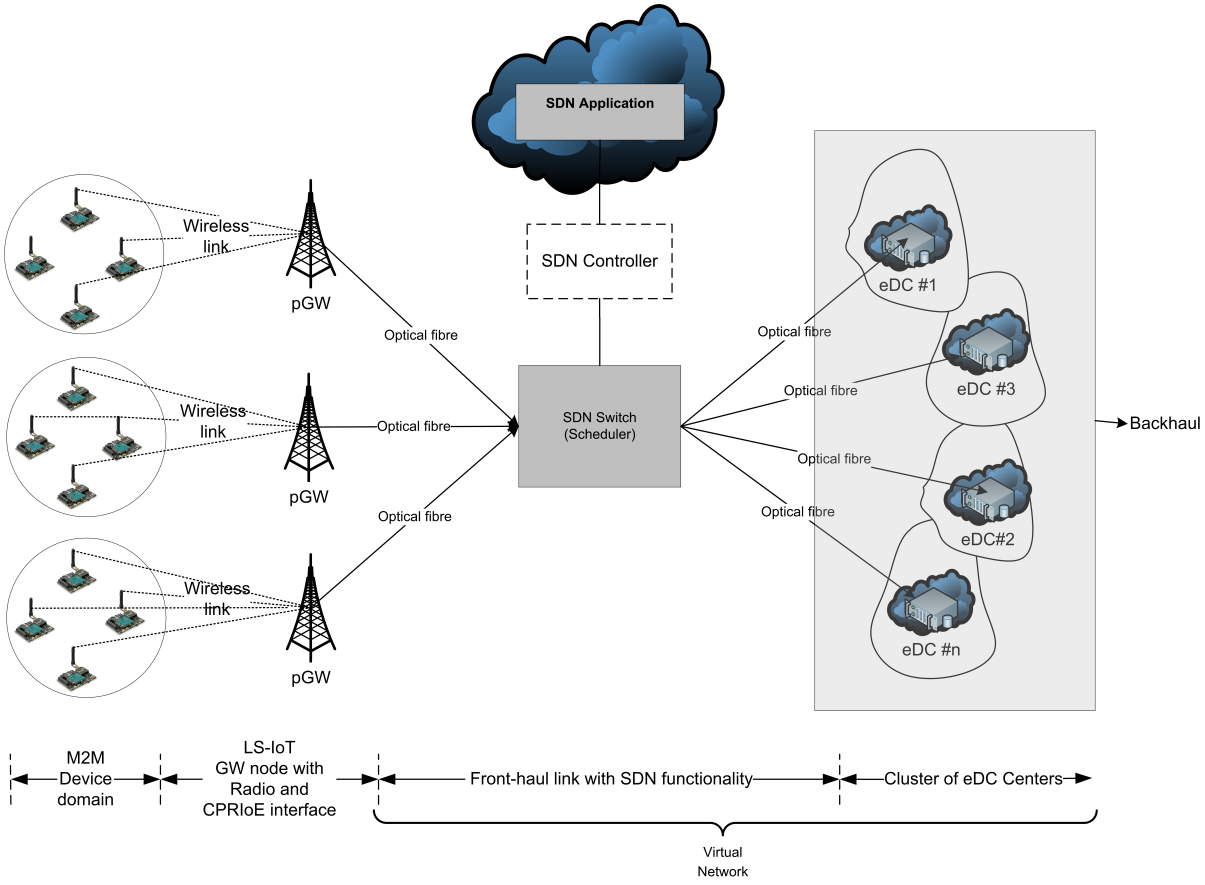

Figure 4.2: Proposed virtual network architecture for GW MAC virtualisation in LS-IoT.

4.3 Analytical Model

The key objective of the analytical model is to find the utilisation and the response time at each node and the whole virtual network architecture proposed. As described previously, the

vMAC frames move from one node to another within the virtual network, hence it is important to consider the traffic arrival rate and variation, the service rates, and the routing of traffic from one node to another. In the queuing network, there is a likelihood of a strong correlation between the arrival times in the downstream queues and departure times in the upstream queues. This makes the estimation of the queueing delay in the system complex since there is no known exact analytical solution for solving such a problem. There are some classical approaches and extended approaches used to approximate the variability in the interarrival and departure times of a network of queues under certain conditions. Some of these approaches found in [95] and [96] include, the Kleinrock approximation which is applied to an M/M/1 queueing model based on the Jackson theorem; Baskett, Chandy, Muntz and Palacios (BCMP) model which is an extension of the Jackson theorem; and Kingmans exponential law of congestion. To solve the complexity of estimating and analysing the stochastic behaviour of a network of queues, an approach called the QNA for modelling a network of queues is proposed by in [97]. The QNA model is employed in a software package developed at Bell Laboratories. QNA has been extensively used in several theoretical and industrial applications [98] where the results have been compared with simulations and other approaches [99]. QNA is a powerful tool and has a relatively very low error margin which makes it ideal for modelling queuing networks.

The QNA method is employed to model the virtual network by first translating the architecture in Figure 4.2 into an appropriate queuing network model as shown in Figure 4.3. Thus, a network of queues and servers is formulated such that the vMAC traffic stream with frames departing from one queue arrive at another queue based on the defined traffic flow. In modelling the virtual network, the function of each node is first modelled. The QNA approximation is then employed to derive the traffic rate and traffic variations of the queueing network. Based on this, the virtual network performance metrics are modelled for each node and the entire network.

In this study the uplink (UL) direction is considered because most LS-IoT applications are generally based on the transmission in the uplink direction [100] and it imposes congestion on

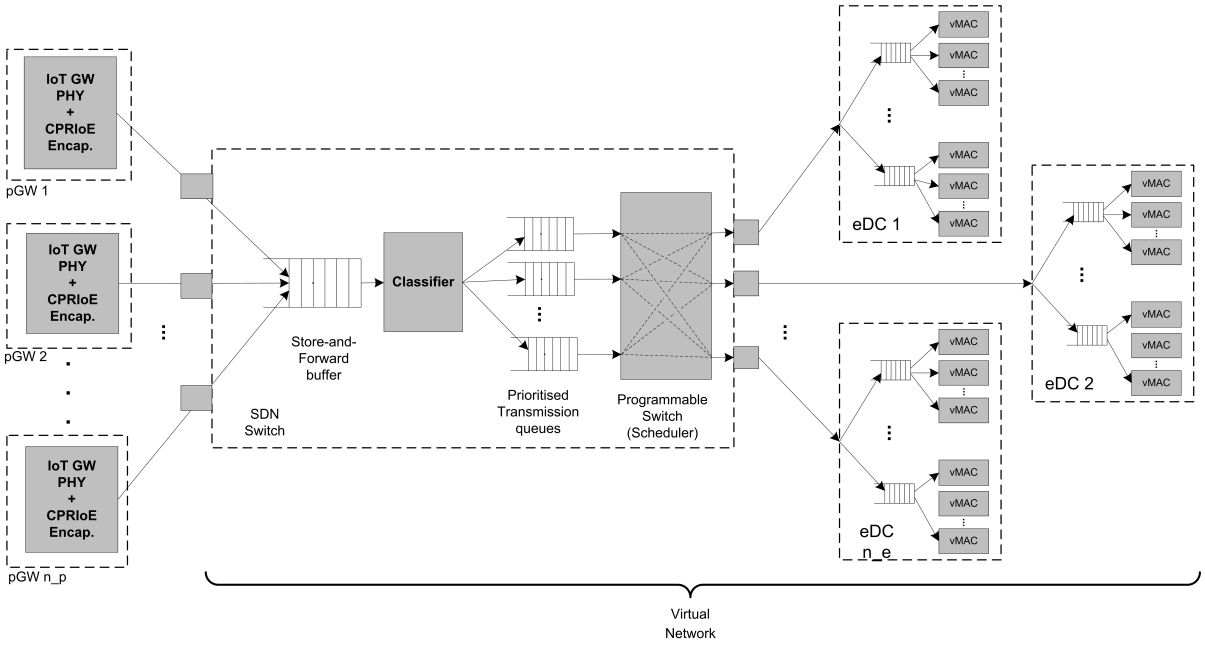

Figure 4.3: A queueing network model representation of the proposed virtual network architecture for GW MAC virtualisation.

the GW MAC [101], [102], [103]. However, it is also assumed that the average virtualisation delay in the forward direction (pGW to eDC) is approximately the same as in the reverse direction (eDC to pGW). This helps simplify the estimation of the delay in transmitting back an ACK frame to the pGW. The optical fibre transmission is assumed to guarantee near real-time transmission with negligible transmission delays, hence it is not considered in the model. The CPRIoE encapsulation rate is assumed to be the same as the decapsulation rate. For all the queues, it is assumed that the frames are scheduled in a First-In-First-Out (FIFO) order; the distribution of the interarrival times and the service times are statistically independent and identical. The queues are considered to have an infinite length, since virtual networks are generally considered as high capacity networks [104], [105].

4.3.1 Virtual Network Nodes

The purpose of this section is to model the processes that define the functions of the different nodes in terms of the average traffic load on the virtual network and the average service rates in frames per second. These are used as inputs for the QNA model presented later.

4.3.1.1 Physical Gateway Node

The objective is to model the CPRoE encapsulation process and derive the pGW's output traffic rate based on the multiplexing of different vAG traffic streams. The pGW is the source of the traffic that goes to the virtual network. Hence the pGW's output traffic rate defines the load of the virtual network. The CPRIoE encapsulation process is executed in the pGW node and the decapsulation process is executed at the eDC node. In the CPRIoE standard [106], the user plane information, in this case, the MAC frames to be virtualised, are encapsulated into the standard Ethernet frame. Therefore, the CPRIoE process of a single stream or flow to be modelled is illustrated in Figure 4.4. The CPRIoE encapsulation processing is made up of the CPRI mapping and the CPRIoE encoding process with different processing rates. The CPRIoE process can be modelled based on the sequential packetisation of CPRI flows into an Ethernet frame without the explicit knowledge of the CPRI data [107].

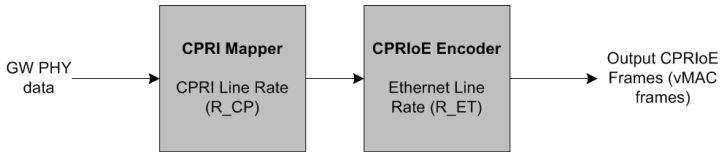

Figure 4.4: The CPRIoE frame generation process for a single flow.

The CPRI mapping process generates n_{bf} number of basic frames at a period of T_{bf} nanoseconds. This implies that, for the average MAC frame length L_{mc} that will be typically processed by the GW MAC, the number of basic frames that will be generated is computed in 4.1 where R_{cp} is the CPRI line rate specified in the CPRI standard [108].

$$n_{bf} = ceil\left\{\frac{L_{mc}}{R_{cp} \times T_{bf}}\right\} \tag{4.1}$$

The payload of the Ethernet frame is therefore made up of n_{bf} CPRI basic frames. Hence, the number of basic frames required to make up the Ethernet payload is such that it does not exceed the maximum length of the Ethernet payload of 1500 bytes. Therefore, the maximum required number of basic frames can be computed as

$$n_{bf}^{max} = ceil\left\{\frac{1500\ bytes \times 8}{R_{cp} \times T_{bf}}\right\}. \tag{4.2}$$

Based on the number of basic frames that are obtained, the duration of the CPRI mapping process T_{cp} can be determined in 4.3. This implies that the CPRI mapper will take T_{cp} seconds to create a CPRI frame made up of n_{bf} number of basic frames which in turn becomes the payload of the Ethernet frame in the CPRIoE encoding stage.

$$T_{cp} = n_{bf} \times T_{bf} = \frac{L_{mc}}{R_{cp}} \tag{4.3}$$

The CPRIoE encoding process encapsulates the CPRI frames into Ethernet frames. The Ethernet payload size is therefore the same as L_{mc}. In order the find the number of Ethernet frames n_{et} required to transmit the payload length, the expression in 4.4 is used. Thus, the actual payload is divided by the maximum Ethernet payload less the CPRIoE header of 6 bytes.

$$n_{et} = ceil\left\{\frac{L_{mc}\ (bytes)}{(1500 - 6)\ bytes}\right\} \tag{4.4}$$

After the CPRI frame is encapsulated into the Ethernet frame, the CPRIoE frame length, which is also the vMAC frame length denoted as L_{vmc} now becomes the sum of the Ethernet frame overhead length L_{eo} and the payload length as computed in 4.5. The Ethernet overhead length is considered as a fixed value of 44 bytes attributed to the standard CPRIoE frame format consisting of a 7-byte preamble, a 1-byte start frame delimiter, 6-byte a source address and a 6-byte destination address, a 2-byte Ethernet type field, a 6-byte CPRIoE header, a 4 byte Frame check sequence and a 12-byte inter-packet gap.

$$L_{vmc} = L_{eo} + L_{mc} \tag{4.5}$$

The CPRIoE encoding (Ethernet encapsulation) time T_{et} for a CPRI frame can be estimated in 4.6 where R_{et} is the Ethernet line rate. Thus, for a L_{mc} length of CPRI frame, the system may require n_{et} number of Ethernet frames for encapsulation. This will require the transmission of the Ethernet overhead n_{et} times.

$$T_{et} = \frac{L_{mc} + n_{et}L_{eo}}{R_{et}}. \tag{4.6}$$

The overall CPRIoE encapsulation rate λ_{vmc} in frames per second for a single stream can be derived in 4.7, where T_{cs} captures a fixed delay as a result of control management and synchronisation of each frame as proposed in [106]. Since T_{cp} and T_{et} are functions of L_{mc}, λ_{vmc} can be derived for any average value of L_{mc}. From 4.7, the encapsulation time per frame denoted as T_{ecp} is simply $1/\lambda_{vmc}$.

$$\lambda_{vmc} = 1/(T_{cp} + T_{et} + T_{cs}) \tag{4.7}$$

As mentioned earlier, the CPRI protocol uses TDM to multiplex different CPRIoE streams onto the transmission medium. Since the pGW node serves different vAGs of different traffic types q, the TDM multiplexer can be used to multiplex n_{ag} number of CPRIoE streams onto the optical fibre medium as depicted in Figure 4.5. Thus, the r^{th} CPRIoE stream corresponds to the encapsulation for the r^{th} vAG where each group belongs to a specific traffic type. Considering that each CPRIoE stream has an average traffic rate of $\lambda_{vmc,r}$ computed using 4.7 based on the average length of the different frames encapsulated for each traffic type, and one CPRIoE frame multiplexed at a time, the mean traffic rate at the output of the multiplexer or the output of the pGW λ_p can be computed using 4.8. This is conditioned around the probability of having at least one transmission for a given vAG.

$$\lambda_p = \sum_{r}^{n_{ag}} (P_{at}^{r} \times \lambda_{vmc,r}) \tag{4.8}$$

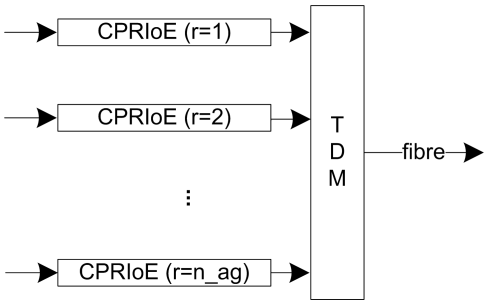

Figure 4.5: Multiplexing of multiple CPRIoE streams onto the transmission medium.

4.3.1.2 Software Defined Networking Switch

The SDN switch comprises two processes. The first process is the classification of the aggregated vMAC frames into the q^{th} prioritised transmission queue. The second process is the scheduling of the frames from the priority-based transmission queues to the output port for transmission. The rate at which the vMAC frames are classified is considered a constant parameter denoted as μ_{cls}. This conforms with the fact that the classifier spends the same amount of time inspecting the frame field irrespective of the size of the frame. In 4.9, the switch's scheduling rate $\mu_{sch,q}$ for a given average vMAC frame length L_{vmc} is presented. It is based on a weighted allocation of the switch's port line rate or bandwidth R_{sw} in bits per second for the q^{th} priority queue using the weighting parameter $w_{rr,q}$. The weight represents the transmission based on a Weighted Round Robin (WRR) process [109].

$$\mu_{sch,q} = \frac{R_{sw}}{L_{vmc}} \times \frac{w_{rr,q}}{\sum_{i}^{n_q} w_{rr,i}} \tag{4.9}$$

4.3.1.3 Edge Data Centre

The objective here is to find the virtual processing rate μ_{vm} of the GW vMAC instance given the average length of the decapsulated GW MAC frame. The virtual processing rate depends on the number of virtual processing resources available in the eDC for virtualisation. The virtual processing resource is modelled similarly to the GW processing resource in MIPS. The resource allocation for a single eDC is depicted in Figure 4.6. It is considered that each eDC

has a virtual processing resource capacity for vMAC virtualisation, denoted as C_e. This value may vary between the different eDC nodes since the nodes are independent of each other. It is also considered that the virtual processing capacity C_e in each eDC is apportioned to vMAC clusters based on the traffic type being processed by the vMAC for service differentiation. This is defined by the weighting factor $w_{vm,q}$. Therefore, considering the average frame length after decapsulation as L_{mc}, the virtual processing rate of a vMAC instance in frames per second in a given cluster $\mu_{vm,q}$ can be derived in 4.10, where the parameter t_{vm} is introduced to represent the impact of the virtual machine scheduling or hypervisor delay on the processing time which could be significant when the number of vMAC instances in the eDC n_{vm} increases. In a typical virtualised environment, the virtual processing resources are not mapped to a dedicated physical processing resource due to performance issues [110]. Therefore, the hypervisor schedules the physical resources dynamically which introduces critical delays such that when a frame is to be processed, the vMAC instance may not be immediately available [111], hence increasing the processing time. The expression in 4.10 implies that the full apportioned virtual processing resource will always be shared equally amongst the vMAC instances $n_{vm,q}$ in that cluster.

$$\mu_{vm,q} = 1/\left(\frac{n_{vm,q}L_{mc}}{l_{cg}C_e w_{vm,q}} + n_{vm}t_{vm}\right) = \left(\frac{l_{cg}w_{vm,q}C_e}{(n_{vm,q}L_{mc} + w_{vm,q}n_{vm}t_{vm}l_{cg}C_e)}\right) \tag{4.10}$$

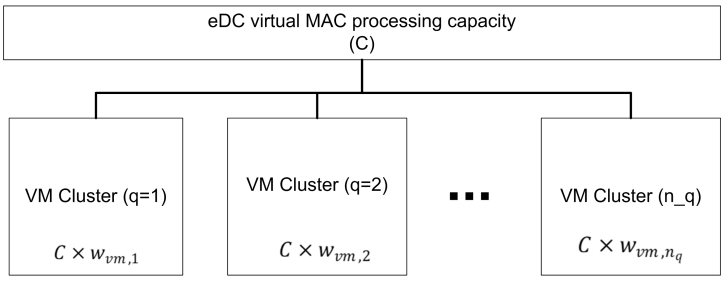

Figure 4.6: eDC virtual MAC processing resource allocation.

4.3.2 Queuing Network Analyser Method

The queueing network in Figure 4.3 is decomposed into individual nodes and modelled as individual G/G/1 or G/G/m queuing nodes. It is important to highlight that if the service time variability and the external arrivals variability approach 1, the G/G/1 and G/G/m queue

act like an M/M/1 and M/M/m queues. The main parameters that are estimated for each node are the traffic arrival rate, the traffic interarrival time variability parameter, the service rate, the service time variability parameter, and the traffic intensities at each node. The QNA estimates the queuing network parameters based on four main traffic flow process operations typical of a network of queues. These operations are namely the splitting process, the superstition process, the departure process, and the arrival process. The QNA then synthesises these models to obtain a system of equations which can be solved using calculus. The different parameters estimated in the QNA method are presented below.

4.3.2.1 Traffic Rates

The objective of this part of the model is to estimate the total traffic arrival rate at each node in the network. The total arrival rate at a node is characterised by external traffic and internal traffic arrivals. External traffic arrivals are arrivals that originate from a source outside of the network. Internal traffic arrivals are traffic arrivals that originate from a source inside the network. For ease of understanding, the queuing network in Figure 4.3 can be said to have nr virtual network nodes, each represented by the index j. The total traffic arrival rate to the j^{th} node is denoted as λ_j and estimated using a system of linear equations given in 4.11 where λ_{0j} represents the external traffic arrival rate to the node; λ_i represents the traffic arrival rate at the node i where the traffic leaves to join the node j; h_{ij} the transition probability which defines the probability that a frame that has completed service at i will go to node j. γ_i is used to represent the creation or combination of new traffic (customers) at a given node. Where there is no creation of traffic frames, γ_i is defined as 1. The transition probabilities for the entire network can be represented as a transition matrix denoted as $H = [h_{ij}]$. The formulation of H is presented in detail in Appendix A.

$$\lambda_j = \lambda_{0j} + \sum_{i=1}^{nr} \lambda_i \gamma_i h_{ij} \tag{4.11}$$

The proportion of arrivals to node j coming from node i is denoted as k_{ij} and computed using

4.12 where the numerator term is simply the traffic arrival rate from node i to node j which is denoted as λ_{ij}.

$$k_{ij} = \frac{\lambda_i \gamma_i h_{ij}}{\lambda_j} \tag{4.12}$$

Based on the total traffic arrival at each node, the utilisation of each network node denoted as ρ_i is defined in 4.13 where μ_i is the service rate at the i^{th} node and m_i is the number of servers associated with the i^{th} node.

$$\rho_i = \frac{\lambda_i}{m_i \mu_i} \tag{4.13}$$

4.3.2.2 Traffic Variability Parameters

Considering the coefficient of variation of an arrival process as c_a, the QNA method estimates the internal traffic flow variability parameters as the squared coefficient of variation of the arrival processes at node j denoted as c_{aj}^2. This is derived using the system of linear equations given in 4.14, where a_j and b_{ij} are constant parameters that are derived based on the traffic flow process operations namely, the splitting process, the superposition process and the departure process. The squared coefficient of variation of the external arrival process to the node j is denoted as C_{0j}^2.

$$c_{aj}^2 = a_j + \sum_{i=1}^{nr} c_{ai}^2 b_{ij} \tag{4.14}$$

The constants a_j and b_{ij} can be estimated using 4.15 and 4.16 respectively as presented in [97], where x_i and w_j are the weighting functions that are introduced to describe the linear (convex) combination of the approximations of the departure process operation and the superposition process operation, respectively.

$$a_j = 1 + w_j \left\{ \left(k_{0j} c_{0j}^2 - 1\right) + \sum_{i=1}^{nr} k_{ij} \left[(1 - h_{ij}) + \gamma_i h_{ij} \rho_i^2 x_i \right] \right\} \tag{4.15}$$

$$b_{ij} = w_j k_{ij} h_{ij} \gamma_i \left(1 - \rho_i^2\right) \tag{4.16}$$

The variables x_i and w_j are estimated as

$$x_i = 1 + m_i^{-0.5} \left(\max \left\{ c_{si}^2, 0.2 \right\} - 1 \right), \tag{4.17}$$

$$w_j = \left[1 + 4 \left(1 - \rho_j\right)^2 \left(v_j - 1\right) \right]^{-1}, \tag{4.18}$$

where c_{si}^2 is the squared coefficient of variation of the service times at the i^{th} node and v_j is defined as,

$$v_j = \left(\sum_{i=0}^{nr} k_{ij}^2 \right)^{-1}. \tag{4.19}$$

4.3.3 Performance Metrics

4.3.3.1 Node Response Time

The response time at each node T_j^{rp} is defined as the sum of the mean waiting time in a queue W_j and the service time of the node $1/\mu_j$ as depicted in 4.17. The service times of each node are based on the respective equations presented in section 4.3.1. However, the waiting time at a node differs from node to node, hence is computed using the output parameters of the QNA model.

$$T_j^{rp} = W_j + 1/\mu_j \tag{4.20}$$

The waiting time of the classifier is modelled as a G/G/1 queue as in Figure 4.7 since it only has one queue and one server. The standard version of the QNA method proposes the use of the Kraemer and Langenbach-Belz formula presented in 4.21 for estimating the queuing delay based on a G/G/1 queue. It introduces a refinement/correction parameter g to improve the accuracy of the well-known Allen-Cunneen formula for G/G/1 queues.

$$W^{G/G/1} = \frac{\rho\left(c_a^2 + c_s^2\right) g}{2\mu\left(1-\rho\right)} \tag{4.21}$$

The parameter g is a function of ρ, c_a^2 and c_s^2 which is derived as

$$g\left(\rho, c_a^2,\ c_s^2\right) = \begin{cases} exp\left(-\frac{2(1-\rho)}{3\rho}\frac{(1-c_a^2)^2}{c_a^2+c_s^2}\right), & c_a^2 < 1 \\ 1, & c_a^2 \geq 1 \end{cases}. \tag{4.22}$$

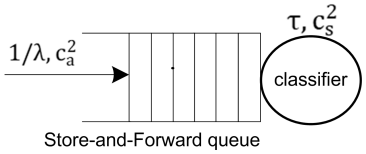

Figure 4.7: The G/G/1 single queue, single server model

The waiting time at the queues in the scheduler component of the SDN switch is also modelled as a G/G/1 queue but with multiple separate queues of different priorities and a single server with a multiplexer as depicted in Figure 4.8.

Irrespective of the port that is used to transmit a frame from any of the queues during a quantum time, the weighted port transmission rate will be the same since all ports have the same bandwidth. In addition, with the WRR scheme, the queuing delay of one queue depends on the scheduling of a specific number of frames in a WRR cyclic manner. Therefore it is

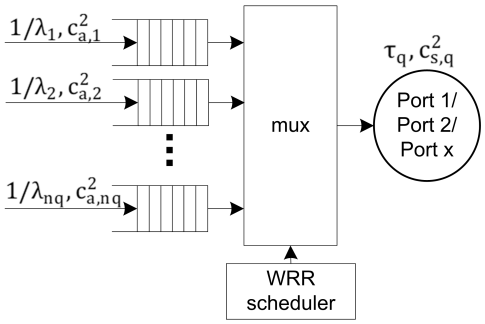

Figure 4.8: G/G/1 multiple queues, single server model with weighted round-robin scheduling

appropriate to represent all the ports as a single server. The prioritised WRR scheduler cycles through all the different queues and forwards the frame from each queue onto the appropriate port before repeating the cycle. In each cycle, the queue with the highest priority is allocated the highest portion of the selected port bandwidth for transmission (thus, the highest number of scheduled frames) which is fixed whereas the lowest priority queue is given the least portion of the port bandwidth for transmission (thus, least number of scheduled frames) which is also fixed. The WRR is work-conserving and starvation free in that, even if there are no frames in one queue the transmission is not idle and then no queue is denied transmission due to concurrent transmissions in other queues. The priority of the queue corresponds to the q^{th} traffic type that is buffered in that queue where the highest priority is attributed to $q = 1$. The waiting time of the q^{th} priority queue W_q based on a single arbiter can be estimated using 4.23. The first summand in 4.23 depicts the delay experienced by a frame in the q^{th} queue to which it arrives at. The parameters L_q and τ_q are the queue length and the weighted service time for the q^{th} queue, respectively. The second summand of 4.23, $\tau_{wr,\bar{q}}$ depicts the additional delay introduced due to the scheduling of other queues $\bar{q}$ (where $\bar{q} \neq q$). This defines the type of scheduling scheme used to schedule the frames from the different queues which are presented subsequently. The parameter ξ is the residual service time or the remaining service time when the server is busy at the time when the frame arrives at the back of the queue.

$$W_q = \tau_q L_q + \tau_{wr,\bar{q}} + \xi \tag{4.23}$$

The length of the queue L_q can be estimated using the Kraemer and Langenbach-Belz approximation for the number of customers in a queue as derived in 4.24 where g is obtained using 4.22.

$$L_q = g\frac{\rho_q^2\left(c_{a,q}^2 + c_{s,q}^2\right)}{2\left(1-\rho_q\right)} \tag{4.24}$$

The delay of the interfering queues $\tau_{wr,\bar{q}}$ can be estimated in 4.25. The WRR scheduling policy is dictated by the scheduling parameter $sch\left(L_q, L_{\bar{q}}\right)$.

$$\tau_{wr,\bar{q}} = \sum_{\bar{q}=1,\bar{q}\neq q}^{n_q} \left(\tau_{\bar{q}} \times wr\left(L_q, L_{\bar{q}}\right)\right) \tag{4.25}$$

In the WRR scheme, the cyclic behaviour of the scheduler implies that the length of each queue depends on the length of the interfering queues in the multi-queue system. Therefore, to capture the cyclic effect, the queuing delay of the q^{th} queue will be bounded by the maximum length of L_q and the minimum length of $L_{\bar{q}}$ if less than L_q. Thus, the scheduler does not schedule an infinite length of the queues which hence captures cyclic behaviour. Therefore, $sch\left(L_q, L_{\bar{q}}\right)$ for the cyclic behaviour can be defined using 4.26 as proposed by Foroutan et al. in [112].

$$wr\left(L_q, L_{\bar{q}}\right) = \min(L_q, L_{\bar{q}}) \tag{4.26}$$

The remaining service time ξ which considers the probability of a busy transmission medium ρ can be estimated using 4.27.

$$\xi \approx \sum_{q=1}^{n_q} \frac{\rho_q\left(c_{a,q}^2 + c_{s,q}^2\right)}{2\mu_q} \tag{4.27}$$

At the eDC, for each vMAC cluster, there is a G/G/m queue as depicted in Figure 4.9 whereby each vMAC instance is represented by a single server in the multi-server model. Thus m, in this case, is equal to the number of vMAC instances $n_{vm,q}$ for each queue.

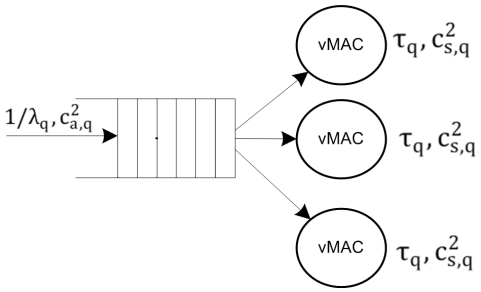

Figure 4.9: G/G/m single queue, multi-server model

The QNA method proposes 4.28 for estimating the delay for the G/G/m queue. The parameter $W^{M/M/m}$ represents the mean queuing delay for an M/M/m queue.

$$W^{G/G/m} = 0.5\left(c_a^2 + c_s^2\right) W^{M/M/m} \tag{4.28}$$

The parameter $W^{M/M/m}$ is derived in 4.29 and 4.30. The parameter $C\left(m,\rho\right)$ is Erlangs C formula. The parameter k in 4.30 represents an integer from 0 to $m-1$.

$$W^{M/M/m} = \frac{C\left(m,\rho\right)}{m\mu - \lambda} \tag{4.29}$$

$$C\left(m,\rho\right) = \frac{\left(\frac{(m.\rho)^m}{m!}\right)\cdot\left(\frac{1}{1-\rho}\right)}{\sum_{k=0}^{m-1}\frac{(m.\rho)^k}{k!} + \left(\frac{(m.\rho)^m}{m!}\right)\cdot\left(\frac{1}{1-\rho}\right)} = \frac{1}{1+(1-\rho)\left(\frac{m!}{(m.\rho)^m}\right)\sum_{k=0}^{m-1}\frac{(m.\rho)^k}{k!}} \tag{4.30}$$

4.3.3.2 Virtual Network Response Time (Sojourn Time)

The virtual network response time T^{vn} is defined as the average time spent in the virtual network from the first arrival to the final departure from the network. There are n_q traffic

types in the virtual network where each traffic type follows particular routes after the classifier component of the SDN switch. Therefore, if each traffic type visits nr_q number of nodes, then the virtual network response time for the specific traffic type denoted as T_q^{vn} can be estimated using 4.31 where $W_{q,j}$ and $\mu_{q,j}$ represent the queuing delays and service rates of the j^{th} node visited by the q^{th} traffic type respectively. Thus, q, j defines the sequence of specific nodes visited by the q^{th} traffic type throughout its stay in the virtual network. It is important to highlight that, in 4.31 any of the traffic types can be routed to any of the eDC nodes depending on the mapping or routing strategy. Therefore, considering all the eDC nodes for a given traffic type, the response time of the q^{th} traffic is computed based on the average of the response times at each eDC node for that particular traffic type.ro

$$T_q^{vn} = \sum_{j=1}^{nr_q} (W_{q,j} + 1/\mu_{q,j}) \tag{4.31}$$

4.3.3.3 Virtual Network Traffic Load and Virtualisation Delay

The virtual network traffic load is the rate at which traffic arrives at the input of the virtual network which in this case is the input of the store-and-forward buffer. In other words, the external traffic arrival rate of the store-and-forward buffer as defined in the QNA model is the traffic load of the virtual network. The traffic load is important because it characterises the traffic in the virtual network which in turn defines the response times and total sojourn. The virtual network traffic load is denoted as λ_{0v} and is computed using 4.32. This represents the superposition of the rates generated by n_p number of independent pGW nodes, where λ_p is derived using 4.8.

$$\lambda_{0v} = \sum_{p}^{n_p} \lambda_p \tag{4.32}$$

After obtaining the total sojourn time in the virtual network based on the traffic load, the overall time used to deploy and augment the GW MAC can be derived. This is called the

virtualisation time, denoted as T_q^{vr} and is computed using 4.33. This takes into account the CPRIoE encapsulation and decapsulation delay at the pGW node and eDC node respectively which affects all the frames. It is assumed that the encapsulation at the pGW and the encapsulation at the eDC are the same. Based on the model assumptions, this is equally the same time it takes for an ACK frame to be sent back to the pGW node.

$$T_q^{vr} = 2T_{ecp} + T_q^{vn} \tag{4.33}$$

4.4 Media Access Throughput with GW MAC Virtualisation

The throughput model presented in Chapter 3 considers the execution of the GW MAC functions locally within the GW node itself as one dedicate GW MAC instance. In this section, the media access throughput is modelled based on the virtualisation of the GW MAC using the proposed virtual network framework. The transmission procedure of a successful transmission cycle for a vAG for a given RAW slot based on the virtualised GW MAC is presented in Figure 4.10. Since the GW MAC is externally deployed in a remote eDC through the virtual network, the GW does not execute any MAC functions. However, the delay imposed by the virtualisation process in transmitting a frame (PS-Poll or Data frames) to the eDC as well as responding with an ACK frame are each accounted for by the virtualisation delay T_{vr}. The rest of the delay activities are the delays encountered between the device and the GW only. For an unsuccessful transmission, a transmission may fail either due to a collision or due to the delay within the virtual network, since it is considered that the virtual network has infinite buffer capacity. The duration of an unsuccessful transmission remains the same as in 3.12 since the IEEE 8011ah protocol can not distinguish between a collision and any other form of failed transmission.

Based on Figure 4.10, the virtualisation delay can be incorporated as part of the transmission procedure by defining T_{su}^{vm} as the duration of a successful transmission within a time slot for a virtualised GW MAC computed using 4.34. Thus, the virtualisation delay for a PS-Poll

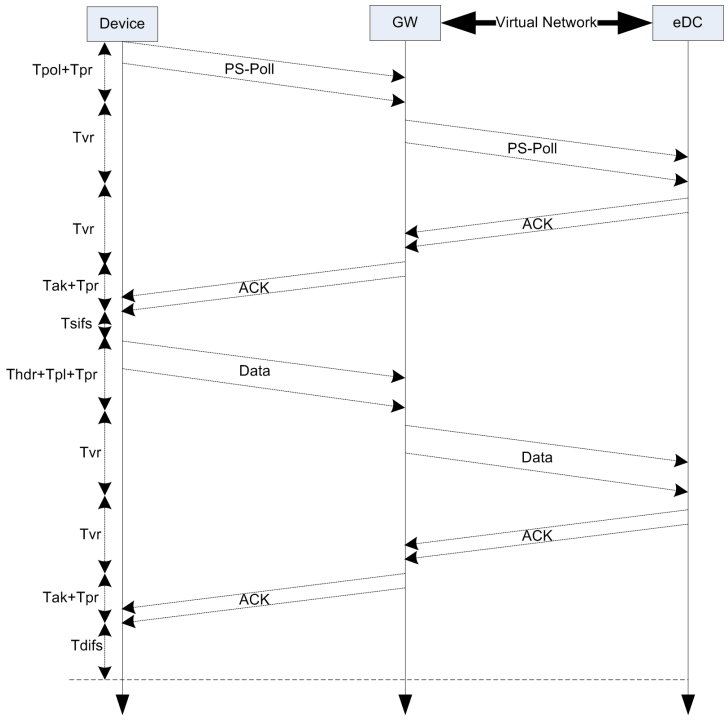

Figure 4.10: The Virtualised GW MAC transmission procedure of a successful frame in a RAW timeslot for S1G IoT devices using PS-Poll.

frame, PS-Poll ACK frame, Data frame and a Data ACK frame (each assumed to incur the same mean virtualisation delay) are added to the delay of the transmission procedure without virtualisation.

$$T_{su}^{vm} = T_{pol} + T_{hdr} + T_{pl} + 2T_{ak} + 3T_{sifs} + 4T_{pr} + T_{difs} + 4T^{vr} \tag{4.34}$$

The virtualisation delay can be high such that the timing requirement for an ACK frame response from the eDC is not met. Therefore, the probability of a failed transmission due to the virtualisation delay denoted as P_{bk}^{vak} can be obtained using 4.35 where the virtualisation delay T^{vr} is doubled to represent the mean virtualisation delay in both directions. It is important to highlight that 4.35 captures the case when a single node experiences over-utilisation, the $P_{bk}^{vak} = 1$. This is because when the utilisation of a single node is 1, the delay of the entire route is infinite which technically indicates the frame transmission will fail.

$$P_{bk}^{vak} = \begin{cases} \frac{2T^{vr}}{T_{ak_{to}} - (T_{sifs} + 2T_{pr})}, & 2T^{vr} \leq (T_{ak_{to}} - (T_{sifs} + 2T_{pr})) \\ 1, & otherwise \end{cases} \tag{4.35}$$

The probability of a successful transmission based on the virtualised GW MAC is denoted as P_{su}^{vm} and is computed using 4.36. This depends on the transmission probabilities due to the ACK time-out and the probabilities of the contention access mechanism.

$$P_{su}^{vm} = P_{su}(1 - P_{bk}^{vak}) \tag{4.36}$$

The throughput considering the GW MAC virtualisation denoted as S_{vm} is derived in 4.37. The network throughput for all the vAGs considered based on the GW MAC virtualisation can simply be computed using 3.14, based on the value of S_{vm}.

$$S_{vm} = \frac{P_{su}^{vm} P_{at} L_{pl}}{(1 - P_{at}) t_{slot} + P_{su}^{vm} P_{at} T_{su}^{vm} + P_{at}(1 - P_{su}^{vm}) T_{cl}} \tag{4.37}$$

4.5 Experiment, Results and Analysis

In this section, the computation of the model studied is presented based on a numerical experiment. The results obtained from the computation of the QNA model are compared with simulation results and analysed to validate and verify the behaviour of the model. The computation of the model was performed using MATLAB R2022a and the simulation was performed using the JSIMgraph simulator of the Java Modelling Tool (JMT) software package version 1.2.1. The implementation of the simulation model is discussed in Appendix B.

4.5.1 Setup and Parameters

A total of 21 queueing nodes were used for computing and simulating the virtual network model, where each node is identified by a node ID from 1 to 21 as shown in table 4.1. The

characteristics of the traffic types used are the same as in table 3.1. The setup considers 4 pGWs at different locations connected to the virtual network. Each vAG of a given traffic type is mapped to exactly 1 vMAC instance in the eDC. The traffic is equally distributed amongst the different eDCs with a one-to-one mapping of the number of vAGs to the number of vMAC instances. Thus, each queueing node in each of the 4 eDCs will have the same number of vMAC instances for the same number of vAG mapped to it. Table 4.2 shows an example of the setup using 1 vAG per traffic type or 4 vAGs per pGW. For example, using 4 vAGs per pGW means that there will be 4 vAGs / 4 traffic types = 1 vAG for each traffic type for all 4 pGWs. This means there will be 4 vAGs x 4 pGW = 16 vAGs virtualised in the whole network with 4 vAGs per traffic type. Each eDC will then host 16 vAGs / (4 eDC x 4 clusters) = 1 vAG per traffic type. Similarly, if there are 8 vAGs per pGW for all 4 pGWs, there will be 2 vAGs per traffic type and each eDC will have exactly 2 vMAC instances per traffic type. In that case, all the eDC queues will have equally the same amount of GW vMAC traffic arrivals to be processed, which makes it convenient to analyse the results.

Table 4.1: Virtual network setup for the queuing nodes.

Node ID	**Node Description**
1	Classifier queue
2,3,4 & 5	Scheduler queue for Traffic A,B,C & D
6,7,8 & 9	eDC 1 (Queue for Traffic A,B,C & D)
10,11,12 & 13	eDC 2 (Queue for Traffic A,B,C & D)
14,15,16 & 17	eDC 3 (Queue for Traffic A,B,C & D)
18,19,20 & 21	eDC 4 (Queue for Traffic A,B,C & D)

Table 4.2: Virtual network setup example for vAGs and vMAC instances.

pGW	**vAGs per Traffic**				**eDC**	**vMACs per Traffic**			
	A	B	C	D		A	B	C	D
1 (4 vAGs)	1	1	1	1	1	1	1	1	1
2 (4 vAGs)	1	1	1	1	2	1	1	1	1
3 (4 vAGs)	1	1	1	1	3	1	1	1	1
4 (4 vAGs)	1	1	1	1	4	1	1	1	1

All the pGW nodes have the same CPRIoE parameters provided in table 4.3. The standard

value of 260.416μs for the duration of a CPRI basic frame is used. The CPRI line rate for option 1 defined as 614.4 Mb/s is also used. The Ethernet overhead length is chosen based on the standard CPRIoE frame specification of 44 bytes overhead. A CPRI protocol switching delay of 10μs is used. Each pGW is equipped with an IEEE802.3 10GBASE-ER Ethernet link interface recommended for the CPRIoE [106], which implies a single lane with a line rate of 10Gb/s over single-mode fibre up to a maximum distance of 30 km. The classifier of the SDN switch is considered to process 10×10^6 frames per second. The scheduler component is linked to the SDN switch port also equipped with an IEEE 802.3 100GBASE-ER4 Ethernet interface over single-mode fibre up to a distance of 40 km, implying 4 lanes for the 4 eDC nodes and a line rate of 25 Gb/s per lane. The round-robin service weight for the different traffic types is chosen such that traffic A has the highest weight, followed by traffic B, C and D with the lowest weight. This provides fairness given that Traffic A has the highest traffic rate and mean frame length followed by Traffic B, C and D. To create a heterogeneous set of eDCs where the resource availability in the eDCs is not the same, eDC 1 has the highest virtual processing resource followed by eDC 2, eDC 3 and eDC 4. The allocation factor per cluster in each eDC is such that Traffic A is apportioned the most resources, followed by traffic B, C and D. The squared coefficient of variation for the external traffic arrivals and the service times are considered to be 1 implying a Poison process approximating the aggregated traffic from multiple pGW nodes. The parameters of the MAC throughput model with virtualisation are the same as the parameter used in table 3.2.

4.5.2 Analysis of Results

4.5.2.1 Node Response Time

Figure 4.11 shows the response time of all the nodes based on 4, 16 and 32 vAGs in each pGW with an equal number of vAGs per traffic type. Here, it is considered that all vAGs always have at least 1 frame to be transmitted (not based on the devices connected/contenting for channel access). For the eDC nodes (nodes 6 to 21), it can be observed that when the number of vAGs is 4 and 16, The queues in eDC 4 have the highest response time followed by eDC 3, eDC 2, and

Table 4.3: Virtualisation Network Experimental Setup Parameters

Parameter	**Value**
CPRI Basic Frame duration (T_{bf})	260.416 μs
CPRI line rate (R_{cp})	614.4 Mb/s (option 1)
Ethernet Overhead length (L_{eo})	44 Bytes
CPRI protocol switching delay (T_{cs})	10 μs
Ethernet line rate (R_{et})	10 Gb/s
Classifier Mean Service Rate (μ_{cls})	10×10^6 frames/s
Switch port line rate (R_{sw})	25 Gb/s
WRR service weight ($w_{rr,q}$)	4, 3, 2, 1 (Traffic A, B, C, D)
Processing Resource (C_e)	1000, 300, 10, 8 $MIPS$ (eDC 1, 2, 3, 4)
Processing Resource weight ($w_{vm,q}$)	3.5, 2.5, 1.5, 1.0 (Traffic A, B, C, D)
Virtual Machine Scheduling Delay (t_{vm})	5 μs
Variability of external arrivals (C_{0j}^2)	1
Variability of service time (C_{sj}^2)	1

eDC 1. When the number of vAGs is 32, the queues in eDC 4 and eDC 3 experience an infinite response time (indicated using 500 μs) for all traffic types. This is a result of the difference in virtual processing resource capacity where eDC 1 has the highest number of resources allocated to each cluster followed by eDC 2, eDC 3 and eDC 4. For the scheduler (nodes 2 to 6), node 5 has the highest response time followed by nodes 4, 3 and 2. This is because node 2 is allocated the highest WRR weight followed by nodes 3, 4 and 5 for priority scheduling. The classifier has a very low response times time compared to all the other nodes.

Figure 4.12 shows the utilisation of each node obtained using the same setup described. The utilisation increases significantly when there are more vAGs. When the number of vAGs is 32, the utilisation is greater than 100% at the nodes in eDC 4 and eDC 3. This is hence reflective of the fact that the delay increases without bounds for these nodes as shown in Figure 4.11.

To further validate the simulated results from the JSIMgraph tool, the confidence interval (CI) of the response time at each node based on a 99% confidence level was computed and is presented in Figure 4.12 for the different number of vAGs per traffic type. A total of 100000 samples were used in the simulator for each measurement. It can be observed that the CI for

each node is relatively narrow for all the different number of vAGs per pGW.

4.5.2.2 Virtualisation Delay

In Figure 4.14 the virtualisation delay for each traffic type is plotted against the number of vAGs for each pGW and all traffic types with an equal number of vAGs. Here, it is also considered that all vAGs always have at least one frame to be transmitted. It can be observed for all traffic types that the virtualisation delay increases as the number of vAGs increases. This is because when there is more vAGs the number of frames transmitted per vAG is high hence the load on the virtual network is high. For traffics C and D, the increase in delay is higher, approaching infinity with 16 vAGs per pGW. For traffic A and B, the delay is relatively lower, approaching infinity with 20 vAGs. This is due to the allocation of resources based on the different traffic types at the scheduler (WRR weight) and the eDC (cluster allocation weight).

Figure 4.15 shows the confidence interval of the simulated virtualisation delay for each traffic type based on a 99% confidence level. A total of 100000 samples were used in the simulator for each measurement. It can be observed that the confidence interval in all cases is relatively narrow which implies a high precision in the simulated results.

4.5.2.3 Failure Probability with GW MAC Constraints and Virtualisation

In Figure 4.16, the transmission failure probability with GW MAC virtualisation and GW MAC constraints for 4 and 8 vAGs and 4 RAWs is plotted against the total number of devices connected to the GW for the different LS-IoT traffic types in each subplot. The GW MAC processing rate of the constrained case is equal to the average of the processing capacities for all 4 eDCs for comparison.

It is evident from the results for traffics A, B and C that when the GW MAC is virtualised for 4 vAGs, the transmission failure probability versus the number of devices increases when compared to using a dedicated GW MAC with constraints also with 4 vAGs. The same be-

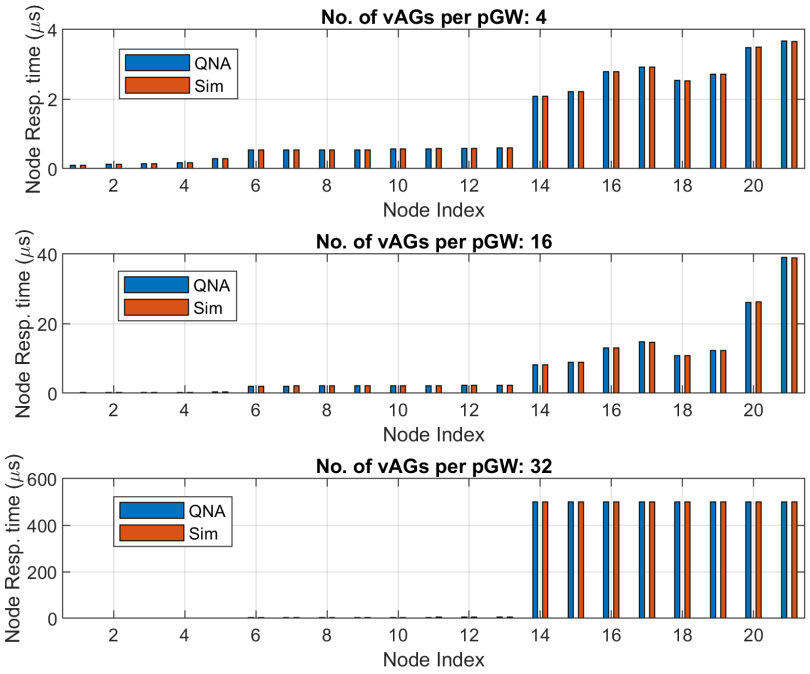

Figure 4.11: Node Response time for each node in the virtual network based on the different number of vAGs per pGW.

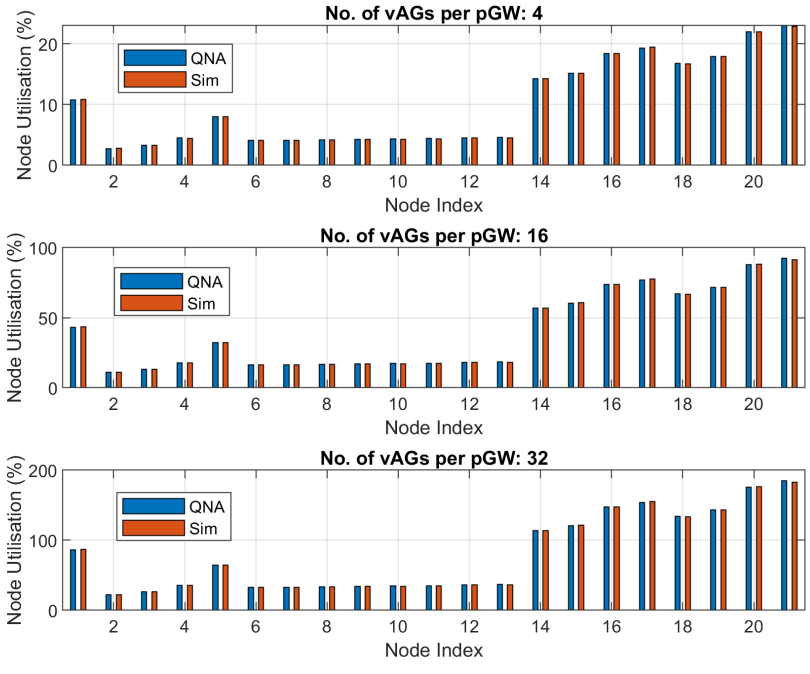

Figure 4.12: Node Utilisation for each node in the virtual network based on the different number of vAGs per pGW.

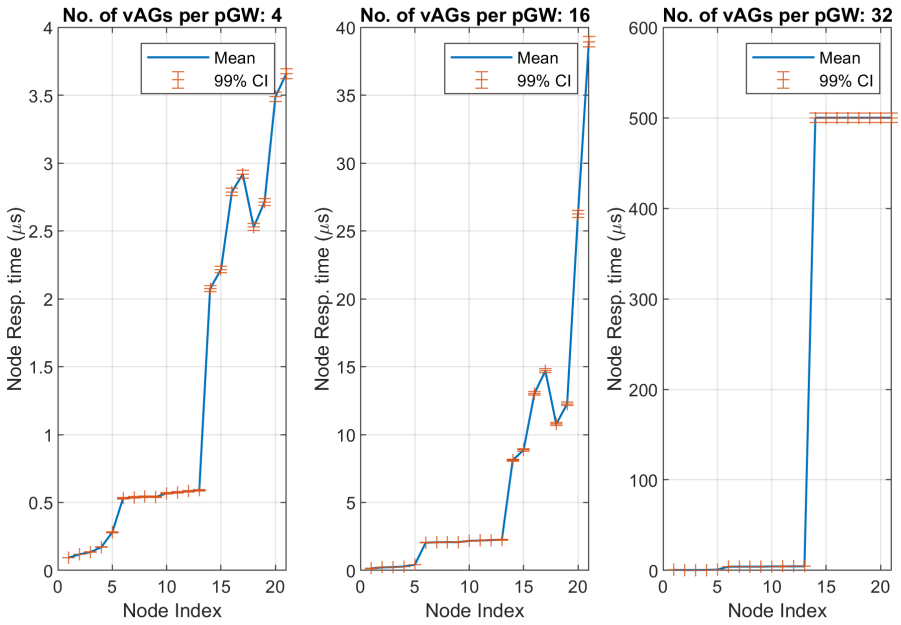

Figure 4.13: Confidence Interval plots for each simulated (JSIMgraph) node response time based on the different number of vAGs per traffic type. Confidence level: 99%

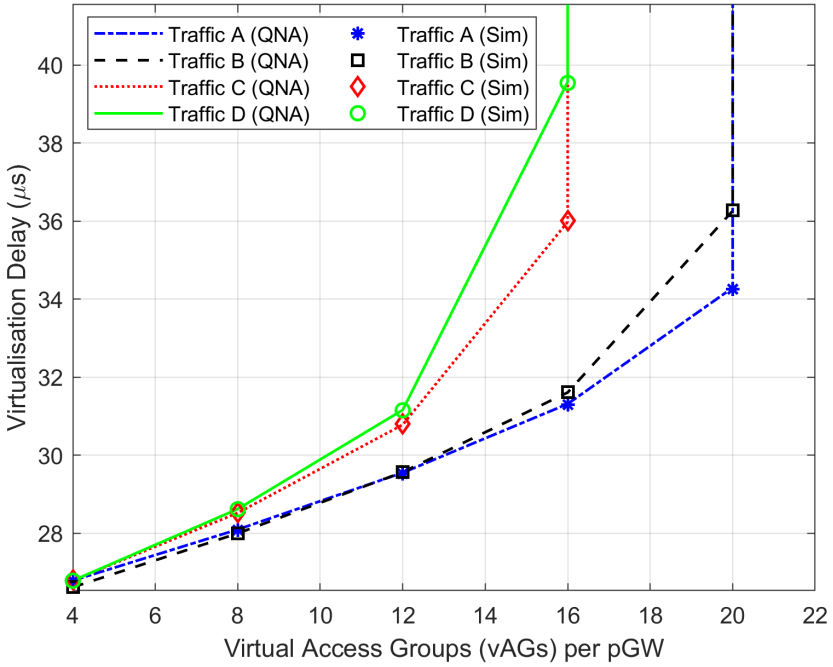

Figure 4.14: Virtualisation delay versus the number of vAGs per pGW node.

haviour can be observed when comparing the virtualised case with the constrained case using 8 vAGs for traffic A, B and C. This is because the constrained case is affected by the single increased blocking probability as a result of the GW MAC buffer and the GW MAC processing constraints. On the other hand, the virtualised case distributes the processes of the GW MAC at different locations over the proposed virtual network with a slower response time and no buffer constraints as opposed to the constrained case. For traffic D, the transmission failure probability is 0 in all cases. This is because of the low traffic rate attributed to traffic D which reduces the aggregated traffic arrivals such that the effects of collisions, the GW MAC constraints and the virtualisation is not pronounced for the given number of devices.

4.5.2.4 Throughput with GW MAC Constraints and Virtualisation

In Figure 4.17, the media access throughput results with GW MAC virtualisation and GW MAC constraints for 4 and 8 vAGs and 4 RAW is plotted against the total number of devices connected to the GW for the different LS-IoT traffic types in each subplot. The GW MAC processing rate of the constrained case is equal to the average of the processing capacities for all 4 eDCs for comparison.

It can be observed for traffics A, B and C that when the GW MAC is virtualised for 4 vAGs, the throughput performance versus the number of devices is better when compared to using a dedicated GW MAC with constraints also with 4 vAGs. The same behaviour can be observed when comparing the virtualised case with the constrained case based on 8 vAGs for traffic A, B and C. The behaviour is because the failure probability is high due to the GW MAC constraints as well the increase in GW MAC response time. With the virtualised case the GW MAC constraints are minimised whereby the response time is reduced due to the distributed processing of the GW MAC frames as well as the weighted routing of the GW MAC frames of the proposed virtual network. For traffic D, the throughput increases linearly but does not reach the peak throughput for all cases. However, when there are 8 vAGs the rate at which the throughput increases linearly is lower when compared to when there are 4 vAGs. This is because of the significant difference in the number of devices per vAG where there is a low

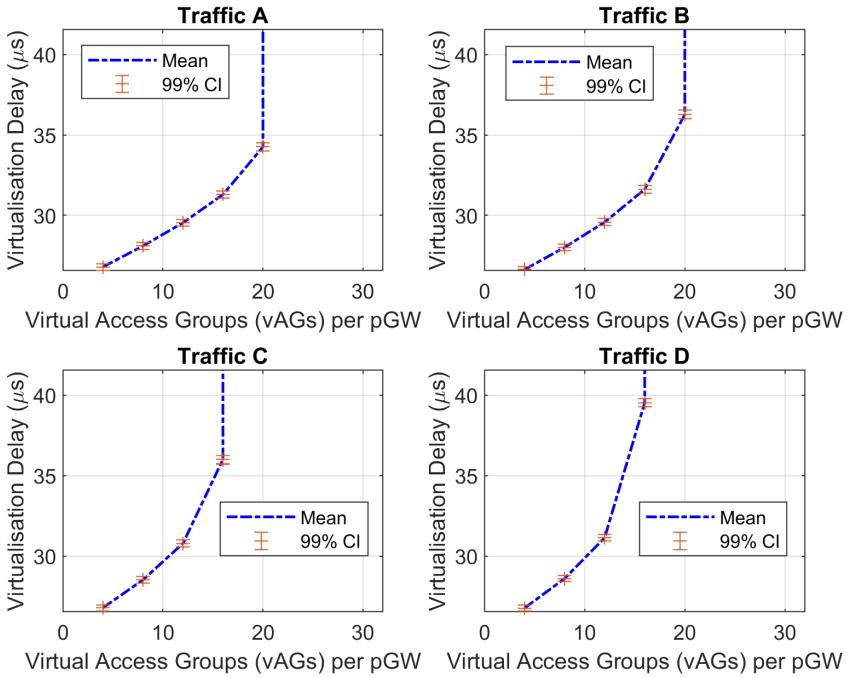

Figure 4.15: Confidence Interval for the simulated (JSIMgraph) virtualisation delay results versus the number of vAGs per pGW node. Confidence level: 99%

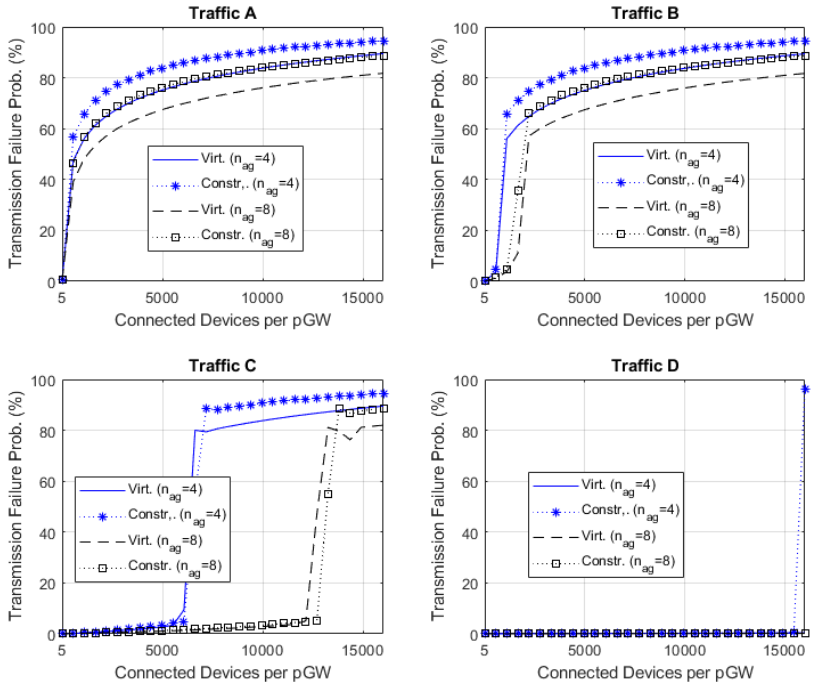

Figure 4.16: Media access transmission failure probability based on the GW MAC virtualisation, and the GW MAC constraints, plotted against the number of devices connected to each pGW.

aggregated traffic for 8 vAGs than 4 vAGs.

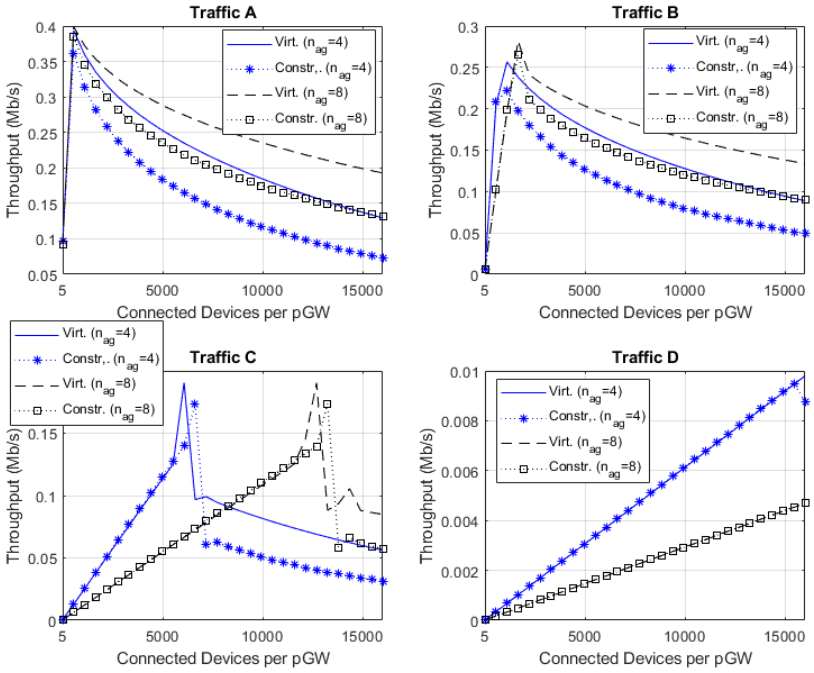

Figure 4.17: Media access throughput performance based on the GW MAC virtualisation and GW MAC constraints, plotted against the number of devices per pGW.

4.6 Conclusion

In this chapter, a virtual network framework was proposed and modelled to support the scalability of large-scale IoT networks. The virtualisation delay was modelled and the impact on the media access throughput and transmission failure probability was derived. The results obtained in this chapter show that the virtual network analytical model conforms with the simulation model, hence validating the QNA approach employed. Overall, the results suggest that the virtualisation of the GW MAC over the proposed virtual network architecture for a LS-IoT can support the scalability of the media access for multiple vAGs. The virtualisation approach can support more vAGs than the conventional approach of having a single dedicated GW MAC localised within the GW itself. Though the proposed virtualisation approach improves the throughput scalability better than the constrained GW MAC, the results presented suggest that if the vAGs are increased, the virtualisation delay increases significantly which

can counter the throughput gains of the virtualisation process. It is imperative to introduce some optimality in minimising virtualisation delays especially when the vAGs are provisioned randomly. Also, there needs to be an efficient allocation of GW vMAC instances based on the number of vAGs to avoid over-provisioning of resources.

CHAPTER 5. A DISTRIBUTED VIRTUAL GATEWAY MAC PLACEMENT APPROACH FOR LARGE-SCALE IOT

5.1 Introduction

To improve the scalability of the GW MAC through the creation of virtual access groups and virtualisation of the GW MAC functions for each access group, the pGW and the virtual network need to be robust such that the MAC throughput is not degraded as the number of devices gradually increases. As the number of devices connected to the pGW node increases, there is a need for more vAGs to support the connected devices. Thus, for each vAG, there is a maximum number of devices beyond which the throughput is degraded significantly. Therefore, provisioning more vAGs may support more devices, although the number of vAGs is limited by the RF virtualisation or RF virtual resource units that can be provisioned at the PHY layer. The creation of more vAGs implies the need for more vMAC instances to be deployed in the eDC. This increases the traffic rate on the virtual network and the number of vMAC frames to be processed in the eDC concurrently. Although each component of the virtual network experiences an increase in delay as a result of the increase in devices and vAG groups, the distributed nature of the eDC nodes and the resource provisioning constraints are very critical in ensuring that the virtualisation delay is minimised and the number of vAGs accommodated is maximised. The deployment of vMAC instances at the edge network needs to ensure that as more devices join the pGW node, the vAGs are provisioned dynamically and the vMAC instances associated with the vAGs are deployed to ensure a minimal response time. It is imperative to propose a distributed vMAC deployment strategy to effectively and efficiently distribute the vMAC instances across the set of eDC hosts such that the effects of latency caused by the virtualisation are reduced. To achieve this, the following objectives/contributions are made:

a. To effectively manage the provisioning of vMAC Instances or placement to support the dynamic creation of vAGs, by proposing an SDN-NFV based application capable of providing real-time discovery of resources, and efficient routing for the execution of the vMAC instances.

b. To adequately model the vMAC placement problem based on the response times of the eDC nodes, the placement decision factors and constraints by using the binary integer linear programming method.

c. To minimise the response time of the vMAC at the eDC as more vAGs are created by developing and employing a heuristic algorithm that searches and selects the best eDC node to place a vMAC request while meeting the constraints of the eDC placement process.

5.2 Proposed NFV-SDN Virtual MAC Placement Application

To achieve the objective of effectively managing the vMAC placement process, an NFV-SDN based application with real-time capabilities is proposed. The proposed application interfaces the NFV Management and Orchestration (NFV-MANO) function with the SDN Controller to effectively and efficiently route packets to the eDC. The NFV-MANO is a framework proposed by the European Telecommunications Standards Institute (ETSI) to provide effective management and orchestration of VNFs within a Network Function Virtualisation Infrastructure (NFVI). The NFV-MANO is made up of three main components as shown in 5.1. These are the Orchestrator, the NFV management and the Virtualised Infrastructure Manager (VIM) component. The orchestrator manages the lifecycle of the network services whereas the lifecycle of the VNF instances is managed by the VNF manager. The orchestrator and VNF manager operate with the support of the VIM component which is responsible for provisioning or allocating the virtual resources for the network functions being virtualised.

In this study, it is considered that the NFV-MANO operates based on consolidated hardware resources from all the eDC nodes where each VNF instance virtual resource is mapped to

one eDC node's physical resources through the virtualisation layer also called the hypervisor. Thus, a VNF instance which runs the vMAC functions is virtualised based on the portion of computing, memory and network resources belonging to one eDC node. This is to eliminate delays imposed due to the distributed physical resources shared for a single vMAC function. For example, using the network resources of one eDC and using the compute resources of another eDC for one task causes more delays due to the transmission of frames from one eDC to another. Hence, this approach eliminates that delay. This also makes the NFV-MANO framework conform with the proposed virtual network architecture in Figure 4.2. The VNFs are also clustered according to traffic types in the same manner as presented in Chapter 4. While the VNF instances are being managed for executing the vMAC functions in the eDC, the proposed SDN-NFV vMAC application manages the routing of vMAC frames and the placement of the vMAC instances through the VIM which creates the virtual resources based on the physical resources of the best eDC node selected by the SDN-NFV application. The proposed SDN-NFV vMAC application is divided into two components as in Figure 5.1. The first component is the Network Activity Monitor (NAM), and the second component is the Placement Decision Maker (PDM). The combination of these components helps manage the virtualisation requests and define where the vMAC frames must be routed to by the SDN switch scheduler through the SDN controller. The functions of the NAM and the PDM are described below.

5.2.1 Network Activity Monitor

To provide a dynamic distributed placement process for the vMAC requests in real-time, the mechanisms for request monitoring, resource monitoring and resource discovery are very critical. Therefore, the NAM monitors the pGW node to establish the desired number of vAGs that maintains a certain level of Quality of Service in terms of transmission collisions. The creation of vAG based on the QoS level is beyond the scope of this study. In addition, the NAM monitors the traffic and utilisation of the eDC nodes based on the vMAC task loads. The NAM also obtains information about the eDC resources through the VIM which manages the records and discovery of resources. The NAM component is described in Psuedo code 5.1. The

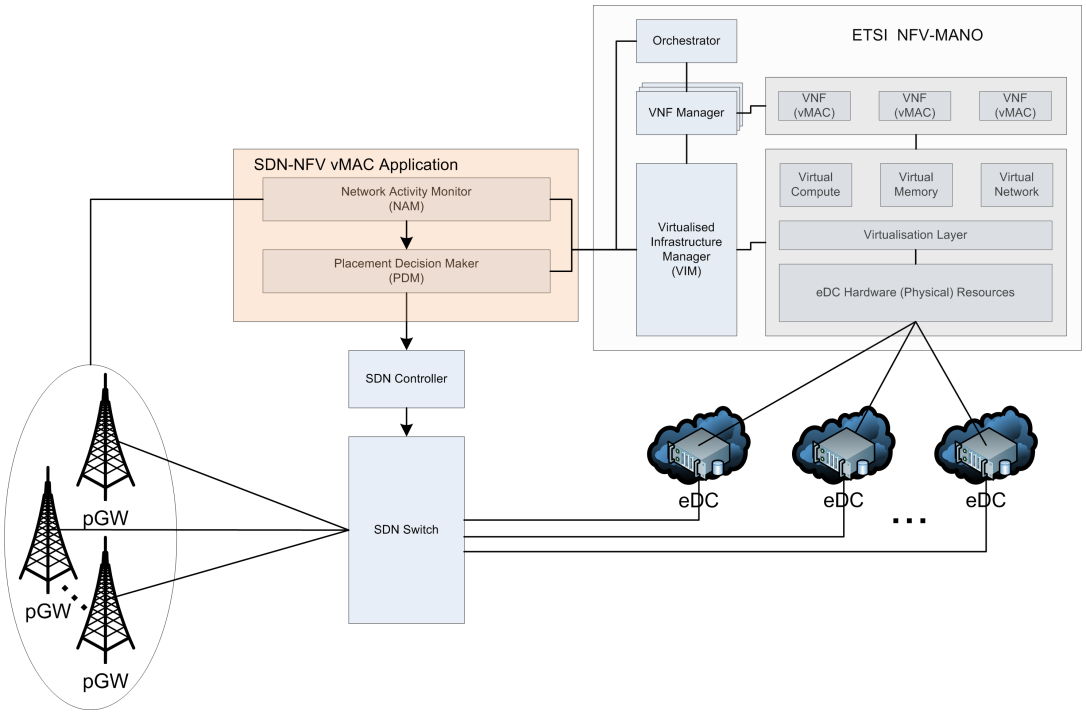

Figure 5.1: SDN-NFV Application for vMAC placement Management

NAM component always updates the IDs of the active eDC nodes, their resource levels, their utilisation and traffic load. These are obtained from NFV-MANO's VIM component. It then keeps checking for any new vAG requests. When a new vAG is required, the vAG characteristics such as the type of traffic it supports and the resource that is needed are obtained from a predefined table. Thus, there exists a table with predefined vAG group IDs for the different traffic types mapped to a fixed processing resources requirement. With all this information available, a vMAC placement request is made to the PDM where the current resource levels and performance information are relayed to the PDM to decide on the best eDC to host the vMAC instance mapped to the vAG. If no vAG needs to be mapped to a new vMAC, the NAM continues with its monitoring and discovery process.

5.2.2 Placement Decision Maker

The PDM is the heart of the application in terms of the aim of this study. The pseudo-code for the PDM component is presented in Pseudo-code 5.2. The PDM's job is to select the most

Pseudo Code 5.1 Network Activity Monitor (NAM) Pseudo-code

```
   Input: [void]
   Output: [vAG and eDC information]
 1: repeat
 2:     Get IDs of active eDCs
 3:     Get eDCs' Current Resources levels for the VIM
 4:     Get eDCs' Current Performance from the VIM
 5:     if New vAG required then
 6:         Get vAG type
 7:         Get vAG Resource and Performance Requirements from a predefined table
 8:         Request new vMAC placement        ▷ flag PDM. Send vAG, and eDC parameters
 9:     end if
10: until forever
```

suitable eDC host based on information received from the NAM. The PDM uses the IDs of the active eDC nodes, the resource levels and the performance indicators to run a heuristic algorithm in search of the most suitable eDC that can host the vMAC instance with a minimal response time while respecting the constraints of the eDC. When the PDM has selected the best eDC, the ID of the selected eDC, the information about the required resources for the vMAC placement and the traffic type it supports is sent to the NFV orchestrator which interacts with the VIM and the VNF manager to instantiate a new vMAC function. Once the instantiation is successful, the PDM updates the routing table for the SDN controller to route traffic associated with the vAG to the eDC that is mapped to the vMAC to the specified eDC and consequently processes the frames. The selection of the best eDC is addressed in the subsequent sections.

Pseudo Code 5.2 Placement Decision Maker (PDM) Pseudo-code

```
   Input: [eDC List, eDC resource levels]                          ▷ from NAM
   Output: [vMAC information, ID of selected eDC]                  ▷ to VIM
 1: repeat
 2:     if New vMAC Request then                                  ▷ flag raised by NAM
 3:         Select best eDC node from eDC list                    ▷ run proposed algorithm
 4:         Request new vMAC Instance                             ▷ with output parameters
 5:     end if
 6:     if vMAC instantiated successfully then                    ▷ flag from NFV-MANO
 7:         Update Routing matrix                                 ▷ vAG-to-eDC route
 8:     end if
 9: until forever
```

5.3 Virtual MAC Deployment Problem Formulation

The objective is to find the best eDC for hosting the GW vMAC instances to support each vAG while respecting the deployment resource capacity bounds and performance factors.

5.3.1 Virtual GW MAC Instances and Candidate Placement Location

A one-to-one mapping of vAGs and the vMAC instances is considered. Thus, each vAG is linked to exactly one vMAC instance in the eDC. This approach is referred to as a Single-feature Single-Request (SFSR) approach [113]. The binary variable $G_{v,r}$ is used to represent the mapping of a vAG r to a vMAC instance v, where a 1 indicates a mapping and a 0 indicates no mapping. Each pGW node consists of a varying number of vAGs that are mapped to exactly one vMAC instance. Thus, there are several vMAC deployment requests from the vAGs made to execute the GW MAC functions virtually in the eDC. The vMAC instances are heterogeneous since different traffic types are used to characterise their resource requirement. A vMAC instance is denoted as v. Each vMAC instance is associated with a traffic type q. Hence a vMAC instance of traffic q can be denoted as v_q. Moreover, each vMAC instance is associated with a virtual access group r. Therefore a vMAC instance belonging to a specific traffic type and a specific vAG can be denoted as $v_{q,r} \in V$, where V is a set of vMAC instances to be deployed. A vMAC instance of traffic type q belonging to vAG r can be placed in an eDC node selected. There is n_e number of distributed eDCs to be considered where each eDC node is denoted as $e \in E$. Therefore if a vMAC instance is deployed in any of the candidate locations, the notation $v_{q,r,e}$ is used. Each vMAC instance requires a minimum amount of eDC virtual computing or processing resources to execute the MAC related functions. Therefore the virtual processing resources required or used by a vMAC instance can be denoted as C_v. In addition to the virtual processing resources, each vMAC instance consumes a portion of the network bandwidth resources of the eDC which is defined by the traffic load (traffic arrival rate) at the eDC node when the vMAC instance is placed.

5.3.2 Virtual GW MAC Placement Decision factors

The placement of the vMAC instance depends on several factors. These factors must be taken into consideration when selecting the best eDC for placing a vMAC instance because they can negatively affect the performance of the network. The VNO provides the virtualisation services to customers (pGW owners) at a cost and customers also have budget constraints. Therefore, the deployment of vMAC instances in the eDC is limited by the number of resources that the VNO offers to the customer for which the customer pays. As the vMAC instances are being deployed in the eDC, the virtual processing resource capacity of the eDC used for virtualisation C_e gets consumed and gradually depletes. The depletion of resources is important for characterising the degradation of QoS of the vMAC instances' functions. To deploy more vMAC requests as the number of vAGs increases, it is imperative to know the number of residual processing resources C_e^{rs} in each eDC to prevent a situation whereby a vMAC placement request is granted however the resources available are not enough to run the vMAC instance. The residual processing resources are expressed in 5.1 as the difference between the sum of all the computational resources used by all the deployed vMAC instances $v' \in V'$ in the eDC and the offered computational resource capacity of the eDC C_e. These processing resources are provisioned as virtual resources.

$$C_e^{rs} = C_e - \sum_{v' \in V'} C_{v'}, \forall e \in E \tag{5.1}$$

Also, the network traffic bandwidth resources of the eDC characterise the amount of traffic in and out of the eDC. The eDC allocates the traffic bandwidth capacity of the eDC for virtualisation denoted as B_e in bits per second which defines the maximum allowed transmission rate in and out of the eDC for GW MAC virtualisation. Therefore, knowing the average MAC frame length per traffic type and the average frame arrival rate at the eDC, the load on the eDC node in bits per second (aggregated traffic rate from the scheduler to the eDC node) denoted as λ_e can be obtained. The residual network bandwidth B_e^{rs} can be obtained as the

difference between the network bandwidth resource capacity and the total traffic load to the eDC as derived in 5.2.

$$B_e^{rs} = B_e - \lambda_e, \forall e \in E \tag{5.2}$$

Also, the stability of the virtual network link to the e^{th} eDC is affected by the placement of vMAC instances. When more vMAC instances are placed, the load on the virtual network increases and may vary depending on the choice of eDC. Therefore to select the best eDC placement location, stability must be considered. The stability of the virtual network link to an eDC is denoted as U_e, which ranges between 0 and 1 where 0 indicates low stability and 1 represents high stability. To establish U_e, the utilisation of each node is used for each route to the eDC. Thus, the probability that any of the nodes visited will have an arrival rate that is higher than the service rate. This is computed in 5.3 where $\rho_{e,j}$ is the utilisation of the $j, ..., nr_e$ node visited when the traffic of a vMAC instance is routed to the e^{th} eDC such that $\rho_{e,j} = 1,\ if\ \rho_{ej} > 1$.

$$U_e = \prod_{j=1}^{nr_e} (1 - \rho_{e,j}), \forall e \in E \tag{5.3}$$

5.3.3 Virtualisation Delay Cost of Placement

The media access throughput over the virtualised network is greatly influenced by the delay of the virtualisation processes, which in turn affects the total number of vAGs that can be virtualised. If there is a high virtualisation delay, fewer vAGs will be deployed which implies that fewer devices will be accommodated for a certain throughput level. The placement of each vMAC instance in the eDC influences the overall virtualisation delay because the placement of a vMAC is mapped to a vAG which implies that the load on the virtual network and hence on the eDC node increases whenever a vMAC instance is placed. The virtualisation delay for the specific traffic type is already established in 4.33. Therefore the virtualisation delay imposed

due to the placement of a vMAC instance in an eDC denoted as $T^{vr}_{v,e}$ can be written based on 4.33 as given in 5.4. Thus, whenever a vMAC instance is placed in an eDC, a vAG is established and the encapsulation and decapsulation delay increases as well as the virtual network response time to a specific eDC.

$$T^{vr}_{v,e} = 2T^{ecp}_{v,e} + T^{vn}_{v,e} \tag{5.4}$$

5.3.4 Binary Integer Linear Programming Formulation

The VNO needs to ensure that the response time of the virtual network is reduced to accommodate more vAGs and therefore improve the robustness of the virtual network as well as the reliability of the GW MAC functions. The eDC contributes significantly to the virtual network response because the eDC resources are shared with other pGW nodes or customers. As a result, delays are imminent when more vAGs need to be virtualised. Therefore, the objective is to find a suitable eDC host where the vMAC instance request can be placed such that the virtualisation delay is minimised while accommodating many vAG as possible. To find the best host that minimises the overall virtualisation latency, the objective function is formulated based on binary linear Integer Programming. The Binary decision variable $D_{v,e}$ described in 5.5 is used to represent whether an eDC host e, is selected or not selected as the host of the vMAC instance request v.

$$D_{v,e} = \begin{cases} 1, & if\ eDC\ e\ is\ selected\ as\ the\ host\ of\ vMAC\ v \\ 0, & otherwise \end{cases} \tag{5.5}$$

The objective function is therefore formulated as

$$min \sum_{v \in V} \sum_{e \in E} T^{vr}_{v,e} D_{v,e}, \tag{5.6}$$

subject to

$$\sum_{e \in E} D_{v,e} = 1, \forall v \in V, \tag{5.7}$$

$$\sum_{r}^{n_{ag}} G_{v,r} = 1, \forall v \in V, \tag{5.8}$$

$$C_e^{rs} > 0, \forall e \in E, \tag{5.9}$$

$$B_e^{rs} > 0, \forall e \in E, \tag{5.10}$$

$$U_e \geq U_e^{min}. \tag{5.11}$$

In 5.6, the objective function minimises the total virtualisation delay for executing the vMAC instances specifically in the eDC nodes. The first constraint in 5.7 ensures that there is only one eDC host that executes a particular vMAC instance. This constraint is important to avoid the duplication of vMAC instances in another host. The duplication of vMAC will therefore impact the availability of resources because resources that could be used to host a different vMAC instance are used to host a redundant vMAC. The constraint in 5.8 ensures that there is a one-to-one mapping of vAGs and the vMAC instance. Thus, each vAG is linked to exactly one vMAC instance, hence each frame from a particular vAG uses only one vMAC instance to execute the MAC functions. The constraint in 5.9 ensures that the total amount of processing resources used by all the vMAC instances in the eDC is limited to or does not exceed the total offered virtual resources. This places a capacity bound to avoid the over-provisioning of computing/processing resources. This is important because, if there is over-provisioning of

eDC processing resources beyond the offered resource capacity, there could be performance interference with other applications and services being run on the same eDC. The constraint in 5.10 also places a capacity bound on the amount of network bandwidth used which is limited by the bandwidth resources offered by the eDC. The constraint in 5.11 ensures that the virtual network stability as a result of the vMAC instances deployed is kept above a recommended threshold U_e^{min} to prevent overloading of the virtual network and subsequently the eDC.

5.4 Virtual MAC Placement Algorithm

To solve the binary integer programming, an online-based heuristic algorithm is proposed and employed since the vMAC requests are generated randomly for any traffic class. In addition, one of the key factors that must be considered in the design of an algorithm for the vMAC placement is the algorithm execution time and time complexity. The suitable eDC node must be found in the shortest time possible to ensure that the vMAC deployment process is executed fast enough. A heuristic algorithm is generally designed in such a way as to solve the problem in a time-efficient manner. The proposed heuristic algorithm is compared with three reference algorithms. The first is an exhaustive algorithm that minimises the objective function by testing all eDC nodes. The second and third algorithms used for comparison are the randomised placement and a round-robin placement approach which do not take into account the minimisation of the virtualisation delay.

5.4.1 Proposed Online-based Placement Algorithm

To solve the vMAC placement problem in the quickest time possible, the placement algorithm is designed in two main phases. The idea is to first obtain a set of suitable eDC candidates using short-listing criteria to minimise the search space. The short-listed candidates are then sorted based on the virtual network stability criteria that affect virtualisation delay. The short-listing and sorting phase is referred to as the Short-and-Sort phase. The short-and-sort phase in Algorithm 5.3 first creates an empty list of candidates. For each eDC, the algorithm checks that the current residual computational resource is sufficient to create a new vMAC

instance and that the maximum network resources are not exceeded when the vMAC instance is deployed. It also checks that the virtual network stability is above the allowed minimum threshold. If these conditions are met, the eDC is added to the list of candidates otherwise the eDC is discarded. After generating the list of candidates, the list is sorted based on the virtual network stability in descending order from the highest stability to the lowest stability. Thus, the eDC that gives the highest stability becomes the first candidate on the list and the eDC with the lowest stability becomes the last candidate on the list.

Algorithm 5.3 Algorithm for the Short-and-Sort phase

Input: [C_v, C^{rs}, B^{rs},U, U^{min}, λ] ▷ for all eDC nodes e in E
Output: [SortedCandidateList]
1: **begin**
2: Create empty CandidateList
3: **for** e **in** E **do**
4: **if** $C_e^{rs} > C_v$ **and** $B_e^{rs} > \lambda_e$ **and** $U_e \geq U_e^{min}$ **then** ▷ check capacity bounds and stability
5: add e to CandidateList
6: **else**
7: discard e
8: **end if**
9: **end for**
10: sort CandidateList by U_e in descending order ▷ mergesort or similar
11: **return** SortedCandidateList
12: **end**

In the next phase, the eDC that yields the least virtualisation delay when the vMAC instance is deployed is selected. This process is referred to as the Select phase. The Select phase is presented in Algorithm 5.4. The sorted short-listed candidates are first obtained by invoking Algorithm 5.3 and the best eDC is guessed as the first candidate in the sorted short-listed list. This is because the first candidate is the candidate with the highest stability effect on the virtual network. When the stability is high, it implies the utilisation of all the nodes visited en route to the eDC is low. Since a high utilisation results in a high response time, there is a high likelihood that the first candidate is the candidate with the least virtualisation delay. For each candidate in the sorted short-listed list except for the first candidate, the virtualisation delay of the first candidate is compared with that of the next candidate in the list. If the

virtualisation delay of the first candidate is less than the next candidate, the first candidate is taken as the best candidate for the placement of the vMAC instance. The algorithm does not check the rest of the candidates since the list is sorted according to stability. This implies that the rest of the candidates will have a higher response time than the first candidate. However, in a practical environment, if there is any external factor that affects the virtualisation delay, the algorithm will test the first candidate with the subsequent candidates. Due to the short-listing and sorting, the search space is reduced and the comparison of the virtualisation delay will be executed between the first and second candidates in the sorted list most of the time, so long as no external factor influences the virtualisation delay in practical environment. This makes the proposed algorithm time-efficient.

Algorithm 5.4 Algorithm for the Select phase

Input: $[C_v, C^{rs}, B^{rs}, U, U^{min}, \lambda]$ ▷ for all eDC nodes e in E
Output: $[e^*]$ ▷ best selected eDC node, e^*

1: **begin**
2: SortedCandidateList ← funcShortSort(C_v, C^{rs}, B^{rs},U, U^{min}, λ)
3: e^* ← SortedCandidateList(head) ▷ head is the initial best eDC
4: **for** e **in** SortedCandidateList except e^* **do**
5: **if** $T^{vr}_{v,e^*} < T^{vr}_{v,e}$ **then** ▷ compare virt. delay of head and next eDC
6: **return** e^*
7: **end if**
8: **end for**
9: **return** No Candidate ▷ NULL output
10: **end**

5.4.2 Reference Placement Algorithms

The three reference algorithms used for comparison with the proposed online heuristic algorithm are presented. The exhaustive placement algorithm is depicted in Algorithm 5.5. Contrary to the proposed algorithm, the reference exhaustive algorithm does not short-list and sort the candidates. It begins by setting the minimum virtualisation delay cost to a high value (infinity). For each eDC in the set, it checks that the current eDC's resource and stability constraints are met for placing the requested vMAC instance. If the constraints are met for that eDC, it compares the minimum virtualisation delay with the virtualisation delay of the current eDC. If

the virtualisation delay of the current eDC is lower than the minimum virtualisation delay, the virtualisation of the current eDC becomes the minimum virtualisation delay and the process is repeated until all the eDC nodes have been exhausted. When all the eDC nodes have been exhausted, the minimum virtualisation delay and the associated eDC node are returned. If the resource constraints and stability requirement are not met and, the current eDC tested is discarded. If the virtualisation delay of the current eDC is greater than the minimum virtualisation delay, the next eDC is selected and the process continues until all the eDC nodes have been exhausted.

Algorithm 5.5 Exhaustive placement algorithm

Input: $[C_v, C^{rs}, B^{rs}, U, U^{min}, \lambda]$ ▷ for all eDC nodes e in E
Output: $[e^*]$ ▷ best eDC node, e^*

1: **begin**
2: $T^{vr}_{v,e^*} \leftarrow \infty$, $e^* \leftarrow NULL$
3: **for** e **in** E **do**
4: **if** $C^{rs}_e > C_v$ **and** $B^{rs}_e > \lambda_e$ **and** $U_e \geq U^{min}_e$ **then** ▷ check capacity bounds and stability
5: **if** $T^{vr}_{v,e} < T^{vr}_{v,e^*}$ **then**
6: $e^* \leftarrow e$ ▷ e is the best eDC e^* if virt. delay of e is less than e^*
7: $T^{vr}_{v,e^*} \leftarrow T^{vr}_{v,e}$
8: **end if**
9: **else**
10: discard e
11: **end if**
12: **end for**
13: **return** e^*
14: **end**

The randomised placement algorithm is depicted in Algorithm 5.6. The algorithm selects a random candidate from the unsorted candidate list and checks to ensure that all the constraints associated with that candidate's eDC are met. If the constraints for the candidate eDC are met, the algorithm returns the selected eDC candidate for vMAC placement. If the constraints are not met, the algorithm removes the randomly selected candidate from the candidate list and selects another random candidate from the remaining candidates in the candidate list until the constraints are met or the candidate list is empty. The round-robin placement algorithm is depicted in Algorithm 5.7. The algorithm selects the first eDC candidate in the unsorted

Candidate list and then rotates the list such that the head of the list becomes the tail of the list. The algorithm then checks if the selected eDC candidate satisfies the constraints. If the constraints are satisfied, the algorithm returns the selected candidate for vMAC placement. Otherwise, the algorithm selects the new head of the Candidate list until the constraints are satisfied or all the candidates have been cycled through.

Algorithm 5.6 Random placement algorithm

```
   Input: [CandidateList, constraints]                    ▷ for all eDC nodes e in E
   Output: [e*]                                           ▷ best eDC node, e*
 1: begin
 2: while CandidateList is not Empty do
 3:     e* ← Random Candidate
 4:     if constraints are satisfied then
 5:         return e*
 6:     else
 7:         Remove Random Candidate from CandidateList
 8:     end if
 9: end while
10: return No Candidate                                 ▷ NULL output
11: end
```

Algorithm 5.7 Round-Robin Placement Algorithm

```
   Input: [CandidateList, constraints]                    ▷ for all eDC nodes e in E
   Output: [e*]                                           ▷ best eDC node, e*
 1: begin
 2: i=1
 3: while i ≤ length(CandidateList) do
 4:     e* ← CandidateList(head)
 5:     Rotate CandidateList by 1
 6:     if constraints satisfied then
 7:         return e*
 8:     end if
 9:     i=i+1
10: end while
11: return No Candidate                                 ▷ NULL output
12: end
```

5.5 Simulation Results and Evaluation

5.5.1 Experimental Setup

The implementation of the proposed vMAC placement approach was done in MATLAB R2022a. Using MATLAB, the mathematical programming was implemented based on the binary integer linear programming problem. The presented algorithms for solving the problem were also implemented pragmatically in MATLAB. To obtain the virtualisation delay based on the different vAGs, whenever a vAG is to be virtualised, the virtual network transition matrices (CE and SE) in Appendix A are updated and the QNA model is executed for the different routes to the different eDC nodes. Based on this, the performance parameters for the placement of each vMAC in an eDC are easily obtained for the placement process. Separate functions in MATLAB are created to simulate the NAM and the PDM processes when a vAG is to be virtualised. The MATLAB program and experiments were run on an HP Pavillion TS 15 Notebook PC running a 64-bit Windows operating system. This is coupled with an Intel Core i5 processor running at 2.30 GHz with 8 GB RAM.

Before simulating the vMAC placement process, each pGW node is initialised with 1 vAG for each traffic type where each vAG is mapped to one vMAC instance distributed equally across all the eDCs. This is the same as the setup in table 4.2. After the initialisation, 24 vAGs were generated for one pGW node (pGW 1) in a random order to be virtualised while the number of vAGs for the remaining pGW (pGW 2, 3 and 4) was kept constant based on initialised number of vAGs. This represents the extra load on the virtual network from other IoT service providers. Since the model parameters for all the pGW nodes are the same, the results obtained for varying the vAGs of one pGW node will be the same if a different pGW node is considered. Based on this, a total of 40 vAGs ([4 pGWs x 4 vAGs] + [1 pGWs x 24 vAGs]) were created and virtualised. The 24 vAGs were randomly generated such that the devices with traffic type A are grouped into 10 vAGs, the devices with traffic type B are grouped into 8 vAGs, the devices with traffic type C are grouped into 4 vAGs and the devices with traffic type D are grouped into 2 vAGs. This is presented in table 5.1 and it conforms with the creation of the

vAGs based on QoS levels whereby high traffic rate applications require more vAGs than low traffic rate applications to accommodate the same number of devices for a given QoS threshold. This is because the probability of an unsuccessful transmission due to collisions is high with high traffic rates than with low traffic rates.

Table 5.1: Number of initialised and randomly generated vAGs for virtualisation for each pGW and traffic type.

pGW	**Initial vAGs**				**vAGs generated during simulation**				**Total**
	A	B	C	D	A	B	C	D	
1	1	1	1	1	10	8	4	2	28
2	1	1	1	1	0	0	0	0	4
3	1	1	1	1	0	0	0	0	4
4	1	1	1	1	0	0	0	0	4

The virtual processing resource consumed or required C_v by each vMAC instance used is 10 MIPS for all traffic types. The virtual processing resource capacity C_e used is the same as in table 4.3. The bandwidth resource capacity B_e for each eDC was initialised with different levels as 10 Gb/s, 0.5 Gb/s,0.1 Gb/s and 0.05 Gb/s for eDC 1, 2, 3 and 4 respectively. The stability threshold U_e^{min} was set to 0.7, 0.5, 0.3 and 0.1 for the traffic types A, B, C and D respectively. For the rest of the parameters and setup, the virtual network and media access control throughput setup and parameters used in Chapters 3 and 4 were used.

5.5.2 Results and Analysis

5.5.2.1 Virtual GW MAC Placement

The number of vMAC instances successfully placed in the eDC nodes is analysed. In Figure 5.2 and table 5.2, the number of vMAC instances deployed is plotted and tabulated respectively for each eDC node and traffic type. The results of the different algorithms are compared. It can be observed that, for the proposed algorithm, most of the vMAC instances are deployed in eDC 1 and eDC 2. The eDC 3 and 4 have 4 vMACs each. This demonstrates that the proposed

algorithm can find the best eDC for the vMAC instance to reduce the cost of placement since eDC 1 and eDC 2 have a higher level of resources and stability followed by eDC 3 and eDC 4. The eDC with the least virtualisation delay is attributed to more processing resources, therefore the algorithm efficiently places the vMAC instances. For the exhaustive algorithm, similarly, most of the vMAC instances are deployed in eDC 1 and eDC 2 where eDC 1 has the highest, followed by eDC 2. This demonstrates the exhaustive algorithm can select the best eDC to host the vMAC instances. For the random case, it can be observed that eDC 3 has the highest number of vMACs, followed by eDC 1, eDC4 and eDC 2. This implies that the vMAC instances are not placed efficiently since eDC 3 has fewer resources than eDC 1 but eDC 3 has more vMACs placed than eDC 1. The same can be said about eDC 3 and 4. In terms of the round-robin case, it can be observed the total number of vMAC instances placed in each eDC is 10. This represents the round-robin process where the eDCs nodes take turns in placing the vMAC instances as requests are randomly generated. Likewise, the round-robin algorithm does not efficiently place the vMAC instances.

Table 5.2: Table of the results of the number of virtual GW MAC instances deployed in each eDC node and each traffic type.

eDC	**Proposed**				**Total**	**Exhaustive**				**Total**
	A	B	C	D		A	B	C	D	
1	8	5	3	2	18	7	7	5	3	22
2	4	5	3	2	14	5	3	1	1	10
3	1	1	1	1	4	1	1	1	1	4
4	1	1	1	1	4	1	1	1	1	4
eDC	**Random**				**Total**	**Round Robin**				**Total**
	A	B	C	D		A	B	C	D	
1	4	5	1	2	12	4	2	2	2	10
2	3	1	1	1	6	2	4	3	1	10
3	3	5	3	2	13	3	3	2	2	10
4	4	1	3	1	9	5	3	1	1	10

5.5.2.2 Virtualisation Delay

The virtualisation delay as a result of the placement strategy is analysed. The results presented in Figure 5.3 show the virtualisation delay for the different traffic types plotted for the vAGs

randomly created. The results of the proposed heuristic placement algorithm are compared with the exhaustive, random and round-robin algorithms. It can be observed for all the traffic types that the proposed placement algorithm and the exhaustive algorithm yield the lowest virtualisation delay when compared to the random and round-robin approaches as the number of vAGs being virtualised increases. This is more pronounce when the number of vAGs is high. This implies that the proposed placement algorithm and the exhaustive algorithm place the vMAC instances linked to the vAGs effectively such that the effect of the eDCs' response time on the virtualisation delay is reduced in comparison to the round-robin and random approaches.

5.5.2.3 Media Access throughput with GW vMAC placement strategy

The results presented in figures 5.4 and 5.5 show the media access throughput results versus the number of connected devices to pGW 1 based on the proposed algorithm as compared with the exhaustive algorithm, the random algorithm, the round-robin algorithm and the GW MAC constrained case. Thus, the number of devices connected to the pGW 1 is varied when all the randomly created or requested vAGs are successfully virtualised.

In Figure 5.4, the throughput is obtained when the 16^{th} vAG is virtualised. It can be observed that the difference between the proposed, the exhaustive, the random and the round-robin approach is minimal for the 16^{th} vAG. This is because with 16 vAGs or 16 vMAC instances the virtualisation delay for all approaches is low. For traffics, C and D, the low traffic rate and low virtualisation delay result in a linearly increasing throughput. Hence, it does not reach the peak throughput for the given number of devices. It can also be observed that all 4 approaches perform better than the constrained case except for traffic C and D where the throughput for the constrained cases is the same with all 4 approaches. This is because of the high traffic rate attributed to traffics A and B and the low traffic rate attributed to traffics C and D. High traffic increases the probability of a failed transmission as a result of GW MAC constraints and the collisions as opposed to the virtualised approach.

In Figure 5.5, the throughput is obtained when the 24^{th} vAG (all vAGs) is virtualised. It can

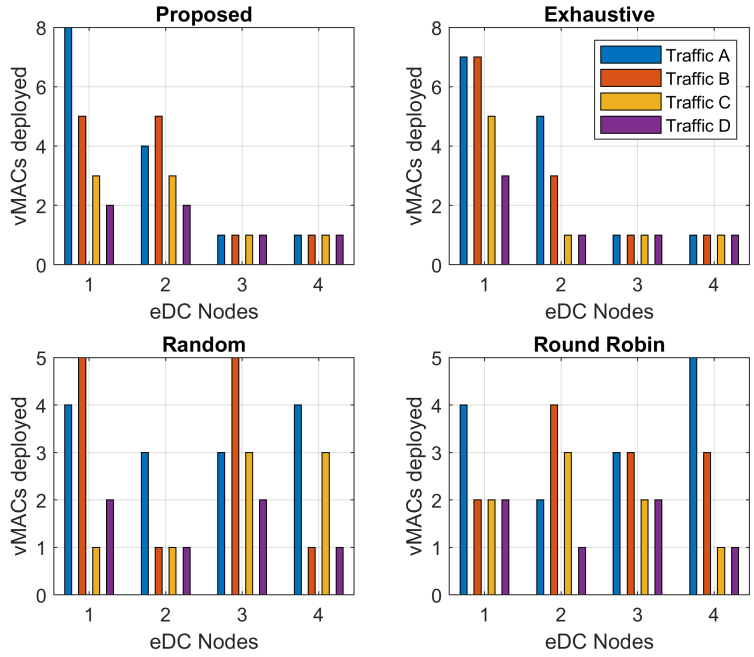

Figure 5.2: Results of the Number of virtual GW MAC instances deployed in each eDC Node and each traffic type.

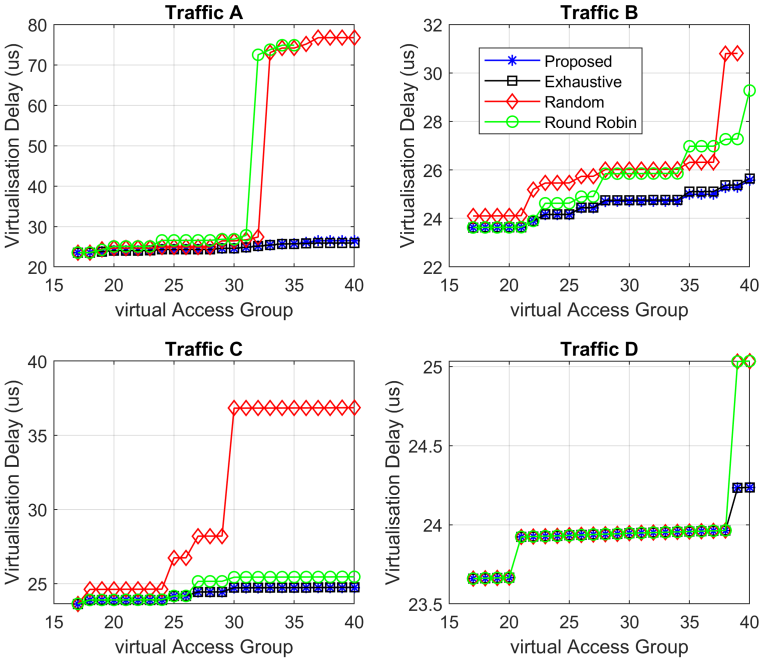

Figure 5.3: Results of the virtualisation delay based on the proposed vMAC placement algorithm compared with the exhaustive, random and round-robin algorithms.

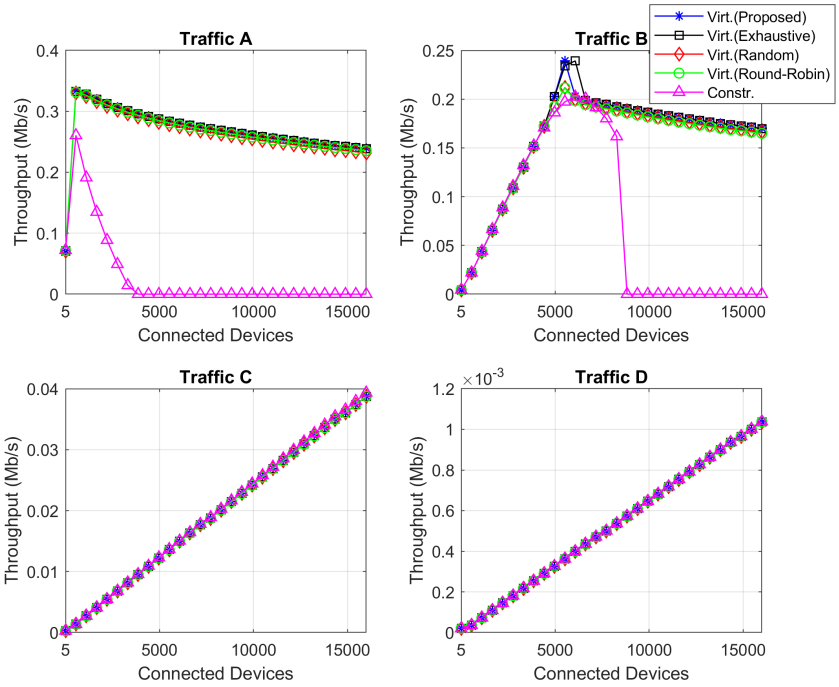

Figure 5.4: Results of the media access throughput based on the proposed placement algorithm when the 16th (half) vAG from the pGW node is virtualised.

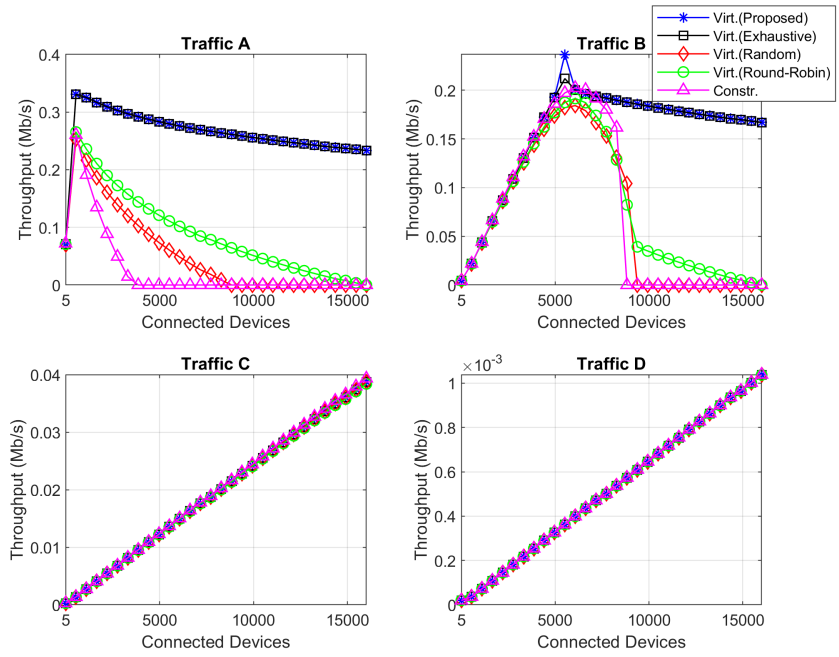

Figure 5.5: Results of the media access throughput based on the proposed placement algorithm when the 24th vAG (all) from the pGW node is virtualised.

be observed that the results based on the proposed placement algorithm and the exhaustive algorithm have superior throughput performance for traffic types A and B. This is because with 24 vAGs/24 vMAC instances the virtualisation delay of the proposed and exhaustive algorithms is significantly lower than the random and round-robin cases. This reduces the probability of a failed transmission due to virtualisation delays. For traffics C and D, the low traffic rate and low virtualisation delay attributed to traffic C and D result in a linearly increasing throughput, hence it does not reach the peak throughput for the given number of devices and all cases. Also, since traffics C and D have a less number of vAGs and a very low traffic rate it reduces the probability of simultaneous transmissions being affected by the virtualisation delay, hence the impact on the virtualisation delay is not significant. For the constrained case, the GW MAC buffer and processing capability (equal to the mean processing resource of all the eDC) have a big impact on the throughput for traffic A and B. In general, the reduced virtualisation delay as a result of the proposed algorithm of the vMAC instances and the deployment of more vMAC instances linked to the vAGs improves the MAC throughput for high IoT traffic rates. For lower traffic rates the throughput increases linearly for the number of devices used.

5.5.2.4 Algorithm Performance

The results presented in Figure 5.6 show the execution time of the different algorithms for each vMAC placement made or each vAG virtualised. It can be observed that the random and round-robin algorithms exhibit the lowest execution time for each vAG virtualised with an average of 0.88 ms and 0.98 ms respectively. This is mainly because the two algorithms do not minimise the virtualisation delay. Thus, both algorithms only check for the capacity bounds and the utilisation which therefore reduces the time for selecting the eDC node for vMAC placement. On the other hand, the exhaustive algorithm generally exhibits the highest execution time for each vAG virtualised with an average of 25.4 ms. However, the proposed algorithm has an average execution time of 14.5 ms which is lower than the exhaustive algorithm but higher than the random and round-robin algorithms. This is because the exhaustive algorithm iterates all the candidate solutions without reducing the search space to find the best solution whereas the proposed heuristic algorithm minimises the search space through the short and sort algorithm

to find the best eDC in the quickest time possible.

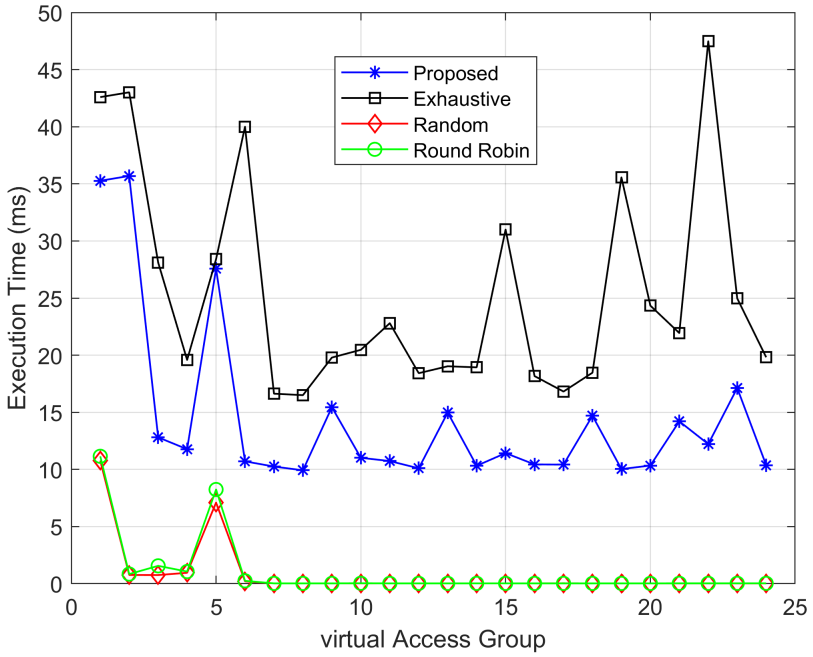

Figure 5.6: Algorithm execution time for each placement request. Average: Proposed = 14.5 ms, Exhaustive =25.54 ms, Random = 0.88 ms, Round Robin =0.98 ms.

To further analyse the performance of the algorithms the time complexity of the algorithms is discussed. The proposed heuristic algorithm has a time complexity of $O(nlog(n))$ as a result of the sorting algorithm for the worst-case scenario assuming a relatively efficient sorting algorithm is used. In the case of the exhaustive algorithm, the time complexity is $O(n)$ since the algorithm does not initially sort the eDC nodes. Similarly, the random and round-robin algorithms have a time complexity of $O(n)$ since all the eDC nodes are exploited in the worst-case scenario. Therefore, the proposed algorithm has high time complexity with higher eDC nodes and has less time complexity for fewer number of eDC nodes when compared with the exhaustive, random and round-robin algorithms for the worst-case scenario. In a practical environment, there are limited eDC nodes in a given area. Moreover, few eDC nodes are sufficient enough to balance the load since several VNFs are already used to handle the functions being virtualised. Therefore, the proposed algorithm may be comparatively useful in a practical environment characterised by a few number of eDC nodes.

5.6 Conclusion

In this chapter, a virtual GW MAC placement strategy was proposed based on an NFV-SDN concept coupled with the online heuristic placement algorithm to find the best eDC for hosting the vMAC instance to minimise the virtualisation delay. The results obtained suggest that the proposed placement strategy can reduce the virtualisation delay and improve the MAC throughput performance for multiple concurrent vAGs better than the random and round-robin vMAC placement strategy. It is also found that the exhaustive algorithm provides similar performance as the proposed heuristic algorithm, although the exhaustive algorithm's execution time and complexity are relatively higher. The results also suggest that for lower traffic rates the throughput increases linearly for the number of devices used whereby there is no significant difference between the constrained GW MAC case and the vMAC with the proposed placement algorithm.

CHAPTER 6. CONCLUSION AND RECOMMENDATIONS

A summary of the main points, the key findings of the research and the recommendations for future work are presented in this chapter.

6.1 Summary of Main Points

In summary, this study was focused on the virtualisation of the GW MAC component of the LS-IoT GW device. The key objective was to translate the GW MAC functions into virtual GW MAC instances which are deployed in a remote edge data centre effectively. This was done by proposing the grouping of devices into virtual access groups where each group of devices communicate with the GW concurrently based on RF virtualisation techniques. The impact of the transmissions from multiple vAGs on the classical GW MAC component with resource constraints and its effect on the 802.11ah MAC throughput were modelled and analysed. In addition, a virtual network framework model was proposed to augment GW MAC functions in the eDC. The model was designed based on the QNA method which synthesises the various network components and processes according to the traffic arrivals and traffic variations at each virtual network node. The delay performance of the virtual network and virtualisation delay were established and incorporated into the MAC throughput model to evaluate the network throughput scalability. Lastly, a virtual GW MAC placement strategy was proposed to manage and effectively deploy the vMAC instances in such a way that the overall virtualisation delay is minimised. This was done by proposing an SDN-NFV vMAC placement application which executes the proposed vMAC placement process based on a binary integer linear programming problem and was solved using a heuristic algorithm that finds the most suitable eDC node for hosting the vMAC.

6.1.1 Study Findings

The key finding of this study indicates that the classical approach of using a dedicated GW MAC component to manage concurrent transmissions over several vAGs for scalability degrades the MAC throughput performance, especially for high traffic rates. Another key finding is that the proposed virtual network model can support the remote deployment of the GW MAC functions for the different vAGs than using a dedicated GW MAC component which is typically constrained by buffer and processing resources. Ultimately, the introduction of a virtual GW MAC placement strategy to mitigate high virtualisation delays indicates improvements in MAC throughput for high traffic rates. However, for low traffic rates, the throughput increases linearly for the number of connected devices used whereby the MAC throughput performance is the same for the constrained dedicated GW MAC, the random placement and round-robin placement approaches. This implies that high traffic rates in a heterogeneous LS-IoT network may require GW MAC virtualisation for a few devices while very low traffic rates may only require GW MAC virtualisation when the devices are extremely large.

6.2 Recommendations for Future Work

As recommendations for future work, it can be suggested that an approach for characterising the traffic profiles of each vAG based on QoS level can be developed. The integration of other LS-IoT standards or technologies such as the 3GPP standard for 5G mMTC may be considered in the virtualisation framework. The vMAC placement optimisation strategy may also include optimising other parameters related to the RF virtualisation, SDN scheduler and CPRI encapsulation. It is also suggested that a framework can be developed to decide in real-time, the optimisation of high IoT traffic while providing a predefined path for low traffic rates since low traffic rates do not exhibit significant throughput improvements with virtualisation. Lastly, the proposed placement algorithm may be further explored for its optimality and complexity.

BIBLIOGRAPHY

[1] (2015, March) Mobile devices and subscribers vs population. Origin. Quebec, Canada. [Online]. Available: http://www.originesenc.ca/en/article/41-mobile-devices-and-subscribers-vs-population.html

[2] "Ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 2: Sub 1 ghz license exempt operation," *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp. 1–594, 2017.

[3] G. C. Madueno, Č. Stefanović, and P. Popovski, "Reliable reporting for massive m2m communications with periodic resource pooling," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 429–432, 2014.

[4] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[5] A. Rajandekar and B. Sikdar, "A survey of mac layer issues and protocols for machine-to-machine communications," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 175–186, 2015.

[6] C. Goursaud and J.-M. Gorce, "Dedicated networks for iot: Phy/mac state of the art and challenges," *EAI endorsed transactions on Internet of Things*, 2015.

[7] M. Bilal, M. Kang, S. C. Shah, and S.-G. Kang, "Time-slotted scheduling schemes for multi-hop concurrent transmission in wpans with directional antenna," *ETRI Journal*, vol. 36, no. 3, pp. 374–384, 2014.

[8] N. Naik, "Lpwan technologies for iot systems: choice between ultra narrow band and spread spectrum," in *2018 IEEE international systems engineering symposium (ISSE)*. IEEE, 2018, pp. 1–8.

[9] S. Martiradonna, G. Piro, and G. Boggia, "On the evaluation of the nb-iot random access procedure in monitoring infrastructures," *Sensors*, vol. 19, no. 14, p. 3237, 2019.

[10] S. S. Zaghloul, "The mutual effect of virtualization and parallelism in a cloud environment," in *AFRICON, 2013*. IEEE, 2013, pp. 1–5.

[11] X. Costa-Pérez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 27–35, 2013.

[12] H. K. Patil and T. M. Chen, "Wireless sensor network security: The internet of things," in *Computer and Information Security Handbook*. Elsevier, 2017, pp. 317–337.

[13] D. Rountree, *Security for Microsoft Windows system administrators: introduction to key information security concepts*. Elsevier, 2011.

[14] D. L. Shinder, T. W. Shinder, C. Todd, and L. Hunter, "Chapter 1 - mcsa/mcse 70-291: Reviewing tcp/ip basics," in *MCSA/MCSE (Exam 70-291) Study Guide*. Syngress, 2003, pp. 1–96. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978193183692050007X

[15] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran, "Mac-layer scheduling in cognitive radio based multi-hop wireless networks," in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*. IEEE, 2006, pp. 10–pp.

[16] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in lte-advanced networks: Issues and approaches," *IEEE communications Magazine*, vol. 51, no. 6, pp. 86–93, 2013.

[17] P. Suriyachai, U. Roedig, and A. Scott, "A survey of mac protocols for mission-critical applications in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 240–264, 2011.

[18] Y. Liu, C. Yuen, X. Cao, N. U. Hassan, and J. Chen, "Design of a scalable hybrid mac protocol for heterogeneous m2m networks," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 99–111, 2014.

[19] M. Doudou, D. Djenouri, N. Badache, and A. Bouabdallah, "Synchronous contention-based mac protocols for delay-sensitive wireless sensor networks: A review and taxonomy," *Journal of Network and Computer Applications*, vol. 38, pp. 172–184, 2014.

[20] C. El Fehri, M. Kassab, S. Abdellatif, P. Berthou, and A. Belghith, "Lora technology mac layer operations and research issues," *Procedia computer science*, vol. 130, pp. 1096–1101, 2018.

[21] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of lorawan channel access," in *2016 International conference on engineering and telecommunication (EnT)*. IEEE, 2016, pp. 10–14.

[22] R. Harwahyu, R.-G. Cheng, C.-H. Wei, and R. F. Sari, "Optimization of random access channel in nb-iot," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 391–402, 2017.

[23] I. Leyva-Mayorga, C. Stefanovic, P. Popovski, V. Pla, and J. Martinez-Bauset, "Random access for machine-type communications," *Wiley 5G Ref: The Essential 5G Reference Online*, pp. 1–21, 2019.

[24] K. A. Aldahdouh, K. A. Darabkh, W. Al-Sit *et al.*, "A survey of 5g emerging wireless technologies featuring lorawan, sigfox, nb-iot and lte-m," in *2019 International conference on wireless communications signal processing and networking (WiSPNET)*. IEEE, 2019, pp. 561–566.

[25] A. Alexiou and A. Gotsis, "Packet scheduling strategies for machine-to-machine (m2m) communications over long-term evolution (lte) cellular networks," in *Machine-to-machine (M2M) Communications*. Elsevier, 2015, pp. 173–186.

[26] J. Kim, J. Lee, J. Kim, and J. Yun, "M2m service platforms: Survey, issues, and enabling technologies," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 61–76, 2013.

[27] Y. Y. Kwon, Y. Y. Fang, and H. Latchman, "A novel mac protocol with fast collision resolution for wireless lans," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 2. IEEE, 2003, pp. 853–862.

[28] P. K. Sahoo and J.-P. Sheu, "Modeling ieee 802.15. 4 based wireless sensor network with packet retry limits," in *Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*. ACM, 2008, pp. 63–70.

[29] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an ieee 802.11 network," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 2. IEEE, 2003, pp. 844–852.

[30] T. Van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 171–180.

[31] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5g: Physical and mac-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, 2016.

[32] K. Zheng, F. Hu, W. Wang, W. Xiang, and M. Dohler, "Radio resource allocation in lte-advanced cellular networks with m2m communications," *IEEE communications Magazine*, vol. 50, no. 7, pp. 184–192, 2012.

[33] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, and M. Zorzi, "The challenges of m2m massive access in wireless cellular networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 1–19, 2015.

[34] A. Lo, Y. W. Law, M. Jacobsson, and M. Kucharzak, "Enhanced lte-advanced random-access mechanism for massive machine-to-machine (m2m) communications," in *27th World Wireless Research Forum (WWRF) Meeting*. WWRF27-WG4-08,, 2011, pp. 1–5.

[35] R. Han, Y. Gao, C. Wu, and D. Lu, "An effective multi-objective optimization algorithm for spectrum allocations in the cognitive-radio-based internet of things," *IEEE Access*, vol. 6, pp. 12 858–12 867, 2018.

[36] H. Zhai, Y. Kwon, and Y. Fang, "Performance analysis of ieee 802.11 mac protocols in wireless lans," *Wireless communications and mobile computing*, vol. 4, no. 8, pp. 917–931, 2004.

[37] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: Use cases, challenges and approaches," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 119–125, 2018.

[38] 3rd Generation Partnership Project (3GPP), "Study on latency reduction techniques for lte (release 14)," *document TR 36.881*, pp. 1–300, 2016.

[39] N. Vasanthi and S. Annadurai, "Aws: Asynchronous wakeup schedule to minimize latency in wireless sensor networks," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, vol. 1. IEEE, 2006, pp. 7–pp.

[40] O. Tickoo and B. Sikdar, "Queueing analysis and delay mitigation in ieee 802.11 random access mac based wireless networks," in *IEEE INFOCOM 2004*, vol. 2. IEEE, 2004, pp. 1404–1413.

[41] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive internet of things for industrial applications: Addressing wireless iiot connectivity

challenges and ecosystem fragmentation," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 28–33, 2017.

[42] X. Jiang, Z. Pang, R. N. Jansson, F. Pan, and C. Fischione, "Fundamental constraints for time-slotted mac design in wireless high performance: The realistic perspective of timing," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 4135–4140.

[43] T. Schmid, O. Sekkat, and M. B. Srivastava, "An experimental study of network performance impact of increased latency in software defined radios," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. ACM, 2007, pp. 59–66.

[44] K. Langendoen, "Medium access control in wireless sensor networks," *Medium access control in wireless networks*, vol. 2, pp. 535–560, 2008.

[45] S. R. Pokhrel, H. L. Vu, and A. L. Cricenti, "Adaptive admission control for iot applications in home wifi networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2731–2742, 2019.

[46] T. Zhong, M. Zhan, and W. Hong, "Congestion control for industrial wireless communication gateway," in *2010 International Conference on Intelligent Computation Technology and Automation*, vol. 1. IEEE, 2010, pp. 1019–1022.

[47] N. Paliwal, "Load balancing through intermediate gateway traffic sharing model (igtsm) for wmn," *International Journal of Computer Applications*, vol. 100, no. 3, 2014.

[48] S. Bhandari, S. K. Sharma, and X. Wang, "Device grouping for fast and efficient channel access in ieee 802.11 ah based iot networks," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018, pp. 1–6.

[49] X. Lei and S. H. Rhee, "Performance improvement of sub 1 ghz wlans for future iot environments," *Wireless Personal Communications*, vol. 93, no. 4, pp. 933–947, 2017.

[50] M. Qutab-ud din, "Enhancements and challenges in ieee 802.11 ah-a sub-gigahertz wi-fi for iot applications," Master's thesis, 2015.

[51] H. Xu, Q. Xue, and A. Ganz, "Adaptive congestion control in infrastructure wireless lans with bounded medium access delay," in *International Mobility and Wireless Access Workshop.* IEEE, 2002, pp. 44–49.

[52] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358–380, 2014.

[53] H. Wen, P. K. Tiwary, and T. Le-Ngoc, "Wireless virtualization," in *Wireless Virtualization.* Springer, 2013, pp. 41–81.

[54] R. Herrero, "Protocol stack virtualization support in IOT," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 11, p. e4340, 2021.

[55] S. Behrad, E. Bertin, S. Tuffin, and N. Crespi, "A new scalable authentication and access control mechanism for 5G-based IOT," *Future Generation Computer Systems*, vol. 108, pp. 46–61, 2020.

[56] V. Theodorou and M.-E. Xezonaki, "Network slicing for multi-tenant edge processing over shared IOT infrastructure," in *2020 6th IEEE Conference on Network Softwarization (NetSoft).* IEEE, 2020, pp. 8–14.

[57] H.-C. Hsieh, J.-L. Chen, and A. Benslimane, "5G virtualised multi-access edge computing platform for IOT applications," *Journal of Network and Computer Applications*, vol. 115, pp. 94–102, 2018.

[58] I. Miladinovic and S. Schefer-Wenzl, "NFV enabled IOT architecture for an operating room environment," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT).* IEEE, 2018, pp. 98–102.

[59] J. Vestin, "Cloudmac: access point virtualization for processing of wlan packets in the cloud," *Bachelor Thesis*, 2012.

[60] B. Fan, H. Tian, Y. Zhang, and Y. Zhang, "Virtual MAC concept and its protocol design in virtualised heterogeneous wireless network," *IET Communications*, vol. 11, no. 1, pp. 53–60, 2017.

[61] D. Garlisi, F. Giuliano, I. Tinnirello, P. Gallo, F. Gringoli, and G. Bianchi, "Deploying virtual MAC protocols over a shared access infrastructure using MAClets," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 17–18.

[62] T. Nagai and H. Shigeno, "A framework of ap aggregation using virtualization for high density wlans," in *2011 Third International Conference on Intelligent Networking and Collaborative Systems*. IEEE, 2011, pp. 350–355.

[63] A. Leivadeas, C. Papagianni, E. Paraskevas, G. Androulidakis, and S. Papavassiliou, "An architecture for virtual network embedding in wireless systems," in *2011 First International Symposium on Network Cloud Computing and Applications*. IEEE, 2011, pp. 62–68.

[64] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "Openran: a software-defined ran architecture via virtualization," *ACM SIGCOMM computer communication review*, vol. 43, no. 4, pp. 549–550, 2013.

[65] G. O. Pérez, J. A. Hernández, and D. Larrabeiti, "Fronthaul network modeling and dimensioning meeting ultra-low latency requirements for 5g," *Journal of Optical Communications and Networking*, vol. 10, no. 6, pp. 573–581, 2018.

[66] G. O. Pérez, J. A. Hernández, and D. L. López, "Delay analysis of fronthaul traffic in 5g transport networks," in *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB)*. IEEE, 2017, pp. 1–5.

[67] A. Gowda, J. A. Hernández, D. Larrabeiti, and L. Kazovsky, "Delay analysis of mixed fronthaul and backhaul traffic under strict priority queueing discipline in a 5g packet transport network," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 6, p. e3168, 2017.

[68] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Analytical modeling for virtualized network functions," in *2017 IEEE international conference on communications workshops (ICC Workshops)*. IEEE, 2017, pp. 979–985.

[69] L. Caeiro, F. D. Cardoso, and L. M. Correia, "Wireless access virtualisation: Addressing virtual resources with different types of requirements," in *2014 European Conference on Networks and Communications (EuCNC)*. IEEE, 2014, pp. 1–5.

[70] I. Sarrigiannis, K. Ramantas, E. Kartsakli, P.-V. Mekikis, A. Antonopoulos, and C. Verikoukis, "Online vnf lifecycle management in an mec-enabled 5g iot architecture," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4183–4194, 2019.

[71] C. Morin, G. Texier, C. Caillouet, G. Desmangles, and C.-T. Phan, "Vnf placement algorithms to address the mono-and multi-tenant issues in edge and core networks," in *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*. IEEE, 2019, pp. 1–6.

[72] S. I. Kim and H. S. Kim, "A vnf placement method based on vnf characteristics," in *2021 International Conference on Information Networking (ICOIN)*. IEEE, 2021, pp. 864–869.

[73] A. Leivadeas, M. Falkner, I. Lambadaris, M. Ibnkahla, and G. Kesidis, "Balancing delay and cost in virtual network function placement and chaining," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 433–440.

[74] F. A. de Figueiredo, R. Mennes, I. Jabandžić, X. Jiao, and I. Moerman, "A baseband wireless spectrum hypervisor for multiplexing concurrent ofdm signals," *Sensors*, vol. 20, no. 4, p. 1101, 2020.

[75] C. Musengimana and S.-E. Yoo, "Real-time scheduling based on multi-channel communication in ieee 802.15. 4 industrial internet of things (iiot)," in *2018 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*. IEEE, 2018, pp. 371–374.

[76] J. Cha, H. Jin, and D. K. Sung, "Multi-channel csma/ca based smart utility networks," in *2011 6th International ICST Conference on Communications and Networking in China (CHINACOM)*. IEEE, 2011, pp. 6–10.

[77] H.-N. Dai, K.-W. Ng, R. C.-W. Wong, and M.-Y. Wu, "On the capacity of multi-channel wireless networks using directional antennas," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 628–636.

[78] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for iot devices," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 204–216.

[79] M. J. A. Jude and V. Diniesh, "Dacc: dynamic agile congestion control scheme for effective multiple traffic wireless sensor networks," in *2017 International conference on wireless communications, signal processing and networking (WiSPNET)*. IEEE, 2017, pp. 1329–1333.

[80] W. Sun, M. Choi, and S. Choi, "IEEE 802.11 ah: A long range 802.11 WLAN at sub 1 ghz," *Journal of ICT standardization*, vol. 1, no. 1, pp. 83–108, 2013.

[81] N. T. Dao and R. A. Malaney, "A new markov model for non-saturated 802.11 networks," in *2008 5th IEEE Consumer Communications and Networking Conference*. IEEE, 2008, pp. 420–424.

[82] G. Bianchi *et al.*, "Performance analysis of the IEEE 802. 11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[83] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on ieee 802.11 ah: An enabling networking technology for smart cities," *Computer Communications*, vol. 58, pp. 53–69, 2015.

[84] M. Kist, J. Rochol, L. A. DaSilva, and C. B. Both, "Sdr virtualization in future mobile networks: Enabling multi-programmable air-interfaces," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[85] J. V. Sudarev, L. B. White, and S. Perreau, "Performance analysis of 802.11 CSMA/CA for infrastructure networks under finite load conditions," in *2005 14th IEEE Workshop on Local & Metropolitan Area Networks*. IEEE, 2005, pp. 6–pp.

[86] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *IEEE/ACM Transactions on networking*, vol. 15, no. 1, pp. 159–172, 2007.

[87] A. Hazmi, B. Badihi, A. Larmo, J. Torsner, M. Valkama *et al.*, "Performance analysis of iot-enabling ieee 802.11 ah technology and its raw mechanism with non-cross slot boundary holding schemes," in *2015 IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2015, pp. 1–6.

[88] G. R. Dattatreya, *Performance analysis of queuing and computer networks*. Crc Press, 2008.

[89] R. Ramaswamy, N. Weng, and T. Wolf, "Characterizing network processing delay," in *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, vol. 3. IEEE, 2004, pp. 1629–1634.

[90] V. Baños-Gonzalez, M. S. Afaqui, E. Lopez-Aguilera, and E. Garcia-Villegas, "Ieee 802.11 ah: A technology to face the iot challenge," *Sensors*, vol. 16, no. 11, p. 1960, 2016.

[91] *IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11ah, 2016.

[92] M. V. Clark, K. K. Leung, B. McNair, and Z. Kostic, "Outdoor ieee 802.11 cellular networks: Radio link performance," in *2002 IEEE International Conference on Com-*

munications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333), vol. 1. IEEE, 2002, pp. 512–516.

[93] S. Aust, R. V. Prasad, and I. G. Niemegeers, "Outdoor long-range wlans: A lesson for ieee 802.11 ah," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1761–1775, 2015.

[94] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "Cloudmacan openflow based architecture for 802.11 mac layer processing in the cloud," in *2012 IEEE Globecom Workshops*. IEEE, 2012, pp. 186–191.

[95] M. Zukerman, "Introduction to queueing theory and stochastic teletraffic models," *arXiv preprint arXiv:1307.2968*, 2013.

[96] E. Gelenbe, G. Pujolle, E. Gelenbe, and G. Pujolle, *Introduction to queueing networks*. Wiley New York, 1998, vol. 2.

[97] W. Whitt, "The queueing network analyzer," *The bell system technical journal*, vol. 62, no. 9, pp. 2779–2815, 1983.

[98] I. Gerhardt and B. L. Nelson, "Approximating performance and traffic flow in nonstationary tandem networks of markovian queues," 2009.

[99] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE journal on selected areas in communications*, vol. 4, no. 6, pp. 833–846, 1986.

[100] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of iot traffic and its impact on lorawan," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[101] A. Bader, H. ElSawy, M. Gharbieh, M.-S. Alouini, A. Adinoyi, and F. Alshaalan, "First mile challenges for large-scale iot," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 138–144, 2017.

[102] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, aloha!" *IEEE access*, vol. 4, pp. 2029–2044, 2016.

[103] M. S. Ali, E. Hossain, and D. I. Kim, "Lte/lte-a random access for massive machine-type communications in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 76–83, 2017.

[104] Y. Zong, C. Feng, Y. Guan, Y. Liu, and L. Guo, "Virtual network embedding for multi-domain heterogeneous converged optical networks: Issues and challenges," *Sensors*, vol. 20, no. 9, p. 2655, 2020.

[105] W. Hou, Z. Ning, L. Guo, Z. Chen, and M. S. Obaidat, "Novel framework of risk-aware virtual network embedding in optical data center networks," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2473–2482, 2017.

[106] *Common Public Radio Interface: eCPRI Interface Specification*, 2nd ed., y Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation and Nokia, 2019.

[107] D. Chitimalla, K. Kondepu, L. Valcarenghi, M. Tornatore, and B. Mukherjee, "5g fronthaul–latency and jitter studies of cpri over ethernet," *Journal of Optical Communications and Networking*, vol. 9, no. 2, pp. 172–182, 2017.

[108] C. P. R. Interface, "Interface specification," *CPRI Specification*, vol. 7, p. 0, 2015.

[109] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM transactions on networking*, vol. 1, no. 3, pp. 344–357, 1993.

[110] J. N. Matthews, E. M. Dow, T. Deshane, W. Hu, J. Bongio, P. F. Wilbur, and B. Johnson, *Running Xen: a hands-on guide to the art of virtualization*. Prentice Hall PTR, 2008.

[111] H. Choi, S. Han, S. Park, and E. Yang, "A cpu provision scheme considering virtual machine scheduling delays in xen virtualized environment," in *TENCON 2009-2009 IEEE Region 10 Conference*. IEEE, 2009, pp. 1–6.

[112] S. Foroutan, B. Akesson, K. Goossens, and F. Petrot, "A general framework for average-case performance analysis of shared resources," in *2013 Euromicro Conference on Digital System Design*. IEEE, 2013, pp. 78–85.

[113] R. Gouareb, V. Friderikos, and A.-H. Aghvami, "Virtual network functions routing and placement for edge cloud latency minimization," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2346–2357, 2018.

[114] M. B. Alaya, Y. Banouar, T. Monteil, C. Chassot, and K. Drira, "Om2m: Extensible etsi-compliant m2m service platform with self-configuration capability," *Procedia Computer Science*, vol. 32, pp. 1079–1086, 2014.

[115] "Machine-to-machine communications (m2m); functional architecture," European Telecommunications Standards Institute, Sophia Antipolis, FRA, Technical Specification, 2013.

[116] T. Macaulay, *RIoT control: understanding and managing risks and the internet of things*. Morgan Kaufmann, 2016.

[117] M. Chen, J. Wan, S. Gonzalez, X. Liao, and V. C. Leung, "A survey of recent developments in home m2m networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 98–114, 2013.

[118] "Study on ran improvements for machine-type communications," Technical Report, 2011.

[119] M. Alam, R. H. Nielsen, and N. R. Prasad, "The evolution of m2m into iot," in *2013 First International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2013, pp. 112–115.

[120] M. Chen, V. C. M Leung, R. Hjelsvold, and X. Huang, "Smart and interactive ubiquitous multimedia services," *Computer Communications*, vol. 35, no. 15, pp. 1769–1771, 2012.

[121] J. Wan, M. Chen, F. Xia, D. L. 0001, and K. Zhou, "From machine-to-machine communications towards cyber-physical systems." *Comput. Sci. Inf. Syst.*, vol. 10, no. 3, pp. 1105–1128, 2013.

[122] X. Yu, P. Navaratnam, and K. Moessner, "Delay model for super-frame based resource reservation in distributed wireless networks," in *International Conference on Mobile Multimedia Communications*. Springer, 2011, pp. 89–104.

[123] Y. Liu, C. Yuen, J. Chen, and X. Cao, "A scalable hybrid mac protocol for massive m2m networks," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 250–255.

[124] P. K. Verma, R. Tripathi, and K. Naik, "A robust hybrid-mac protocol for m2m communications," in *2014 international conference on Computer and communication technology (ICCCT)*. IEEE, 2014, pp. 267–271.

[125] B. Yang, X. Cao, and L. Qian, "A scalable mac framework for internet of things assisted by machine learning," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–5.

[126] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya, "Multichannel mesh networks: challenges and protocols," *IEEE Wireless Communications*, vol. 13, no. 2, pp. 30–36, 2006.

[127] H.-j. Lei, G. Chao, Y.-C. Guo, and Z.-Z. Zhang, "Survey of multi-channel mac protocols for ieee 802.11-based wireless mesh networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 2, pp. 33–44, 2011.

[128] C.-Y. Hsu, C.-H. Yen, and C.-T. Chou, "An adaptive multichannel protocol for large-scale machine-to-machine (m2m) networks," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2013, pp. 1223–1228.

[129] E. Shitiri, I.-S. Park, and H.-S. Cho, "Ormac: A hybrid mac protocol using orthogonal codes for channel access in m2m networks," *Sensors*, vol. 17, no. 9, p. 2138, 2017.

[130] E.-J. Kim, S.-P. Heo, H.-J. Chong, and H.-W. Jung, "Integrated hybrid mac and topology control scheme for m2m area networks," in *2012 14th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2012, pp. 1–5.

[131] L. Alonso, R. Ferrus, and R. Agusti, "Wlan throughput improvement via distributed queuing mac," *IEEE Communications Letters*, vol. 9, no. 4, pp. 310–312, 2005.

[132] A. Samir, M. M. Elmesalawy, A. S. Ali, and I. Ali, "An improved lte rach protocol for m2m applications," *Mobile Information Systems*, vol. 2016, 2016.

[133] A.-T. H. Bui, C. T. Nguyen, T. C. Thang, and A. T. Pham, "Free access distributed queue protocol for massive cellular-based m2m communications with bursty traffic," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–5.

[134] A. Laya, L. Alonso, and J. Alonso-Zarate, "Contention resolution queues for massive machine type communications in lte," in *2015 IEEE 26th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE, 2015, pp. 2314–2318.

[135] L. Tian, J. Famaey, and S. Latré, "Evaluation of the ieee 802.11 ah restricted access window mechanism for dense iot networks," in *2016 IEEE 17th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*. IEEE, 2016, pp. 1–9.

[136] A. Šljivo, D. Kerkhove, L. Tian, J. Famaey, A. Munteanu, I. Moerman, J. Hoebeke, and E. De Poorter, "Performance evaluation of ieee 802.11 ah networks with high-throughput bidirectional traffic," *Sensors*, vol. 18, no. 2, p. 325, 2018.

[137] M. Dong, Z. Wu, X. Gao, and H. Zhao, "An efficient spatial group restricted access window scheme for ieee 802.11 ah networks," in *2016 sixth international conference on information science and technology (ICIST)*. IEEE, 2016, pp. 168–173.

[138] Y. Kim, G. Hwang, J. Um, S. Yoo, H. Jung, and S. Park, "Throughput performance optimization of super dense wireless networks with the renewal access protocol," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3440–3452, 2016.

[139] L. Tian, E. Khorov, S. Latré, and J. Famaey, "Real-time station grouping under dynamic traffic for ieee 802.11 ah," *Sensors*, vol. 17, no. 7, p. 1559, 2017.

APPENDIX A. QNA TRANSITION MATRIX FORMULATION

A.1 Virtual Network Transition Matrix

The transition probabilities for the entire virtual network is defined by the transition matrix $H = [h_{ij}]$ in A.1, whereby the that traffic from a queuing node j (columns) transitions to node i (rows) with a probability of h_q^{sch} or $h_{e,q}^{edc}$ computed using A.3 and A.5 or otherwise 0.

$$H = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ h_1^{sch} & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{n_q}^{sch} & \vdots & \vdots & \vdots & \vdots \\ 0 & h_{1,1}^{edc} & \vdots & \vdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & h_{1,n_q}^{edc} & \vdots \\ \vdots & h_{2,1}^{edc} & \vdots & \cdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \cdots & h_{2,n_q}^{edc} & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & h_{n_e,1}^{edc} & \vdots & \vdots & \vdots \\ \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & h_{n_e,n_q}^{edc} & 0 \end{bmatrix} \tag{A.1}$$

A.2 Transition Probabilities

The probability that a frame from the classifier's queue goes to the q^{th} priority queue of the scheduler h_q^{sch} is computed in A.2. The sum of the number of vAGs serving the q^{th} traffic type in all the pGW nodes is divided by the total number of vAGs. This depends on the matrix CS in A.3 which defines the number of concurrent vAGs each pGW node serves and the traffic types associated with the vAG. The rows represent the pGW nodes $p = 1, ..., n_p$ and the columns represent the traffic types $q = 1, ..., n_q$. The probability that a frame from one of the q^{th} priority queue of the scheduler goes to the q^{th} queue of the e^{th} eDC is denoted as $h_{e,q}^{edc}$ and computed using A.4. Thus, the number of vAGs of traffic type q allocated to the e^{th} eDC is divided by the total number of virtual sectors of traffic type q allocated to all the eDC nodes. This depends on the matrix SE in A.5 which defines the number of vAGs each eDC node hosts and the traffic types associated with the vAGs. The rows represent the eDC nodes $e = 1, ..., n_e$ and the columns represent the traffic types $q = 1, ..., n_q$.

$$h_q^{sch} = \frac{\sum_{p=1}^{n_p} n_{ag,p,q}}{\sum_{q=1}^{n_q} \sum_{p=1}^{n_p} n_{ag,p,q}} \tag{A.2}$$

$$CS = \begin{bmatrix} n_{ag,11} & \cdots & n_{ag,1n_q} \\ \vdots & \vdots & \vdots \\ n_{ag,n_p1} & \cdots & n_{ag,n_pn_q} \end{bmatrix} \tag{A.3}$$

$$h_{e,q}^{edc} = \frac{n_{ag,e,q}}{\sum_{e=1}^{n_e} n_{ag,e,q}} \tag{A.4}$$

$$SE = \begin{bmatrix} n_{ag,11} & \cdots & n_{ag,1n_q} \\ \vdots & \vdots & \vdots \\ n_{ag,n_e1} & \cdots & n_{ag,n_en_q} \end{bmatrix} \tag{A.5}$$

APPENDIX B. JSIMGRAPH VIRTUAL NETWORK SIMULATION MODEL

JSIMgraph is an open-source simulation tool used to simulate complex queuing networks graphically. It provides a discrete-event based simulation for analysing queuing network models. The software tool provides statistical performance parameters about the queuing network and graphically plots the values obtained while the simulation is running. To validate the analytical virtual network model based on the QNA approach, the simulation model in B.1 consisting of source nodes and queueing stations are used to represent the various pGW nodes and the queuing nodes proposed in Figure 4.3. A sink node is also used for customers that have completed their task to exit the system. In the simulation, four customer classes are generated to represent the different IoT traffic types. For each queuing station and source station, the properties are set by the parameters (e.g. service rate, weights etc.) used for the numerical solution. The pGW nodes are configured by linking the customers from the pGW to the different customer classes and their associated interarrival time distribution. This is obtained from the CPRIoE encapsulation rate which is pre-calculated based on the number of vAGs. In each queuing node, the path of the customers are specified based on the routing matrix which is obtained prior and configured in the simulation model. The tools allow for specifying the routing probability of each link. Therefore the probabilities are computed based on the equations in Appendix A and are reconfigured in the simulation model. The queuing station that represents the scheduler is configured to simulate the WRR process proposed in the analytical model. JSIMgraph allows for setting the service capacity according to a Generalised Processor Sharing principle which is a simple approximation of the WRR process.

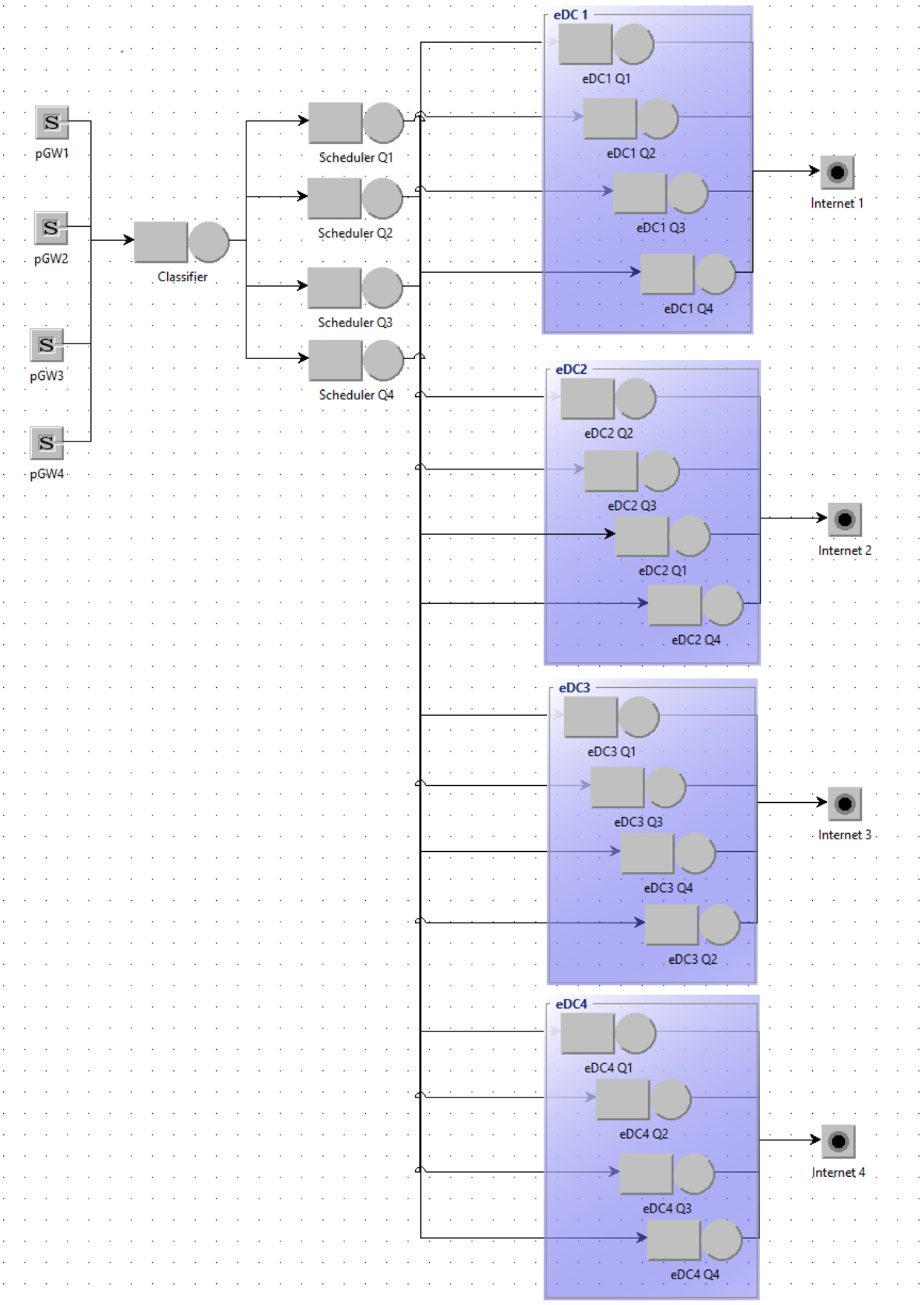

Figure B.1: A JSIMgraph simulation model for the validation of the proposed QNA-based analytical Virtual Network model.

APPENDIX C. MEDIA ACCESS CONTROL IN LARGE-SCALE IOT NETWORK

C.1 Overview of the Large-Scale IoT Network

To characterise an IoT network as a large-scale network, it is fundamental to study the different network domains. This also helps to identify and examine the bottleneck in the LS-IoT network. Various architectures or characterisations of the IoT network have been presented in literature. A typical high-level architecture and functional description of the IoT network is specified by the European Telecommunications Standards Institute (ETSI) [114], [115]. A classical IoT architecture is illustrated in Figure C.1 which highlights the following domains: the M2M local network or capillary network [116], the access network or network edge, the core network or backbone, the cloud data centre, and the application domain.

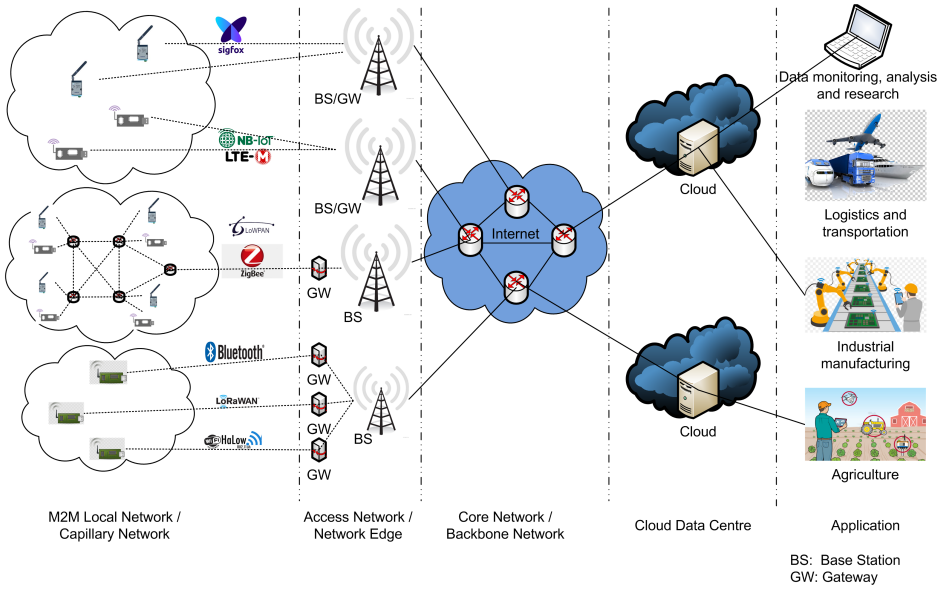

Figure C.1: An illustration of the IoT network with multiple domains, heterogeneous devices, technologies and applications.

The M2M local network domain consists of the M2M devices which provide autonomous sens-

ing or actuation mechanisms [117]. The M2M devices can be connected directly to the GW or Base Station (BS) without an auxiliary GW. The M2M devices can also be interconnected in a star or mesh topology to form a capillary network which is coordinated by a GW device acting as a proxy for a remote BS. In the context of a LS-IoT network, the M2M local network is attributed to an enormous magnitude of wirelessly interconnected heterogeneous M2M devices in a star network topology through a GW device. Such characterisation is widely sensitised by many stakeholders such as the 3rd Generation Partnership Project (3GPP) which predicted that the number of devices in a cell of a typical Long-Term Evolution (LTE) network will exceed 30000 compared to the current average of 50 devices per cell [118]. Moreover, in the LS-IoT context, the M2M local network involves massive distributed sensing and actuation which render large amounts of data, requiring enormous onboard signal and data processing and short-term storage. Depending on the use case, the M2M local network involves frequent node-level media access scheduling, radio transmissions and reception for data exchange in a large-scale context. The access network is made up of GW and BS devices collectively called access network devices. They provide access to a wider third-party network which is the Internet in the context of IoT. In the context of LS-IoT, The access network domain comprises one or more GW or BS devices for adequate long-range coverage in LS-IoT where each GW device serves an enormous magnitude of devices. The access network domain involves the processing of massive heterogeneous traffic arrivals such as short traffic, burst traffic, aperiodic traffic and periodic traffic. The scale of processing at the GW is also high due to the complex multiplexing of traffic arrivals onto the core network. The access network's large-scale attributes also include extremely high communication resource management, utilisation and scheduling over constrained physical resources. Thus, to manage many devices there is high utilisation of computing, memory and communication resources within the GW/BS devices. The core network domain on the other hand represents the classical Internet infrastructure which routes packets between different networks based on the Internet Protocol (IP). It also enables connection from an M2M local network to the cloud data centre. The cloud data centre is an infrastructure consisting of a network of hardware equipped with computing, storage and networking capabilities. The application domain enables logical human interaction with the

M2M local network through specialised services. An Application Programming Interface (API) is typically used to provide the interaction between the cloud data centre and the application services. In the context of LS-IoT, the core network, cloud data centre and application domain are large-scale infrastructures in principle. The core network is concerned with the allocation of many IP addresses and packet routing over the Internet. The cloud data centre and application domains involve the influx of large and complex data that need complex systematic approaches to establish substantial logic for the various services and applications through complex and dynamic API deployment and execution in the context of LS-IoT.

C.2 LS-IoT Network Characterisation

Contrary to application-specific networks or networks with minimal diverse applications such as cellular networks and Wi-Fi (IEEE 802.11) networks, the IoT network builds on many diverse applications, functions and network components. To provide adequate context to the meaning of LS-IoT networks in this paper, the description of the LS-IoT network characteristics is presented in two dimensions based on the functional and physical architecture of the LS-IoT network. The structure of the elements related to the processes such as computing, sensing, protocols, transmissions, etc., in the LS-IoT network constitutes the functional architecture. The structure of the actual physical devices on which all the functional elements are executed such as the M2M devices, GW, BS, Routers, Data Centres, etc., constitute the physical architecture of the network. Both aspects are significantly influenced by the magnitude of applications and services.

Moreover, the characterisation of LS-IoT networks is underpinned by M2M communications which form the core principle behind the entire IoT network concept. M2M communication is described by Alam *et al.* [119] as the utilisation of wireless data connection to establish a link between systems, remote devices, locations and individuals to enable the collection of information, setting of parameters and sending or receiving indications of some phenomena on an enormous number of machines. The concept of M2M is driven by the idea that multiple interconnected devices are highly beneficial and valuable because they easily enable the support

for ubiquitousness, pervasiveness and autonomousness of systems and processes using machines [120], [121]. The pervasive, ubiquitous, smart, intelligent and autonomous characteristics of M2M communication manifests into LS-IoT.

Therefore, all the above culminates into providing context to the large-scale nature of IoT networks. The key network domains and the two-dimensional characteristics of the large-scale nature of the IoT network are discussed below with reference to Fig. C.1. Table C.1 also provides a summary of the LS-IoT characteristics and is presented for each LS-IoT network domain based on their functional or physical characteristics.

C.2.1 Functional Characterisation of LS-IoT

LS-IoT networks involve massive distributed sensing and actuation which render large amounts of data, requiring enormous onboard signal and data processing and short-term storage. Depending on the use case, the M2M local network involves frequent node-level media access scheduling, radio transmissions and reception for data exchange in a large-scale context. The M2M local network may also require dense and complex wireless network routing of frames given the geographic large-scale nature of the M2M local network.

The access network domain of a LS-IoT network involves the processing of massive heterogeneous traffic arrival patterns such as short traffic, burst traffic, aperiodic traffic and periodic traffic. The scale of processing relates to the complex multiplexing of traffic arrivals onto the much wider IP-based network. The access network's large-scale attributes also include extremely high communication resource management, utilisation and scheduling over constrained physical resources for a very large number of M2M devices. Thus, to manage many devices in the access network, complex algorithms and processes are required, and hence, the reason for the high utilisation of computing, memory and communication resources.

In the Core Network, the LS-IoT involves the allocation of many IP addresses and packet routing over the Internet with the reliance on a highly constrained IP address availability. The scale of

IP addresses is directly associated with the number of actively connected devices (GWs, BSs or M2M devices) communicating through the core network. The cloud data centre and application domains involve the influx of large and complex data that need complex systematic approaches to establish substantial logic for the various services and applications through complex and dynamic API deployment and execution in the context of LS-IoT.

C.2.2 Physical Characterisation of LS-IoT

In terms of the physical characteristics of the LS-IoT network, the M2M local network is attributed to the existence of an enormous magnitude of wirelessly interconnected heterogeneous M2M devices. Multiple local M2M networks of devices may coexist. Such characterisation is widely sensitised by many stakeholders such as the 3rd Generation Partnership Project (3GPP) which predicted that the number of devices in a cell of a typical Long-Term Evolution (LTE) network will exceed 30000 compared to the current average of 50 devices per cell [118]. This is attributed to the proliferation of M2M devices in the M2M local network.

The access network domain comprises one or more GW or BS devices for adequate long-range coverage in LS-IoT. However, one main characteristic is the coexistence of heterogeneous access network devices for LS-IoT networks. Due to the diverse applications and use cases of LS-IoT networks such as agricultural and industrial production monitoring, heterogeneous wireless communication technologies are prominent in different M2M devices, and hence, require association with compatible network technologies in the access network.

The core network, cloud data centre and application domain are by nature already large-scale infrastructure for traditional networks, and therefore, their characterisation in terms of LS-IoT networks is more adequately defined in terms of their functional architectural characteristics in LS-IoT networks. However, the core network, cloud data centre and application domain can still be characterised as a high-capacity network infrastructure, a consolidation of data centres to form a cloud platform and a large number of devices connected to the cloud for application services respectively.

Table C.1: A summary of the characteristics of LS-IoT networks based on the respective domains, their functional and physical characteristics.

Type of Characteristics	**M2M Local Network**	**Access Network**	**Core Network, Cloud and Application**
Functional	Massive distributed sensing and actuation. Large amounts of data acquisition. Enormous onboard data processing and storage. Frequent node-level media access scheduling, radio transmission and reception. Dense and complex routing of frames in the capillary network.	Processing of massive dynamic and heterogeneous M2M traffic pattern arrivals. Complex multiplexing of traffic onto the core network. High communication resource management, utilisation and scheduling.	Allocation of many IP addresses. Enormous data storage and computational capabilities in the cloud. Complex and large data analysis for application logic. Dynamic and frequent API interaction for service applications.
Physical	Enormous magnitude of wirelessly interconnected M2M devices. Existence of many heterogeneous M2M devices. Existence of multiple local M2M networks (capillary networks).	A moderate number of access network devices. Coexistence of heterogeneous access network devices.	High-capacity connectivity infrastructure with worldwide access to M2M packets. Distributed data centres are consolidated into one cloud platform infrastructure. A large number of devices supporting IoT logical IoT applications and services such as smartphones and computers connected to the cloud.

C.3 Existing Approaches for Supporting Media Access Scalability in LS-IoT

Several proposed strategies in literature for addressing the media access scalability in LS-IoT exist. These proposed strategies are presented below.

C.3.1 Hybrid Frame-based Approach

Hybrid frame-based approaches are based on dividing the available transmission time into independent service periods whereby each service period is allocated for a MAC specific function and is orchestrated either by the device or by the GW. The transmission time frames are made up of contention-free periods and contention periods [122]. The contention for media access is executed during the Contention Only Period (COP) and transmission is executed during the Transmission Only Period (TOP). Some related works on this approach are presented below.

In [123], the authors propose a scalable frame-based hybrid MAC scheduling protocol for massive M2M networks. The time frame is divided into four distinct sub-frame periods. A Notification Period (NP) and an Announcement Period is proposed to notify devices of the start of a COP and to announce the transmission schedules in the TOP respectively. Contention in the COP is based on the classical p-persistent CSMA/CA media access method. Frame transmissions in the TOP are based on Time Division Multiple Access (TDMA) approach. The authors developed a hybrid scheduling approach for the COP and TOP durations by formulating an optimisation problem which sought to maximise the throughput and find the optimal value of the contention probability and the optimal number of transmissions during COP and TOP respectively. The results presented in their work show better throughput, delay, and frame utilisation when compared with s-ALOHA and TDMA-based approaches. Another frame-based approach is proposed in [124] to combat clock synchronisation failures between a large number of M2M devices which affects scalability and robustness. The technique is based on the same principle presented in [123]. However, the authors optimised the TOP by controlling the number of transmitting devices as the number of nodes increased. The results presented show better throughput than s-ALOHA and TDMA-based approaches. Yang et al. proposed the use

of machine learning to optimise the frame length in a frame-based hybrid approach to achieve a scalable MAC scheduling for IoT networks in [125]. Their proposed strategy is based on predicting the number of IoT devices actively connected to the network and using that information to optimally and dynamically adjust the length of the frame to achieve stable throughput with an increasing number of devices. The authors presented a near-stable throughput performance with increasing devices.

C.3.2 Multichannel Hybrid Approach

In multichannel hybrid approaches, multiple access is granted through multiple frequency channels. Devices may choose frequency channels to transmit on but may still employ contention-based or contention-free based protocols to reduce or avoid possible frame collisions as a result of simultaneous channel access. A rendezvous algorithm is sometimes required to provide coordination of the search for a common channel to exchange frames. The multichannel approach is known to provide improved capacity in wireless networks [126], [127]. Therefore, the use of multichannel has been widely used to address MAC related challenges in LS-IoT networks. Some related work on this approach is presented below.

An adaptive multichannel approach to MAC scheduling in large-scale M2M networks is proposed by authors in [128]. The proposed method is based on a distributed negotiation of media access using a Common Control Channel (CCC). Devices estimate the number of devices, access probability and optimal duration of the negotiation phase. In the negotiation phase of the CCC, transmitting devices contend and reserve channels. Successful devices transmit simultaneously during a transmission phase using the reserved channels and collided devices re-negotiate again. The results presented perform well in terms of near-optimal channel utilisation compared to other references. An approach to MAC scheduling based on orthogonal codes called OrMAC for M2M networks is proposed in [129]. The concept is based on the pre-assignment of orthogonal codes to devices which are used as a media access key, representing orthogonal contention signals. Devices transmit their contention signals (different frequencies) and are then put in a queue until the currently scheduled transmission is completed. The proposed approach per-

forms relatively well in a large-scale scenario. In [130] the authors use a multichannel approach to achieve improved throughput and accommodate devices in an interference range for M2M networks. In their approach, logical channels are created by having sub-frames and time slots within the sub-frames for each channel. All the logical channels are combined to create multiple super-frames. The logical channels are assigned to devices by the BS. Information from the network layer is used to optimise the allocation of the logical channels to devices. The performance of the proposed strategy is evaluated in simulation and shows a relatively improved throughput for an increasing number of devices when compared to a hybrid MAC in literature.

C.3.3 Distributed Queuing Approach

The Distributed Queuing (DQ) is based on multiple queues (typically two queues) whereby devices either enter a queue to resolve a collision or enter a queue to perform transmission. This type of MAC scheme is employed to eliminate the need for back-off periods and to enable scheduling based on QoS requirements [131]. It provides superior performance of MAC scheduling by minimising access delays and increasing throughput for large networks such as LS-IoT networks. Several authors have employed this approach and proposed unique modifications to address MAC scheduling in M2M and LS-IoT networks as presented below.

In [132], authors extended the Distributed Queuing Random Access Procedure (DQRAP) technique to improve the random access process in LTE to support many M2M devices for enabling LS-IoT. In their approach, a virtual LTE frame structure is created such that the preambles used in LTE for media access are virtually mapped to the mini-slots of the DQRAP. A device that has data to be transmitted first selects one of the preambles of a selected virtual group. A new device may select a preamble in a different virtual group based on a proposed request transmission rule to reduce the collision probability in the resolution queue. Devices keep track of the number of collisions in the resolution queue. The results presented show improvement in the access delay for many devices when compared to the classical Enhanced Access Barring (EAB) procedure in LTE advanced (LTE-A). The authors in [133] considered the use of the DQ approach with a proposed modification to eliminate the need for employing class barring

in 5G and LTE-A networks for M2M devices. They proposed multiple virtual queues based on groupings to resolve a collision. The proposed concept estimates the number of collided devices during a random access process. Based on the estimated collided devices, the devices are divided into multiple groups. If the random access contention is unsuccessful the device randomly selects a group to join in the queue. The results presented for their approach show improved performance when compared with the classical access barring scheme. In [134], a strategy is proposed based on DQ to provide a scalable MAC strategy for massive M2M devices in LTE. When a collision is detected during a random access process, the collided devices are split into virtual distributed queues using a proposed algorithm before being scheduled for the transmission opportunity. When a certain number of devices choose the same preamble, a resource block is allocated to the group of devices that collided and is immediately put into a collision resolution queue. The proposed strategy shows improvement in the access delay for a relatively large number of simultaneous message arrivals over the standard random process in LTE.

C.3.4 Group-Synchronised Distributed Coordinated Approach

The group-Synchronised Distributed Coordinated scheduling approach is based on restricting media access to groups of devices within a period. The restricted period is known as the RAW. Within the restricted period, devices contend for channel access in a distributed manner. Restricting media access to a group of devices reduces the average collision probability which enables the network to achieve stable throughput with an increasing number of devices. Group synchronisation is studied in literature and is considered as being very useful for improving the MAC scheduling performance in terms of scalability of the LS-IoT where simultaneous media access contention is highly probable [135], [136]. This is a key motivation for the adoption of group-synchronised distributed media access in the 802.11ah standard to support many M2M devices [2].

One of the standardised applications of the Group-synchronised distributed coordinated approach is in the proposed 802.11ah standard for LS-IoT [2]. In the 802.11ah standard, the

RAW is divided into slots within which the classical Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used to schedule transmissions distributively. The devices are grouped based on a hierarchical structure made up of Pages, Traffic Identification Map (TIM) groups, sub-blocks and devices. This approach reduces overhead in the beacon frames and reduces collision probability. In [137], the allocation of RAW slots is based on a spatial grouping algorithm. The authors propose the provisioning of the slot index based on the geographic location of the device. The coverage area is divided into the same number of RAW slots present in the RAW. The results of the proposed strategy show throughput some improvement. A grouping strategy is proposed in [138] called the Grouped Renewal Access Process. The grouping of devices based on transmission attempts is proposed in this approach to deal with RAW slot crossing challenges in super dense IoT networks. A slot is adopted such that each slot is defined by the number of transmissions within a group of which the end of a slot is the end of the last transmission. The success probability results presented show better success probability when compared to the conventional Distributed Control Function (DCF). The authors in [139] propose a strategy to optimise the RAW parameters dynamically based on traffic conditions in real time. The BS records the number of successful transmissions and the status of a transmission result for a given device. The BS continuously computes the estimated transmission interval for a station and determines the optimal beacon interval, the number of RAW groups or RAW slots and the duration of the RAW slot. Based on the transmission frequency of the device, which is assumed to be predictable, the BS assigns a device to a RAW slot using the optimised values. The results obtained show relative improvement.

APPENDIX D. RESEARCH OUTPUTS

The following papers were published in reputable scientific journals during this study:

1. Sotenga, P.Z., Djouani, K. and Kurien, A.M., 2020. Media access control in large-scale Internet of Things: a review. IEEE Access, 8, pp.55834-55859.
2. Sotenga, P.Z., Djouani, K. and Kurien, A.M., 2023. A virtual gateway media access placement approach for large-scale Internet of Things. Journal of Network and Computer Applications, 209, p.103543.
3. Sotenga, P.Z., Djouani, K. and Kurien, A.M., 2023. A Virtual Network Model for Gateway Media Access Control Virtualisation in Large Scale Internet of Things. Internet of Things, 21, p.100668.

The following papers were presented at reputable scientific conferences and published in the respective conference proceedings during this study:

1. Sotenga, P.Z., Djouani, K. and Kurien, A.M., 2019, September. Evaluation of 802.11 ah TIM and RAW Scheduling Protocol for IoT. In 2019 IEEE AFRICON, pp. 1-6.
2. Sotenga, P.Z., Djouani, K. and Kurien, A.M., 2021. M2M Device Scheduling with GW-MAC Virtualisation in Large-Scale Internet of Things. In Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2021, pp.92-97.
3. Sotenga, P.Z., Djouani, K. and Kurien, A.M., 2022. Massive Internet of Things: A virtual Wi-Fi Halow Media Access Gateway. In Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2022, pp. 219-224.