Auckland University of Technology

Opportunistic Fog Computing for Next-Generation Radio Access Networks

Jofina Jijin

A thesis submitted to Auckland University of Technology
in fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

2021

School of Engineering, Computer and Mathematical Sciences
Faculty of Design and Creative Technologies
Auckland University of Technology

New Zealand

ATTESTATION OF AUTHORSHIP

I, Jofina Jijin, hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university of institution of a higher learning.

14/09/2021

Signature

Date

ABSTRACT

The next-generation radio access networks (RAN) will not only enhance peak data rate and encourage ultra-low-latency applications, but also play a crucial role in connecting a broad range of edge devices resulting from the Internet-of-Thing (IoT) paradigm. This will lead to an exponential growth of data traffic and a need for ubiquitous processing. To support this, researchers have proposed the concept of cloud radio access networks (C-RAN) where client data received by base stations (BSs) are transmitted over fibre links to a commodity cloud platform for processing.

However, the current C- RAN may not be a suitable candidate when it comes to dealing with issues of i) limited backhaul capacity; ii) excessive load concentration on the centralised base band unit (BBU) pool; and iii) challenges of meeting the delay-sensitive requirements of 5G. A promising alternative to C-RAN is fog-based radio access networks (F-RAN), which is based on the philosophy of harnessing the distributed resources of collaborative edge devices to deliver localised RAN services to end users. Still, the current F-RAN is mainly utilising dedicated processing hardware and does not leverage on the available large number of distributed edge devices.

Upon undertaking an extensive literature review, we realised that there exists a considerable research gap in tackling the following issues: i) under-utilisation of resourceful end-user devices and constrained backhaul capacity; ii) optimal resource assignment for computationally intensive tasks; iii) limited flexibility and scalability of current solutions for computation offloading; and iv) secure management of distributed resources. The research undertaken in this thesis aims to provide insights and potential solutions to the aforementioned issues. Simulation and experimental evaluations, along with prototype implementation are also presented.

The key research contributions of this thesis are reported in the following four chapters: **Chapter IV** addresses research gap (i) by proposing an opportunistic fog radio access network (OF-RAN)

which can contribute a scalable solution to the key challenges faced by current RANs and address the issue of service load balancing in this type of RAN. To address research gap (ii), **Chapter V** investigates and resolves the task-node assignment problem in the proposed OF-RAN by utilising a multi-objective optimization approach. **Chapter VI** addresses research gap (iii) by analytically modelling and evaluating the performance of our proposed OF-RAN against existing RANs in order to gain insights into how OF-RAN can complement the existing architectures. **Chapter VII** addresses research gap (iv) by proposing the concept of blockchain-enabled OF-RAN which builds on the inherent security of blockchain decentralization and the collaborative processing of OF-RAN. The concept is investigated by both simulated and real experiments for a federated deep learning application that harnesses the edge devices in OF-RAN for object detection.

ACKNOWLEDGEMENTS

When I write this section of my thesis, the first thought that comes to my mind is “I am so happy, I came so far”. In this journey I would first and foremost thank God for helping me throughout.

I would like to sincerely express my gratitude towards my supervisor Dr Boon-Chong Seet for supporting me with his constant guidance and encouragement to explore new technologies. He gave me the freedom to bring in new ideas and innovations. He demonstrated the importance of explaining ideas in a refined and clear manner both while writing research papers and doing technical presentations. It has been a great pleasure working with him.

Special thanks to my second supervisor Dr. Peter Chong for his support, guidance, and helpful suggestions in this journey. His support has been much valued and appreciated. I am also thankful to all the staff members in the Engineering PhD Team, Postgraduate Team and Scholarship Office for supporting me throughout my PhD journey. Special mention to Josephine Prasad and Jessica Yamamoto for their support, especially during the final year of my PhD.

I am grateful to my parents P.V. Jose and Seevi Jose for supporting and guiding me throughout my life and always believing in me, I am also thankful to my husband’s parents M.D. Johney and Mariamma Johney who have always given their tremendous support and motivation. Special mention to my sister, close friends and family members who were there to chat and encourage me.

Last, but not the least I would like to thank my dearest husband Jijin Johney for being there with me in all my ups and downs, constantly motivating and supporting me. I would have never reached so far without him.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

GLOSSARY

BBU	Base Band Unit
BLER	Block Error Rate
BS	Base Station
COAC	Centralised Opportunistic Access Control
CPRI	Common Protocol Radio Interface
C-RAN	Cloud Radio Access Network
CRRM	Cooperative Radio Resource Management
CUE	Cellular User Equipment
D2D	Device-to-Device communication
DL	Deep Learning
DRAC	Distributed Random Access Control Scheme
D-RAN	Distributed Radio Access Network
DTWN	Digital Twin Wireless Network
DUE	D2D User Equipment
eRRH	Enhanced Remote Radio Head
FAP	Fog Access Point
FCM	Femto Cell Manager
FEC	Forward Error Correction
F-RAN	Fog Radio Access Network
F-UE	Fog User Equipment
HARQ	Hybrid Automatic Repeat Request
ID	Identity
IoT	Internet of Things
IoV	Internet-of-Vehicle
IP	Integer Programming
LTE	Long-Term Evolution
MD	Mobile Device
MEApps	Mobile Edge Applications
MEC	Mobile Edge Computing
MEHosts	Mobile Hosts
mmWave	Millimetre-wave
MOEA/D	Multi-Objective Evolutionary Algorithm using Decomposition
MOP	Multi-objective Problem
NFV	Network Function Virtualization
OFDMA	Orthogonal Frequency Division Multiple Access
OF-RAN	Opportunistic Fog Radio Access Network
oppnet	Opportunistic Resource Utilization Network
PEP	Pair wise Error Probability
PF	Pareto Frontier
PoS	Proof of Service
PoW	Proof of Work

PS	Pareto Set
QoE	Quality of Experience
QoS	Quality of Service
RAP	Radio Access Point
RF	Radio Frequency
RRH	Remote Radio Head
RSU	Roadside Unit
SD	Standard Deviation
SDN	Software Defined Networking
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
TDMA	Time Division Multiple Access
TNA	Task-to-Node Assignment
UD-CRAN	Ultra-dense Cloud Radio Access Network
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
UTRAN	Universal Mobile Telecommunication System Radio Access Network
V2I	Vehicle- to-Infrastructure
v-FAP	Virtual Fog Access Point
VM	Virtual Machine
WSN	Wireless Sensor Network
WTA	Weapon-Target Assignment

CHAPTER I: Introduction

1.1. Motivation and Scope

The fifth generation (5G) and beyond cellular networks will not only cater high-speed and reliable human communication services, but also support communications between a large number of smart objects or 'things' in the coming era of the Internet of Things (IoT) [1]. To sustain these objectives, centralized radio access networks was developed where client data received by base stations (BSs) are transmitted over fiber links to a central unit for processing on specialized hardware [2]. Recently, the concept of cloud radio access networks (C-RAN) was proposed to replace the specialized hardware in centralized RAN with commodity cloud-computing platform to allow for more flexible splitting and allocation of RAN functionalities between radio access points (RAPs) and the cloud, depending on the available cloud resources. However, C-RAN has the shortcomings of: i) constrained backhaul capacity; ii) load concentration on the centralized base band unit (BBU) pool; and iii) difficulty in meeting the ultra low-latency requirements of 5G [3]. More recently, Fog-computing based Radio Access Network (F-RAN) is proposed as a promising candidate to tackle the aforementioned challenges. Fog computing is a paradigm that extends cloud computing by placing cloud-equivalent resources including processing and storage resources at the edge of the network [4]. In literature, fog computing is also considered as a more general concept of mobile edge computing (MEC). F-RAN harnesses the presence of such collaborative edge devices to deliver localized RAN services to end-users [5]. Its main philosophy is to make full use of local radio signal processing, cooperative radio resource management (CRRM) and distributed storage capabilities in edge devices [6]. Through ingestion and processing of end-user tasks close to their sources, F-RAN has potential to meet the stringent latency and bandwidth requirements of 5G services and applications. There is a presence of a large number of other distributed edge

devices in the proximity of FAPs such as WiFi access points, femtocell base stations, and resource-rich end-user devices that can be incentivised to lease their resources and collaboratively serve as 'service' nodes to other end-users. However, the current proposed fog access points (FAPs) of F-RAN have been implemented mainly as dedicated fog servers, or fog-enabled remote radio heads (RRHs) or macrocell base stations [7]. It does not leverage on the presence of a large number of other distributed edge devices in the proximity of FAPs such as WiFi access points, femtocell base stations, and resource-rich end-user devices that can be incentivised to lease their resources and collaboratively serve as 'service' nodes to other end-users. The above-mentioned issues motivated us to propose the opposrtunistic fog radio access network (OF-RAN) which comprises of virtual fog access points (v-FAPs). The v-FAPs are formed opportunistically by one or more local edge devices also referred to as service nodes, such as WiFi access points, femtocell base stations and more resource rich end user devices under the coverage and management of the physical FAP, which can be dedicated fog server, fog-enabled remote radio heads (RRHs) or macrocell base stations. The proposed OF-RAN can be a low latency and high scalable solution for 5G cellular networks.

However, proposed research in OF-RAN is still in its infancy, and there is still much room for improvement in current F-RAN solutions. To achieve the best outcomes, the following research questions in both cloud RAN and F-RAN need to be tackled:

1. How to efficiently utilize the widely avialable local computing resources efficiently in order to provide a low-latency and highly scalable alternative to the current F-RAN and C-RAN approaches?

2. How to ensure an optimal assignment of tasks to nodes with a goal of minimizing energy, delay and maximizing fairness in order to provide complementary solution to the existing F-RAN and C-RAN?

3. How to perform a comparative study of the current F-RAN and C-RAN in terms of delay, energy consumption and failure rate using analytical model in high stressed scenario and provide a suitable solution to support them?
4. How securely can we manage a distributed local computing resources in order to provide an efficient and scalable architecture which can complement the existing F-RAN and Cloud-RAN architectures?

1.2. Contributions

The significant contributions in this thesis for the above-mentioned questions are enumerated as follows:

- In Chapter IV, we propose the opportunistic fog RAN (OF-RAN) inspired by the concept of opportunistic resource utilisation network (oppnet) [8] to address research question **1**. The oppnet is a type of specialised ad hoc network that features opportunistic expansion and opportunistic utilization of local resources gained by the expansion. It is a dynamic form of network comprising originally of a small set of 'seed' nodes, which can be expanded on-demand by recruiting 'helper' nodes in their local areas not employed initially but join the seed nodes in order to fulfill a given task. The FAPs in the current F-RAN and the local edge devices (also referred to as service nodes in our proposed OF-RAN) can be considered to resemble the seed nodes, and helper nodes, respectively, in oppnet [8]. Hence, we propose the concept of a virtual FAPs (v-FAP) formed by two or more local edge devices and monitored by physical FAPs for 5G radio signal processing. Intuitively, by being in close proximity to the user equipment (UE) that generate the processing tasks and harnessing the collective plethora of local computing resources, the proposed OF-RAN could emerge as a low-latency and high-scalability alternative to the current F-RAN and C-RAN approaches. In this chapter we also address the issue of service load balancing in

our proposed OF-RAN with a goal to ensure fairness while distributing service tasks to service nodes.

- In Chapter V, we address research question **2** the issue of assigning the client task to the service nodes of the v-FAP, i.e., task-to-node assignment (TNA), which is a fundamental problem in our proposed OF-RAN architecture while taking into consideration multiple objectives. We formulate and solve the TNA as a multi-objective optimization problem, with the goals of minimizing energy and latency of the v-FAP, while maximizing fairness (or load balancing) amongst its service nodes by minimizing their maximum load.
- In Chapter VI, we have analytically modelled and evaluated the performance of OF-RAN architecture against existing RAN architectures to address research question **3**. We develop an analytical model to evaluate the offloading performance of three RAN architectures: the traditional C-RAN, the existing F-RAN, and the proposed OF-RAN. The performances are analyzed in terms of their energy consumption, completion delay, and failure rate, under the effect of varying scenarios. The simulation result displays that there exists an optimal number of service nodes for which the failure rate of OF-RAN is minimized. OF-RAN also outperforms C-RAN and F-RAN while processing complex tasks for large number of clients. Hence, we understand from this analysis that our OF-RAN architecture is highly scalable and can complement the existing C-RAN and F-RAN.
- In Chapter VII, we address research question **4** by proposing an application paradigm of federated learning using blockchain-enabled OF-RAN, which take advantages of the collaborative processing of OF-RAN and the inherent security through decentralization of blockchain technology. The proposed OF-RAN extends the computation capacity of existing F-RAN by establishing v-FAP which opportunistically recruits resourceful user devices that participate as service nodes. In this chapter, federated deep learning (DL) is modelled and executed at the v-FAP for less resourceful clients such as IoT devices, while

blockchain is employed to ensure the confidentiality and integrity of federated DL and service nodes involved in an OF-RAN. An experimental evaluation using a testbed of Raspberry Pi devices shows the impact of our system on DL application in terms of latency and accuracy. A simulation study is also performed to investigate the impact of blockchain parameters such as block size and block interval on the system throughput and security.

1.3. List of Publications

Journals:

- J. Jijin, B. C. Seet, & P. H. J. Chong, "Performance Analysis of Opportunistic Fog Based Radio Access Networks," *IEEE Access*, vol. *8*, pp 225191-225200, 2020.
- J. Jijin, B. C. Seet, & P. H. J. Chong, "Multi-objective Optimization of Task-to-node Assignment in Opportunistic Fog RAN," *Electronics*, vol. 9, no. 3, 14 pages, 2020.
- J. Jijin, B. C. Seet, & P. H. J. Chong, "Blockchain-enabled Opportunistic Fog Radio Access Network: A Deep Learning Application Case Study," (submitted to a journal)

Conferences:

- J. Jijin, B.-C. Seet, & P. H. J. Chong, "Blockchain enabled opportunistic fog-based radio access network: a position paper", In Proc. *29th International Telecommunication Networks and Applications Conference (ITNAC)*, Auckland, New Zealand, November 2019.
- J. Jijin & B.-C. Seet, "Opportunistic fog computing for 5G radio access networks: A position paper", In Proc. *3rd EAI International Conference on Smart Grid and Innovative Frontiers in Telecommunications (SmartGIFT)*, Auckland, New Zealand, April 2018.
- J. Jijin, B.-C. Seet, P. H. J. Chong, & H. Jarrah, "Service load balancing in fog-based 5G radio access networks", In Proc. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Montreal, Canada, October 2017.

CHAPTER II: Background

This chapter introduces the background concepts relevant to this research, including radio access networks, opportunistic network, multi-objective optimization techniques, blockchain and machine learning technologies.

2.1. RAN

A radio access network (RAN) acts like a middleman connecting the user equipment (UE) to the core network. It helps in assisting network resources across the network. In this section, we will do a background study on the evolution of RAN architecture for 5G mobile systems. The traditional macro base station of 2G mobile network, also referred as base station subsystems (BSS) where all radio and baseband processing are integrated into the base stations (BS) [9]. The traditional BS comprises of a radio equipment controller (REC) and a digital unit (DU). The DU is responsible for functioning such as amplification, modulation, demodulation, analog to digital conversion etc., on the other hand REC is responsible for baseband signal processing as shown in Figure 2.1.

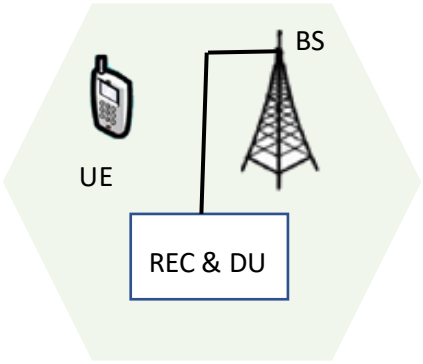

Figure 2.1 Traditional 2G BS

The digital units of traditional BS are separated from each other in universal mobile telecommunication system (UMTS) RAN, also referred as UTRAN. In the distributed RAN

(D-RAN), the radio unit close to the macro-BS are the RRH. Here, the baseband signal processing unit is placed in a convenient, easy, and accessible location which is called BBU. The BBU is responsible for allocating network resources to its corresponding RRH. The RRH is mainly responsible for radio frequency (RF) communication with the end user devices. In this architecture, each BBU is connected to RRH via common protocol radio interface (CPRI) for transmitting the baseband signal. Both optical fiber and microwave links can be used to establish the link between RRH and BBU. Figure 2.2 shows a D-RAN architecture with distributed RRHs and BBUs. The D-RAN architecture is utilized for 3G and 4G networks.

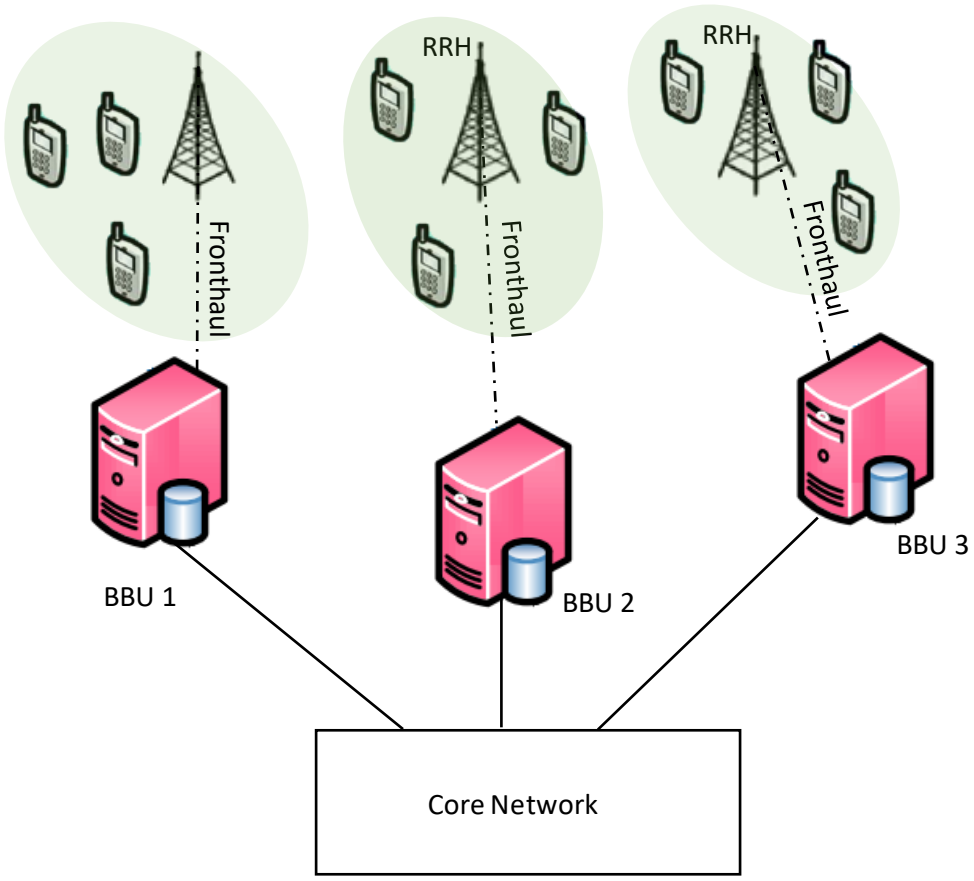

Figure 2.2 D-RAN Architecture

However, as the amount of data traffic increases, in order to maintain quality of service (QoS) requirements, network operators have resorted to centralization and cloudification of BBU.

2.1.1 C-RAN

C-RAN introduces the cloud computing into the RAN architecture and mainly focuses on the centralization and virtualization of BBU processing. The cloud computing paradigm has played a key role in bringing networking, storage, and computing infrastructure close to various applications [10].

The goal of centralization is to improve spectrum efficiency, decrease energy consumption and optimize overall performance of the network. Figure 2.3 illustrates a C-RAN with RRHs, BBU pool and UEs. The main idea of this architecture is to separate all BBUs from the corresponding RRH (unlike the traditional RAN where BBU and RRH are placed together). Every RRH is connected to the BBU pool via the fronthaul link. The BBU pool is connected to the core network via the backhaul links as seen in Figure 2.3.

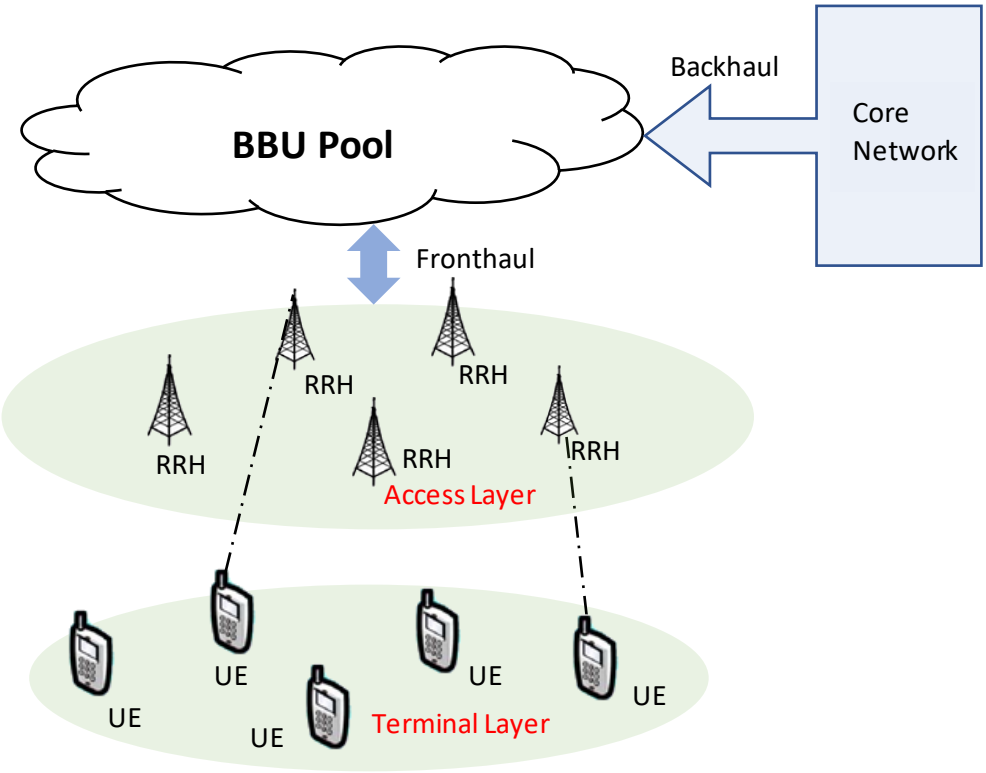

Figure 2.3 C-RAN Architecture

The main aim of virtualizing baseband processing is to reduce the CAPEX and OPEX of 5G RAN. Based on segregating functions between RRH and BBU, C-RAN can be classified into two types: a) Fully Centralized C-RAN; and b) Partially Centralized C-RAN.

a) Fully Centralized C-RAN

   In a fully centralized C-RAN, all functions related to physical layer, transport layer and radio resource control are in the BBU. Some of the benefits of fully centralized C-RAN are better network coverage, network resource sharing and helps in collaborative signal processing among multicell. However, some of the major challenges faced by this architecture is the need for high bandwidth fronthaul to transmit in phase quadrature phase signals between RRH and BBU [11].

b) Partially Centralized C-RAN allows flexible assignment (or splitting) of RAN functionality between the RRH and BBU, depending on the available cloud resources [12,13]. In this scenario, all the physical layer function is performed in RRH and all higher layer functions (e.g., Transport layer, radio resource control) are implemented at the cloud. However, difficulties have been observed for current cloud-based solution to keep its latency within timing requirements of the Long-Term Evolution (LTE) standard [14,15]. More specifically, the overall processing has to be completed in 3 ms in order to comply with the Hybrid Automatic Repeat Request (HARQ) timing, the most critical timing requirement defined in LTE [5]. This tight timing constraint is posing a significant challenge to current cloud-based execution of high-complexity tasks in standard RAN functions such as physical-layer forward error correction (FEC) [16].

Fully and partially centralized C-RANs have been studied from various perspectives to understand their use in 5G radio access network. However, C-RAN may not be a suitable candidate for delay sensitive applications due to its fronthaul constraints, thus F-RAN is proposed.

2.1.2 F-RAN

F-RAN is a promising paradigm for 5G wireless technology capable of providing high spectral and energy efficiency. It is developed by utilizing the concepts of fog computing and can complement the existing C-RAN architecture by providing radio functionalities close to the IoT devices, thereby encouraging both caching and computation offloading.

Fog computing acts a bridge between cloud and IoT devices by allowing features such as networking, computing, storage, processing, and management of data on network nodes also referred as fog nodes within the close vicinity of IoT devices. These fog nodes can also act as a gateway between IoT devices and cloud, for instance pre-processing, compressing data before transmitting it to the cloud [17].

F-RAN comprises of FAPs equipped with both storage and signal processing capabilities. As such, it can perform both collaborative radio signal processing (CRSP) and CRRM [18]. Figure 2.4 illustrates a F-RAN with RRHs, F-APs, BBU pool, UEs, and fog user equipment (F-UEs). The network layer is composed of F-APs and RRHs. The FAPs process and forward information received from the F-UEs. The F-APs are interfaced to the BBU pool and core network via the fronthaul and backhaul link respectively. The F-UEs can be used in relay mode where they communicate with each other via device-to-device communication (D2D).

The F-RAN is classified into two types: distributed F-RAN and centralized F-RAN [19]. In distributed F-RAN, the BBU distributes some of its functionalities to such a computation resource management and storage to the RRHs. On the other hand, centralized F-RAN employs software defined networking (SDN) and network function virtualization (NFV) to support resource management and allocation [19, 20].

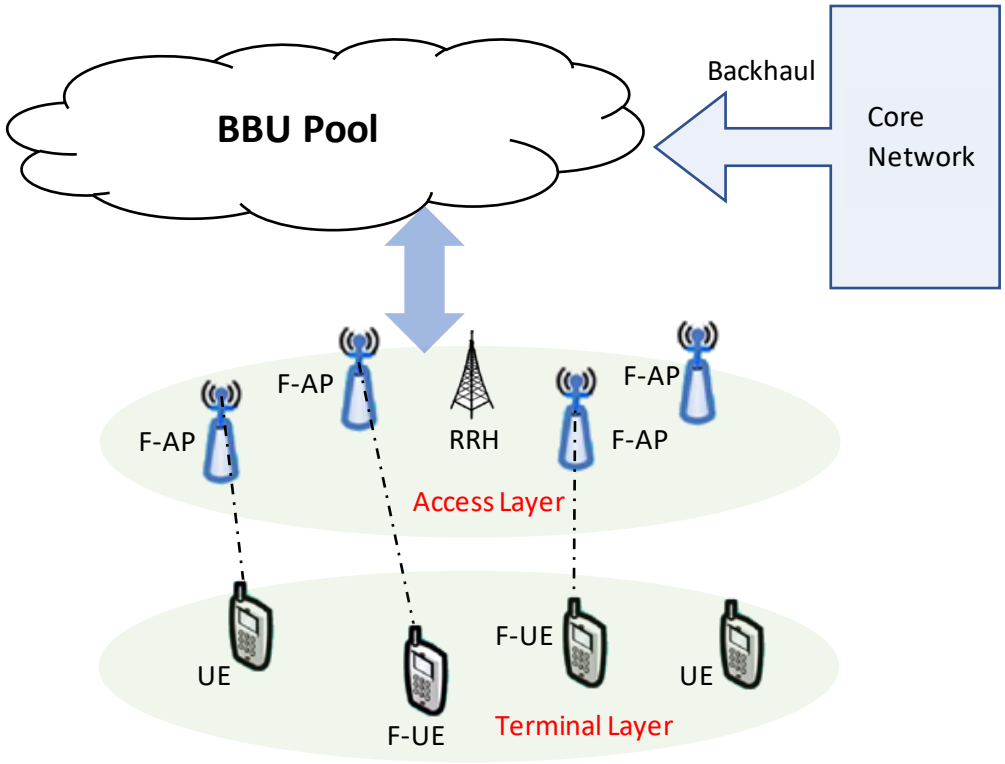

Figure 2.4 F-RAN Architecture

2.2. Oppnet

The opportunistic resource utilization network (oppnet) is a type of specialized ad hoc network that features opportunistic expansion and opportunistic utilisation of local resources gained by the expansion. It is a dynamic form of network comprising originally of a small set of 'seed' nodes, also referred as seed oppnet, which can be expanded on-demand by recruiting 'helper' nodes in their local areas not employed initially, but join the seed nodes in order to fulfil a given task.

Each seed oppnet is formed of a set of nodes which can comprise of a group of nodes or a single node. These initial set of nodes are referred as control center (CC) which is capable of adding or expelling nodes. The nodes which become a part of the seed oppnet are the helper nodes. Along with helper nodes the seed node can recruit Lites which are 'lightweight helper nodes" which are computationally less capable and are equipped with standard oppnet communications, these Lites are mainly sensor nodes that transmit sensed data to the seed node.

Traditional opportunistic network only utilizes class 1 opportunism where the devices within range interact with each other. Oppnet utilizes so-called class 2 opportunism by employing resource expansion and utilization of helper nodes. A detailed study of oppnet is done in [8].

2.3. Multi-objective Optimization Techniques

The multi-objective optimization techniques have become a subject of great interest in research community for solving various multi-objective problems (MOPs) in which multiple objectives are optimized simultaneously subject to set of constraints. However, it is difficult for multiple objectives to obtain their respective optima at the same time, and thus a globally optimal solution satisfying all the objectives may not exist. Nonetheless, a pareto-optimal solution that generates a pareto-optimal outcome or objective vector exists. This solution is also known as pareto frontier (PF) [21].

The PF solutions are a specific set of solutions for which none of the multiple objectives can be improved without giving up the other objectives. This set of PF solutions are referred to as a *non-dominated* solution, which forms the pareto set (PS) of solutions that is mapped to the PF. Different approaches such as scalarization, nature inspired meta-heuristic techniques, multi-objective evolutionary algorithms have been considered for finding the PS of MOPs.

In our work, we have mainly focused on the multi-objective evolutionary algorithms (MOEA), as such we will have a brief overview on this technique. In MOEA, the initial population comprises of randomly generated populations. Based on what objective function need to be minimized, the initial population is ranked based on its non-domination. With respect to the rank, the best candidates are selected for further crossover and mutation to generate the new population. This process is continued until the stop condition is also reached.

In the following section, we will be discussing about two widely used types of MOEA: non-dominated sorting genetic algorithm II (NSGA-II), and multi-objective evolutionary algorithm using decomposition (MOEA/D).

2.3.1 NSGA-II

The NSGA II is based on many layers of classification of the individuals where the non-dominated individuals are selected based on the fitness value a pareto ranking based fitness approach is followed [22]. Based on the selected individuals the child population is generated using crossover operators. However, the main drawback of this technique is that it treats MOP as a "black box" i.e., without using problem specific knowledge. Hence, the incorporation of problem-solving technique in order to perform optimization needs to be considered.

2.3.2. MOEA/D

The MOEA/D technique helps us to tackle the aforementioned issue by decomposing the MOP into many scalar sub-problems that are optimized in parallel by using neighboring information and scalar techniques [23]. For each generation, the population is composed of the best solution found so far for each sub-problem. The neighborhood relation of the sub-problems is established based on the distance between aggregation coefficient vectors. Each sub-problem in MOEA/D is optimized using the neighboring subproblems. A detailed study of MOEA/D is performed in Chapter V.

2.4. Blockchain

Recently, there has been a growing interest in the application of blockchain technology, not only for financial applications but also in healthcare, real estate, IoT and edge computing applications. The key features of blockchain that has enabled it to gain widespread popularity includes lack of need of a central authority, support decentralized applications, security, and smart contracts.

Smart contracts are self-executing scripts that reside on the blockchain and allow distributed and heavily automated workflows [24].

Blockchain technology can be defined as an interconnected chain of blocks which acts as a distributed ledger and is utilized for recording transactions. These transactions are initiated by executing smart contract scripts. The smart contracts that are responsible for execution are stored in block and broadcasted across the blockchain network. The network nodes also referred as miners are responsible for validating the block. These miners approve the block by going through the digital signature and confirming its validity before appending it to a blockchain. The miners create a new block with transaction by solving a puzzle also referred as consensus algorithm such as proof of work (PoW) and this procedure is known as mining. Consensus refers to the mutual agreement between group of miners before adding the block to the blockchain.

A new block in a blockchain comprises of a block identity (ID), parent's block hash and payload data. The block ID which has the hash value of the present block hash and other fields of the payload. Parent's block hash comprises of hash value of previous block which helps in the generation of blocks from the genesis block. Payload data comprises of the digital transactions and information that needs to be announced and spread among network users.

The chain of block is created from the genesis block to the current block. The genesis block is the first block. Each block is linked to the previous block by referencing the parent's block hash. Hash is a function that utilizes a cryptographic algorithm to generate the hash key which helps in protecting the block from tampering. The newly created block can be regarded as a confirmation of the parent block, thereby contributing to maintaining the consensus trust of the publicly distributed database.

2.5. Deep Learning

Deep learning is a neural network based, brain-inspired computing paradigm. Neural network that models a brain neuron comprises of three layers, input layer, hidden layer, and output layer. The deep neural network has multiple hidden layers which map the input layer to the output layer [25]. Each hidden layer are weight vectors and the goal of the DNN layer is to optimize the weight vectors. A dataset is first split into training and testing. The training dataset is used for the optimization of the weigh vectors. The weights are calculated using stochastic gradient descent (SGD) in which the weights w are updated using the learning rate λ, i.e the step size of the gradient descent in each iteration and the partial derivative of the loss function L. The SGD formula is as given below in equation (2.1).

$$w = w - \lambda \frac{\delta L}{\delta w} \tag{2.1}$$

A detailed study of the DNN layer is presented in Chapter VII.

2.6. Chapter Summary

In this chapter, we have performed a background study on the existing RAN architectures, opportunistic networks, multi-objective optimization techniques, blockchain and deep learning. The above-mentioned technologies play a significant role in the motivation and development of our OF-RAN architecture and utilizing it for real world applications. In the following chapter, a detailed literature review of the various challenges faced by the relevant RAN architectures are studied and the research gaps are identified.

CHAPTER III: Literature Review

This chapter performs a critical review of the relevant of the research literature. The review is organised based on five key challenging issues identified, namely: i) limited fronthaul capacity; ii) flexibility and cost of deployment; iii) efficient offloading and task node assignment; iv) large delay and energy consumption; v) security of distributed management.

3.1. Key Challenging Issues

3.1.1. Limited Fronthaul Capacity

In [34], the authors addressed the limited capacity of fronthaul links in C-RAN by proposing an adaptive compression and joint detection scheme at BBU pool, which exploits the correlation among the RRHs to minimize the fronthaul transmission rate while satisfying the QoS requirements. The RRHs are less sophisticated compared to classical base stations, thus they are considered as relaying nodes that forward IQ signals from the UE to the BBU pool. The block error rate (BLER) of the proposed scheme is analyzed in closed form by using pair wise error probability (PEP). Analytical result showed that a compression efficiency of 350% can be achieved by the proposed scheme. However, the authors have assumed that the BBU always have perfect knowledge of all channel information, which may not be practical and can incur significant overheads in their acquisition due to frequent large-scale message exchanges. When it comes to limited capacity of front haul, overhead reduction becomes more important in C-RAN, which is intended to support many users.

In [35], a joint power control and fronthaul rate allocation scheme for uplink communication in an OFDMA based C-RAN is proposed. The proposed scheme is designed for throughput maximization under fronthaul capacity constraint, which is found to have a significant impact on the optimal power control policy. The result showed that the joint design approach achieved better performance than an approach based on optimizing only power control or fronthaul rate

allocation. However, the authors have assumed all fronthaul links to be of equal rates and perfectly lossless, and all mobile users are pre-allocated with the same number of sub-carriers, which made the analysis simple, but may not be realistic in real-world heterogeneous environment.

The authors in [36] studied the joint design of cloud and edge processing for the downlink in F-RAN. The BBU performed joint processing for its enhanced RRHs (eRRHs), which cache frequently requested contents of their users, in addition to being a conventional RRH. The objective was to maximize the minimum delivery rate of requested files under the constraints of limited fronthaul capacity and eRRH power. Two fronthauling modes: hard and soft transfer, with different baseline and pre-fetching strategies are considered. In hard transfer, non-cached files are delivered over the fronthaul links, whereas in soft transfer, fronthaul links conveyed quantized baseband signals as in a cloud-RAN. A simulation performance comparison between hard and soft transfer showed that the latter is more effective in using fronthaul resources except in very low signal-to-noise ratio (SNR) regime. However, the authors have largely adopted a heuristic approach to associating UEs to eRRHs, which may not achieve optimal performance as would be achieved under a more theoretically grounded approach, e.g., by formulating a user association problem which finds the optimal set of eRRHs to associate with the UEs such that the minimum delivery rate is maximized under the constraints of limited fronthaul capacity and eRRH power. Furthermore, the non-associated eRRHs may be put into sleep mode in order to reduce the energy expenditure.

In [27-30], the authors investigated the performance of a C-RAN under flexible centralization, which refers to the concept of suitably proportioning the BBU processing chain (or functional split) between the cloud and RRH, in order to alleviate the issue of limited fronthaul capacity. Various centralization options are analyzed with respect to their required fronthaul capacity, achievable latency and challenges for the signal processing. To enable different information types

beyond raw I/Q samples to be transported over the fronthaul when a functional split is implemented between BBU and RRHs, a packet-based transport approach is proposed in [30]. However, this requires a careful design of the packetization method in order to minimize both header-related overhead and payload-filling latency. It is generally observed that existing Cloud RANs have difficulties in keeping their latencies within the timing requirements of the LTE standard [31,32]. More specifically, the overall processing has to be completed in 3 ms in order to comply with the HARQ timing, which is the most critical timing requirement defined in LTE [28,33].

3.1.2. Achieving flexibility and cost of deployment

In [37], the authors investigated millimeter-wave (mmWave) downlink transmission for the ultra-dense cloud radio access network (UD-CRAN). The fronthaul is shared among RRHs via time division multiple access (TDMA). The joint resource allocation over TDMA based mmWave fronthaul and orthogonal frequency division multiple access (OFDMA) based wireless transmission is studied to maximize the weighted sum rate of all users. The authors have specifically considered a system, where user assigned on any sub-carrier frequency can potentially be served by multiple RRHs subject to fronthaul constraint. The numerical solutions showed that the proposed solution for OFDMA based UD-CRAN can achieve throughput gains of more than 150% over a conventional LTE-A where each user is associated with a single RRH and the mmWave fronthaul bandwidth is equally divided among RRHs. However, the authors have assumed a clear line-of-sight for the mmWave link between RRH and BBU, which may not be possible in densely urban or hilly terrain environments.

The authors in [38] proposed a low-cost approach to network densification through on demand deployment of mobile small cells using either mobile handsets or remote radio units. The mobile small cell base stations transmit RF signals to UE in downlink or forward baseband signals from UE to BBU pool for further processing in the uplink. The simulation result showed

that proposed approach improved throughput and service quality over the coverage of the network. The proposed solution does not require extensive network planning, but there is high potential for inter-cell interferences with the deployment of heterogeneous small cells alongside macro-cells.

The authors in [39] proposed an adaptive algorithm for downlink F-RAN users to select between two content access modes: FAP and D2D, by taking into consideration of their locations, cache sizes and fronthaul delay cost. The proposed algorithm is based on the evolutionary game approach and comprises of three entities: players, strategies, and payoff. Players are users who can choose between multiple access modes. Strategies refer to the selection method, and payoff quantifies the performance satisfaction level of a potential player. Simulation results showed that the proposed scheme can achieve better payoffs than a maximum rate algorithm. However, the authors have not considered the channel conditions between the FAP, F-RAN and D2D users in their proposed user access mode selection.

The authors in [40] proposed a centralized opportunistic access control (COAC) with user access mode selection. They considered a D2D underlaid cellular network composed of both D2D user equipment (DUE) and cellular user equipment (CUE), i.e., DUEs communicate with each other using the same radio resources as the CUEs. The user access mode selection, i.e., for selecting between cellular or D2D mode, is based on the user's signal-to-interference ratio (SIR) with respect to cellular base station and DUEs, and the achievable spectrum efficiency. The COAC scheme is compared with a distributed random access control scheme (DRAC) where sub-channels are allocated randomly. The simulation results showed that the user access mode for COAC performed better than the DRAC scheme. However, little attention has been given to the study of considering fronthaul delay in the user access mode selection and its impact on the network latency performance.

3.1.3. Achieving efficient offloading and task node assignment

The authors in [41] focused on achieving ultra-low latency in F-RAN and proposed an algorithm to determine the optimal number of F-RAN nodes (small- and macro-cell BSs) and amount of resources required for a given distributed computing task. The optimisation problem is firstly formulated to tackle the trade-off between communication and computing resources, followed by cooperative task computing to decide how many F-RAN nodes should be selected with proper resource allocation and computing task assignment. Under the proposed scheme, a target user first sends its processing data to a nearby master F-RAN, which then selects an F-RAN node to serve the user and is responsible for splitting and combining the tasks. Simulation results showed that the proposed scheme can significantly reduce the total service latency and achieve ultra-low latency. However, the authors have not considered the pre-existing computing load of the F-RAN nodes, as well as the load balancing issue, when assigning a new task to them.

The authors in [42] investigated the formation of a femto-cloud (coalition of femtocell access points) for collaborative processing, in order to avoid using remote cloud while enhancing user's quality of experience (QoE). A cooperative game approach to forming the femto-cloud is proposed, such that the available computation resources are maximally exploited while participating femtocell access points are selected based on satisfying the user's quality of experience (QoE) and monetarily rewarded in a fair manner. The femto cell manager (FCM), which is installed and maintained by the network operator coordinates the formation of femto-cloud. The FCM is also responsible for facilitating information exchanging with neighbouring FCMs. The simulation results shows that the execution delay by using femto-cloud can be reduced up to 50% when compared to that by a single femtocell access point. However, very little attention has been given to the load distribution among the femtocell access points.

In [43], the authors proposed an algorithm for selecting small-cell BSs in a small-cell cloud (similar to femto-cloud in [42]) to process offloaded applications from UEs. The algorithm considers both UE's computation demand and the computation capacity and load of small-cell BSs in order to achieve high user QoS while maintaining relatively balanced communication and computation load among small-cell BSs. The simulation results showed that the proposed algorithm can achieve 100% user satisfaction as long as the task offloading rate is within a certain limit. However, the authors have not considered the dynamicity of the network, such as UE mobility and changing available computation capability during the processing of offloaded application.

The placement of decomposable application components onto physical MEC nodes was investigated in [44]. The user application and physical nodes are modelled as graphs whose nodes and edges represent the computation and communication resource entities, respectively. Several algorithms for placing the application to physical graphs in different scenarios are proposed with the aim of balancing the load and minimizing the sum resource utilisation at the physical nodes. However, the existing work is mainly focused on offloading or placement of application computation to base-station type nodes in MEC. On the other hand, our work addresses the RAN task assignment problem to a virtual group of co-located edge devices, including UEs in F-RAN.

In [45], the authors formulated the sensor placement and transmit power assignment in a wireless sensor network (WSN), as a multi-objective optimization problem. They solved the problem with MOEA/D and NSGA-II to obtain an optimal network design, which gives the location and transmit power of sensors that maximize the network coverage and lifetime. The results showed that MOEA/D outperformed NSGA II in terms of both quantity and quality of solutions. However, the authors have only focused on meeting the sensing coverage and lifetime requirements in finding their optimal network designs. They did not consider the connectivity requirements to the sink, which may require some sensors to relay data for others, incurring further energy cost that

can impact their lifetime. In real-world network design, the different roles of nodes and the cost of using them should be considered (as in our work herein).

In [46], the authors analyzed through queuing models the trade-offs between energy consumption, delay and payment cost for mobile devices (MDs) when they performed the computation intensive tasks or offloaded them to the cloud or nearby fog nodes. A multi-objective optimization problem was formulated and solved using the interior-point method to determine the optimal offloading probability and transmission power for each MD with the goals of minimizing energy consumption, delay, and payment cost. The results showed that the proposed scheme achieved lower weighted sum cost of energy, delay and payment than existing schemes.

The authors in [47] focused on minimizing energy and delay in computation offloading from MDs to MEC servers. A method for chromosome encoding was proposed and NSGA-II was used to obtain pareto-optimal offloading decisions. Similarly, a semidefinite relaxation algorithm was presented in [48] to optimize offloading decisions using MD's processor frequency and channel quality with the servers which can impact both energy and delay performances. In these works, dedicated fog or MEC servers were used and other computing resources that may lie close to the MDs were not exploited. Not considering the servers' workload in the offloading decisions may also result in overloaded servers which can become points of bottleneck or failure. Moreover, the solution quality and real-timeliness of the heuristics used are not guaranteed because a huge number of iterations is often required for reaching a satisfying near-optimal solution.

In [49], the authors presented a modified MOEA/D approach to solve an asset-based weapon-target assignment (WTA) problem. Asset refers to some valuable entity to be protected against attacks by some target, and weapons are used to annihilate the targets. The optimization goal is to maximize the asset value and minimize the weapon consumption, using a modified MOEA/D approach. Simulation results showed that the proposed approach performs well in terms of

convergence and efficiency. However, the authors' claim that conventional MOEA/D is inefficient in solving discrete problems does not appear to be well-supported by literature [50]. On the contrary, MOEA/D is known to find optimal solutions within reasonable time for similar discrete optimization problems such as the Traveling Salesman and Knapsack problems.

In [51], the authors presented a fog-based solution that offers computation and storage services to IoT applications. It aims to reduce the network latency of these applications by placing services into fog nodes close to the IoT devices. Multi-objective optimization techniques, including MOEA/D, NSGA II, and Weighted Sum Genetic Algorithm (WSGA) II were used to obtain optimal fog service placements, which minimize network latency, maximize service coverage and the use of available resources of the fog nodes. The results showed that MOEA/D and NSGA II outperformed WSGA II in all three objectives. However, the authors did not address the fairness issue as the workload among the fog nodes is not considered.

3.1.4. Large delay and high energy consumption

The authors in [52] proposed a power model to determine the power consumption and energy efficiency of F-RAN. They also evaluated its latency and compared the results with C-RAN. It was found that F-RAN incurred lower latency, but consumed more power, leading to a lower energy efficiency. However, in the latency analysis, the authors considered the processing time as a constant factor and did not consider the impact of task complexity on both latency and power consumption.

In [53], a cooperative algorithm is proposed to enable cooperation between multiple F-RANs to provide low-latency computing services. The cooperation is coordinated by a master fog node that allocates computation tasks to each of the other cooperating fog nodes, considering their available resources. However, the authors have only evaluated the service latency, and did not consider other aspects such as energy expenditure. It is also unclear how the cooperative F-RAN

may perform against other architectures such as a hybrid cloud-fog RAN. Similarly, the authors in [54] only focused on the latency issue and proposed to minimize the latency of offloading users through a joint optimization of the communication and computation resources.

In [55], the authors analyzed the performance of an F-RAN under different caching strategies and transmission modes. The former refers to different ways of utilizing mobile devices, RRHs, and FAPs to store and deliver popular content to the clients. The latter defines the ways by which the client can access the content, such as from a fog access point (FAP mode), from other user devices via relaying (relay mode), or remotely from the cloud (C-RAN mode). A testbed was also implemented in [56] to demonstrate their F-RAN for video content acquisition. In all of the aforementioned works, however, the authors have only focused on evaluating the caching but not the offloading performance.

In [57], the performance of F-RAN under an opportunistic computation offloading strategy is studied. The strategy utilizes a probabilistic computation offloading model, which determines the likelihood of the wireless channel in use to support the transmission rate required for offloading, leading to three possible processing modes: local mode in which client processes the task by itself; fog mode in which client offloads the task to a FAP; and cloud mode in which client offloads the task to cloud computing center via a FAP. However, the authors have only analyzed the average delay performance, and did not consider factors such as the delay sensitivity of the task in their offloading strategy.

Similarly focused on offloading strategy, but additionally concerned about jamming and interferences from nearby radio devices during offloading, the authors in [58] proposed reinforcement learning based schemes that jointly optimizes the selection of edge device for offloading, offloading rate, and transmit power so that the computational latency and energy consumption are minimized while the offloading signal-to-noise-plus-interference ratio is maximized. The authors analyzed the computation complexity of the proposed schemes and

showed they can reduce computational latency and energy in the presence of jamming and interferences.

The deep reinforcement learning has also been applied to realize network slicing in F-RAN in [59], where computing, caching and radio resources are orchestrated to meet the performance requirements of two different types of services: hot-spot and vehicle- to-infrastructure (V2I). To address the challenges of high cost of data offloading and model training for implementing network intelligence at the edge, an evolved architecture of F-RAN is proposed in [60], which employs federated learning (a.k.a. collaborative learning) to realize intelligent signal processing and network management with less communication overhead and greater efficiency than existing centralized learning paradigms.

The authors in [61] have considered a decentralized asynchronous coded caching scheme for F-RAN to alleviate the burden of the fronthaul link. They have a proposed an encoding set partition method to cater multiple users with different delay sensitive content requirement. Here each FAP server which is the part of the F-RAN architecture requests and store cache content from the cloud based on a placement scheme where each server has its own subfile of the total content. On content request from a client user, each FAP server should be able to recover the cached subfile and transmit it to the user, while exploiting the coded multicasting opportunities and asynchronous and synchronous transmission methods. However, the authors have only considered static content popularity distribution whereas user request content keeps on changing with time. Beside FAP server may have limited memory resources when it comes to large files.

In [62], the edge caching problem in F-RAN is investigated here the FAP in FRAN can cache the content files from the cloud server. The content files can be cached as a single file in each FAP or divided into equal subfiles and transmitted to corresponding FAPs. All FAPs can jointly transmit the files to a client user requesting for content if a single file cached to each FAP or

concurrently transmit the content as multiple coded subfiles. The performance analysis is done for both scenarios. However, this caching technique will not be a suitable candidate when there is a rise in content users. This is because the FAPs have limited storage capability. Hence, we cannot consider F-RAN as a scalable solution.

Delay and energy efficiency are investigated in [63] for F-RAN. The authors have derived delay model for coded, non-partitioned, cached and un-cached files. They have derived an energy efficiency model considering circuit computation, transmission and backhaul links. A multi-objective optimization problem considering both energy and delay in order to achieve the most optimal caching have been formulated. However, this paper has only considered minimizing the delay and maximizing energy efficiency, not taking into consideration of other objectives such as fairness as seen in [91], besides, they have only considered dedicated fog nodes for their architecture as such they cannot scale with the increase in client.

In [64], the authors have focused on task admission where a task of given size, computation requirement and delay tolerance for a given device is offloaded to MEC server while ensuring minimum energy consumption at the MEC server but also ensuring task processing within the delay tolerance of the device. They have formulated an integer programing (IP) model for task admission taking into consideration resource capacity of the offloading devices. The MEC server utilizes this model while considering its own resources to decide if a task need to be offloaded by the device to MEC server or performed locally at the device. However, the authors have considered the management and control of devices by a single fog server, recent literature has considered multiple fog server under the assistance of centralized coordination.

In [65] the authors have considered tackling the computation offloading problem in a fixed fog/cloud system architecture by jointly optimizing the offloading decision and allocation of resources for both fog node and cloud server. The allocated resources include transmit power, computational resource, and radio bandwidth while also considering user fairness and delay

sensitivity of offloaded tasks. Although the authors have considered various tradeoffs while offloading different tasks, but the authors have only considered fixed and dedicated fog devices for the optimization problem this technique will not be scalable when the number of client requests to offload increases.

The authors in [66] have investigated an energy efficient and performance guaranteed task offloading scheme in the context of mobile edge computing. They formulated a new optimization scheme taking into consideration the energy consumption and task completion time for each mobile user when doing the task locally and also while partially offloading the task to an MEC server based on which an optimal decision is made on the amount of task to be performed by mobile user locally and the amount of task to be offloaded to the MEC server. The decision also takes into consideration the server computation capacity and channel bandwidth, while the energy consumption and completion time must satisfy the task requirement. However, the parallel local and cloud execution of task may not be suitable for large number of mobile users as it may lead overload at the server causing failure in task completion.

In [67], the authors have focused on the importance of QoS when offloading deadline driven tasks from less resourceful IoT device to fog nodes which are equipped with computing and storage resources at the network edge. They have focused on efficient fog provisioning i.e., determining which virtual machine (VM) in a fog node need to be rented and how to distribute different tasks to VMs to minimize the system cost and improve reliability. However, in this paper the authors have defined reliability as the completion of offloaded task within a given completion time but not considered the failure which is also possible due to a link connection between the IoT device and fog node which also has a direct effect on the QoS.

3.1.5. Security for distributed management

The authors in [68] attempted to solve a placement problem for mobile edge applications (MEApps) by proposing edge chain, a blockchain based architecture to make mobile edge application placement decision for the mobile hosts (MEHosts) of multiple service providers (SPs). The blockchain uses the logic of the algorithm as a smart contract with the consideration of resources from all mobile edge host participating in the system. The proposed algorithm is designed to provides fairness during resource sharing among multiple SPs. However, the authors have only considered fairness while assigning tasks, but other factors such as energy consumption and latency which are very crucial when considering delay sensitive applications have not been considered.

The authors in [69] mainly focused on edge computing as an enabler to blockchain. They studied edge computing resource management and pricing to support mobile blockchain applications in which mining process of miners, which are mobile phones, can be offloaded to an edge computing service provider. However, blockchain requires more powerful nodes for execution, utilizing the battery-operated mobile devices as miners are not very feasible.

In [70], the authors proposed a blockchain based SDN data chain to record network data to help SDN network manage heterogeneous devices and maintain the interoperability between the control and data plane. They studied various challenges faced by the SDN network and the application of blockchain, but a clear idea on data chain has not been given.

The authors in [71] introduced the concept of software defined blockchain based architecture with SDN controlled fog nodes in order to address some of the issues of traditional network architecture which may not be efficient, secure and resilient enough to serve the diverse number of IoT devices connected to internet and cater low latency applications. This architecture comprises of three layers fog, cloud and device. The device layer comprises of IoT devices data to the blockchain based fog layer. In the fog layer, the fog node refers to distributed fog

computing entities that are connected, aggregated as a single entity using blockchain technique and perform distributed services. The fog layer is responsible for proving data analysis and service delivery in a timely manner to the device layer and ensure offloading computationally intensive tasks to the cloud layer. The blockchain based cloud layer is responsible for low cost, on-demand access to the most competitive computing resources in a secure manner. Although blockchain has been used to enhance data security using PoW and PoS mechanism, not much have been studied about the impact of blockchain parameters such as block size and block interval on the overall performance of the fog layer.

A distributed blockchain technology access control is proposed in [72] for assigning roles and permissions to IoT devices in a distributed manner rather than being controlled by a centralized entity. The system enumerates the following features which includes, mobility, scalability, accessibility which ensures access control rules are available all the time. However, the authors mainly focused on using blockchain for access management of IoT devices that are end users, rather than the management of RAN servicing entitles as in this thesis.

In [73], the authors integrated blockchain and deep learning for IoT to support collaborative DL at the device level, and which provides device integrity and confidentiality in the IoT network. The learning task is performed locally at IoT devices, and the local learning models are aggregated at edge server through blockchain transactions. The implemented system is composed of IoT devices and a powerful edge server acting as a blockchain node for mining blocks and coordinating blockchain transactions. Results show that the system is efficient in terms of accuracy, time delay, and security. However, due to limited resources of IoT devices, the data set used for training its model is small. Thus, the system is not suitable when large or complex data set is involved, particularly for delay sensitive applications.

In [74], a secure federated learning technique called Deepchain is proposed where multiple parties upload their local gradients to a server that updates the model parameters. It uses blockchain cryptographic features to preserve the privacy of local gradients and guarantee the auditability of training process. A prototype of Deepchain is implemented and evaluated in terms of cipher size, throughput, accuracy, and training time. However, using blockchain technology to secure federated learning process can incur a high cost for implementing and maintaining miners and cause a long delay in the information exchange due to the consensus protocol of the blockchain network. In contrast, we do not use blockchain for federated learning in this thesis, but to securely share and maintain information about our RAN servicing entities.

Similarly, in [75], the authors integrated federated learning with blockchain for Internet-of-Vehicles (IoV) with the aim of improving driving experience by enabling collaborative traffic prediction or path selection. A client vehicle can share its data with a multi-access MEC server hosted on a roadside unit (RSU). The MEC server initializes the federated learning process by selecting the participating nodes (i.e., vehicles) based on deep reinforcement learning and distributes the initial weight vectors and client's shared data to the local model of each selected node. Upon training, each local model sends its updated weight to the server which in turn uploads them to the blockchain for verification and aggregation. The global model in the blockchain is then updated and used by the server to make prediction or selection as requested by the client vehicle. Numerical results show that the proposed technique can provide high learning accuracy and fast convergence. However, it is unclear if the delay incurred by this technique is sufficiently low to be feasible for delay-sensitive applications.

The authors in [76] proposed a system of blockchain-assisted federated learning for edge nodes to cooperatively train and predict popular files to be cached for IoT devices. Each edge node trains its local model using local data, and the local gradients are compressed and sent to a cloud server through blockchain to aggregate and update the global model. The updated global

model parameters are then returned to the edge nodes for further training or selecting the files to be cached. The results show that the proposed system improves the cache hit rate and reduces the uploading time required. Although blockchain has been used to enhance data security, not much have been explored about the impact of blockchain parameters such as block size and block interval on the caching efficiency or security.

In [77], a digital twin wireless network (DTWN) model based on blockchain and federated learning is proposed to realize robust edge intelligence. A digital twin is a representation of a physical asset in digital space. Here, the DTWN comprises a collection of end devices digitally represented in the edge servers. The end devices generate running data and synchronize them with their corresponding digital twins. The running state and optimization of the actual end devices can be conducted directly in DTWN. Each edge server aggregates the local models from digital twins of end devices and transmits them to a macro base station for updating a global model via federated learning. This system utilizes permissioned blockchain to record model parameters from digital twins and manages the participating end devices through permission control. Numerical results show that the proposed system reduces learning latency and achieves good learning convergence. However, it is unclear how deviations in the data between actual end devices and their digital twins can impact the resulting edge intelligence.

In [78], the authors proposed a security architecture for IoT networks based on three core technologies: SDN, blockchain, and fog/edge computing. They utilize decentralization in blockchain to secure sharing of data and resources in an IoT network such that it overcomes many-to-one traffic flows and central control dependency. A SDN enabled edge switch continuously monitors the flow of sensor data and traffic traces to the fog nodes. The traffic traces are learned and analysed at the fog controller to identify malicious traffic flows. The authors used deep learning algorithms to detect attacks at network edge. Their evaluation shows

that the proposed architecture performs well in mitigating attacks. However, they did not evaluate its impact on the performance of delay sensitive IoT applications or the QoS of the IoT network.

3.2. Identified Research Gaps

In current literature, centralization of RAN is considered a feasible approach. However, difficulties have been observed for centralized cloud-based solutions to keep its latency within timing requirements of the LTE standard. Based on our literature survey, F-RAN seems to be a suitable candidate to address the issues related to cloud RAN. However, there is still much room for improvement in current F-RAN solutions. To achieve the best outcomes, the following research gaps in both cloud RAN and F-RAN need to be tackled:

1. The capacity limited fronthaul is one of the main challenges faced by current C-RAN. Although many compression techniques and flexible centralization have been introduced, this requires RRHs to be more sophisticated and capable of handling exponential traffic growth. Similarly, the excessive load concentration in BBU pool is another issue, as the devices depending on BBU pool for processing are not limited to cellular devices, but also the massive number of IoT devices.
2. An efficient scalable solution which can leverage nearby resource such as sophisticated end user devices is still underway.
3. Another significant problem is the optimal assignment of resource intensive tasks on edge servers and F-RAN nodes, while not considering the minimization of energy consumption and overall computation delay along with no node failure due to computation tasks overload.
4. To perform analytical modelling of current F-RAN and C-RAN in terms of delay, energy consumption and failure rate, to understand its performance in high stressed scenario and provide suitable solution which can complement the existing RAN architecture.

5. We still need to utilize a secure, scalable, and easily manageable comprehensive solution when involving edge servers and F-RAN nodes for computation of resource intensive task from client devices. This can be done by inculcating existing techniques such as blockchain which have become very popular in the IoT network.

Although current F-RAN ia a suitable candidate to solve the above-mentioned reserarch gap but it mainly comprises of dedicated fog servers, fog-enabled RRHs or macrocell base stations as FAPs. Very little attention has been given to the utilisation of a large number of other distributed edge devices such as WiFi access points, femtocell base stations, and resource-rich user devices that can be incentivised to lease their resources and collaboratively serve as 'service' nodes to other end-users. Hence, the present F-RAN may not be an efficient and scalable solution.

3.3. Chapter Summary

This chapter has made a detailed literature study on the existing challenges and issues based on which we have enumerated the research gaps. To achieve a solution that can efficiently tackle these challenges and issues, while not replacing but complement existing RAN solutions, this thesis has proposed a new scalable RAN architecture, utilized multi-objective optimization techniques for efficient task-node assignment, analytically modelled the proposed architecture, and harnessed blockchain technology for secured management of our RAN servicing entitles, which we will elaborate over the next four chapters.

CHAPTER IV: Proposed OF-RAN

4.1. Introduction

The 5th Generation (5G) and beyond cellular networks will not only focus on providing high-speed and reliable human communication services, but also support communications between billons of smart objects or 'things' in the coming era of the IoT [1]. To achieve these goals, centralized radio access networks was introduced where data received by base BSs are transmitted over fibre links to a central unit for processing on specialized hardware [2]. Recently, the concept of C-RAN was proposed to replace the specialized hardware in centralized RAN with commodity cloud-computing platform to allow for more flexible splitting and allocation of RAN functionalities between RAPs and cloud, depending on the available cloud resources. However, C-RAN has following challenges [3, 4]: i) constrained backhaul capacity; ii) load concentration on centralized base band unit (BBU) pool; and iii) ultra-low-latency requirements of 5G.

F-RAN is a promising candidate to tackle the aforementioned challenges by harnessing distributed resources of collaborative edge devices to deliver localized RAN services to end-users [5]. Its predominant philosophy is to make full use of local radio signal processing, CRRM and distributed storage capability in edge devices [6]. Through ingestion and processing of end-user tasks close to their sources, F-RAN has potential to meet the stringent latency and bandwidth requirements of 5G services and applications [7].

The F-RAN comprises mainly of geo-distributed fog access points (FAP) that may serve end-users directly or through other supporting edge devices acting as service nodes [7]. FAPs can be implemented as dedicated fog servers, or fog enabled RRHs or macrocell base stations.

However, the current F-RAN does not efficiently utilize a large number of other distributed edge devices in the proximity of FAPs such as WiFi access points, femtocell base stations, and resource-

rich end-user devices that can be incentivised to lease their resources and collaboratively serve as ‘service’ nodes to other end-users.

Here we introduce the concept of OF-RAN which comprises of v-FAPs formed by resource-rich end-user devices such as tablets and the high-end smartphones that can be incentivized to lease their processing resources and collaboratively serve other end-users under the coverage and management of the FAPs. This is inspired by the concept of oppnet which is a type of specialised ad hoc network that features opportunistic expansion and opportunistic utilization of local resources gained by the expansion. It consists of a small set of ‘seed’ nodes, which can be expanded on-demand by selecting ‘helper’ nodes in their local areas that are not employed initially but may join the seed nodes to fulfill a given task. The FAPs in the current F-RAN and the local edge devices (also referred to as service nodes in our proposed OF-RAN can be considered to resemble the seed nodes, and helper nodes, respectively, in oppnet [8]. Figure 4.1 shows the OF-RAN architecture with the proposed v-FAPs at the access layer.

As shown in Figure 4.1, end-users (referred to as client nodes hereinafter) requests for help from its nearby RRH, which in turn dynamically forms a v-FAP from a set of service nodes in the client's locality. The seed node in the v-FAP decides the set of service nodes to serve the client and the workload assignment to each of the service nodes based on their resource availability. We consider heterogeneous service nodes having different resource capacities (thus different costs of utilizing the resources). Three resource types: computation, storage, and communication resources, can be considered when assigning tasks from client node to each service node.

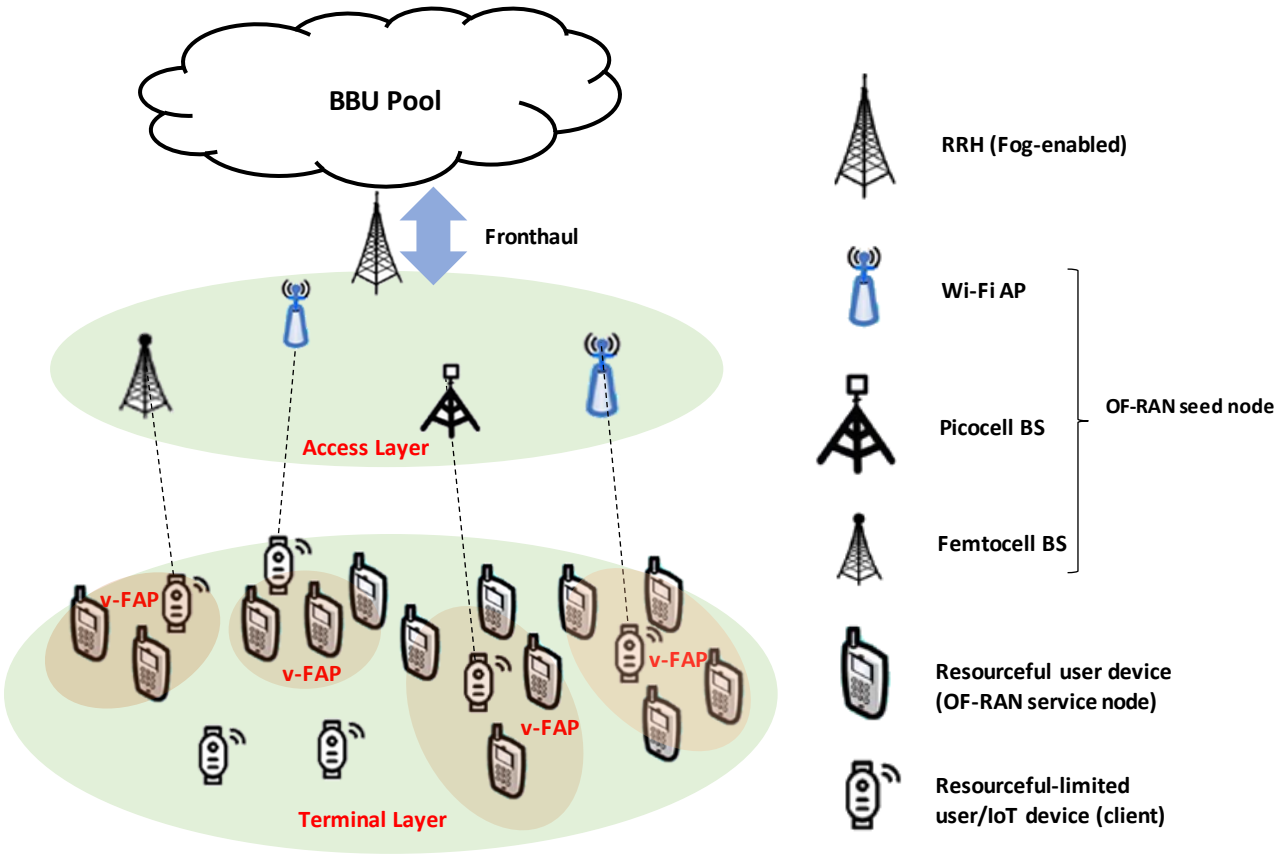

Figure 4.1 OF-RAN Architecture

Although OF-RAN is promising, its performance may be limited by the service nodes that constitute the v-FAPs, which are less resourceful compared to the FAPs or the core cloud. Overloaded service nodes may become single points of failure in the system. Therefore, an important issue is deciding which service node(s) should be assigned to process what tasks from end-users in the F-RAN. The tasks here refer to user processing tasks that would have normally performed by the FAP or remotely by the cloud.

This chapter focuses on addressing some of the existing RAN issues by proposing OF-RAN. We also focus on solving the problem of service load balancing by formulating a single-objective task assignment problem by first modelling user tasks as a task graph and service nodes as an edgeless service graph. In this model, the OF-RAN seed node is responsible for mapping and maintaining the task graph and service graph. We then formulate an optimization problem to find the optimal

assignment of tasks with objective of balancing loads at service nodes. The rest of the chapter is organized as follows. Section 4.2 formulates the problem and presents the proposed service load balancing scheme. Numerical results are discussed in Section 4.3. Finally, Section 4.4 concludes the chapter.

4.2. Problem Formulation of Proposed Scheme

4.2.1. Definitions

Task graph: The task to be processed is abstracted as a graph in which nodes represent the tasks and edges between the nodes represent the relationships, e.g., execution sequence, of the tasks. Each node $m \in M$ in the task graph is associated with parameter $r_{m,k}$ that represents the demand for each resource type $k \in K$ by task m. The types of resources may include but not limited to computation, storage, and communication resources.

Service graph: The service nodes that may be selected to serve a client are also represented as nodes in an edgeless graph. Each node $n \in N$ in the graph is associated with parameter $u_{n,k}$ that represents the unit cost of using k-type resource of n^{th} service node for a given task, and is defined to be inversely proportional to the resource capacity, i.e. smaller the capacity of a resource, higher the cost of its use, and vice-versa.

Mapping: A mapping defines a specific pattern by which a client's tasks are assigned to the service nodes. Figure 4.2 shows an example of mapping tasks from a client to available service nodes.

4.2.2. Formulations

For a particular client, the cost of using a j^{th} mapping from a set of all possible mappings π for assigning M tasks to N service nodes, each with K types of resources needed for task execution, is given by:

$$c_j = \sum_{n=1}^{N} v_{j,n} \tag{4.1}$$

where $v_{j,n} = \sum_{k=1}^{K} w_{n,k}$ is the cost of using n[th] service node in j^{th} mapping, $w_{n,k} = u_{n,k}\ D_{n,k}$ is the cost of using resource k in n, and $D_{n,k}$ is the total demand for resource k by a set of tasks $M_n \subseteq M$ assigned to n. Denote $R_{n,k} = \{r_{1,k},..,r_{|Mn|,k}\}$ as the set of demands for resource k by the tasks in M_n. Then $D_{n,k} = \sum R_{n,k}$ if M_n is non-empty, or zero otherwise.

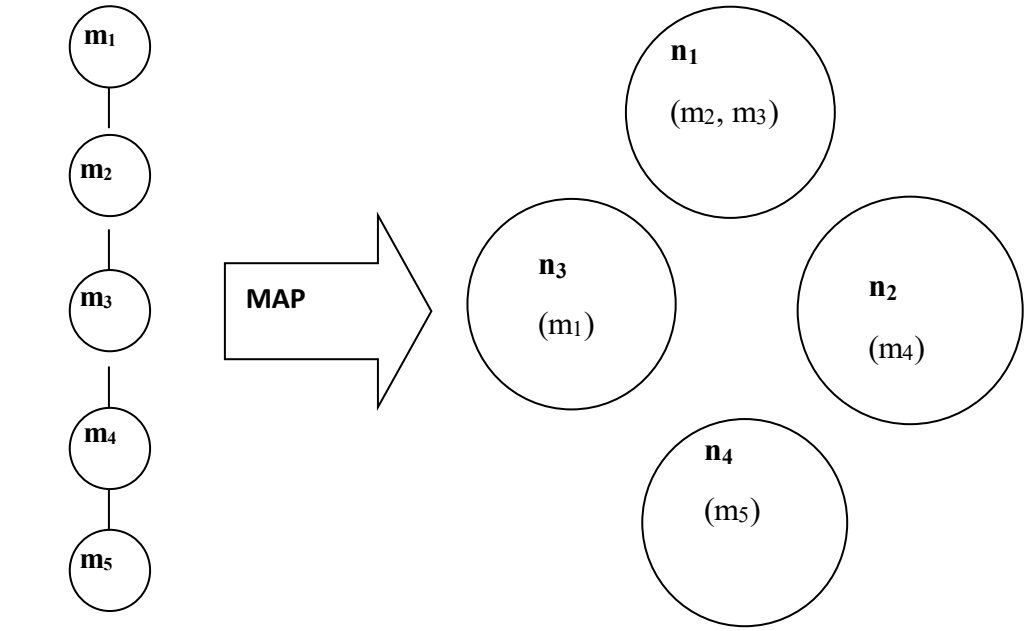

Figure 4.2 Example of mapping tasks to service nodes with M=5 and N=4

Objective function: The optimization objective function of the proposed algorithm is defined as:

$$\min_{1..J} \max_{1..N} \{v_{j,n}\} \tag{4.2}$$

where J is cardinality of set π. Equation (4.2) determines the mapping which has the lowest maximum cost service node.

On the other hand, the objective function of a conventional greedy scheme that simply determines the mapping which has the minimum total cost is defined as:

$$\min_{1..J}\{c_{i,j}\} \tag{4.3}$$

The pseudo-code of the proposed and greedy schemes are shown in Algorithm 4.1, and Algorithm 4.2, respectively. The proposed algorithm not only considers the cost of resource utilization but also load balancing whereas the greedy algorithm emphasizes mainly on minimizing the overall cost of resource utilization. Both the proposed and greedy algorithms require an optimization problem to be solved as a subroutine, for each client's tasks assignment.

Algorithm 4.1. Proposed scheme

1. **for** $j = 1\ to\ J$ **do** // for each mapping
2. **for** $n = 1\ to\ N$ **do** // for each service node in a mapping
3. **for** $k = 1\ to\ K$ **do** // for each resource in a service node
4. $D_{n,k} = \sum R_{n,k}$ // find total demand for resource k in n^{th} service node
5. $w_{n,k} = u_{n,k}\, D_{n,k}$ // find cost of using resource k in n^{th} service node
6. **end for**
7. $v_{j,n} = \sum_{k=1}^{K} w_{n,k}$ // find cost of using n^{th} service node in j^{th} mapping
8. **end for**
9. **end for**
10. mapping $\leftarrow \min_{1..J} \max_{1..N}\{v_{j,n}\}$ // select mapping which has the lowest
11. **return** mapping // maximum cost service node

Algorithm 4.2. Greedy scheme

1. **for** $j = 1\ to\ J$ **do** // for each mapping
2. **for** $n = 1\ to\ N$ **do** // for each service node in a mapping
3. **for** $k = 1\ to\ K$ **do** // for each resource in a service node
4. $D_{n,k} = \sum R_{n,k}$ // find total demand for resource k in n^{th} service node
5. $w_{n,k} = u_{n,k}\, D_{n,k}$ // find cost of using resource k in n^{th} service node
6. **end for**
7. $v_{j,n} = \sum_{k=1}^{K} w_{n,k}$ // find cost of using n^{th} service node in j^{th} mapping
8. **end for**
9. $c_j = \sum_{n=1}^{N} v_{j,n}$ // find total cost of using j^{th} mapping
10. **end for**
11. mapping $\leftarrow \min_{1..J}\{c_{i,j}\}$ // select mapping which has the minimum total cost
12. **return** mapping

4.3. Simulation Environment

In this chapter, we focus on balancing service loads in F-RAN by enhancing the overall resource utilization of the service nodes through more effective task assignments. Thus, a key performance metric is the standard deviation (SD) of the normalized load of service nodes. We expressed the normalized load of a service node as a percentage of its total resource capacity consumed for processing its assigned tasks.

We implemented the proposed and greedy schemes in MATLAB and evaluated their performance under varying number of service nodes and service tasks. Table 4.1 summarizes the simulation parameters and values used.

Table 4.1 Simulation Parameters

Parameter	Value(s)
Number of service nodes (N)	1, 3, 5, 7, 9
Number of service tasks (M)	2, 4, 6, 8, 10
Number of resource types (K)	3
Number of mappings (J)	$\frac{(N+M-1)!}{M!\,(N-1)!}$
Number of Simulation runs	100

4.4. Results and Discussions

We have considered a heterogeneous real-world like scenario where each service node has different available resource capacity, and each client task has different resource demand from the service nodes. We simulated these heterogeneous quantities as random numbers drawn from a bounded uniform distribution.

We further considered two cases. In the first case, the FAP (e.g., fog-enabled RRH) which manages the v-FAPs under its coverage is assumed to have full knowledge of the resource capacity of individual service nodes that constitute each v-FAP. Thus, the FAP can ensure that only a service node whose resource capacity meets the resource demand of a task will be assigned with the task. However, this may require non-trivial amount of information exchanges between the FAP and service nodes, which will only increase with the number of service nodes and number of resource types considered. Therefore, we investigate a second case where FAP does not have such information and evaluate the potential for *mapping failure*, which we defined as an event when one or more service nodes of a mapping chosen by the proposed or greedy algorithm based on cost considerations are unable to perform their assigned tasks due to insufficient resources.

Firstly, we analyze the results of the first case. Figure 4.3 shows the impact of the number of service nodes N in a v-FAP on the SD of both schemes under two service loads: M=3 and M=7 tasks. The result shows that the SD of the proposed scheme is consistently lower than that of

greedy scheme, which reflects a better load balanced performance. This can be attributed to the proposed scheme using mappings that have the lowest maximum cost service node, which tends to spread the total load across more service nodes. It is also observed that for both schemes, the SD reduces as *N* increases. This is expected as more service nodes are available to share the total processing load, which in turn reduces the load on any particular service node. Furthermore, as *M* increases from 3 to 7, the SD of the greedy scheme increases more significantly than the proposed scheme. This is because the greedy scheme always prefers to use lower cost nodes, which are loaded more heavily as the total processing load increases.

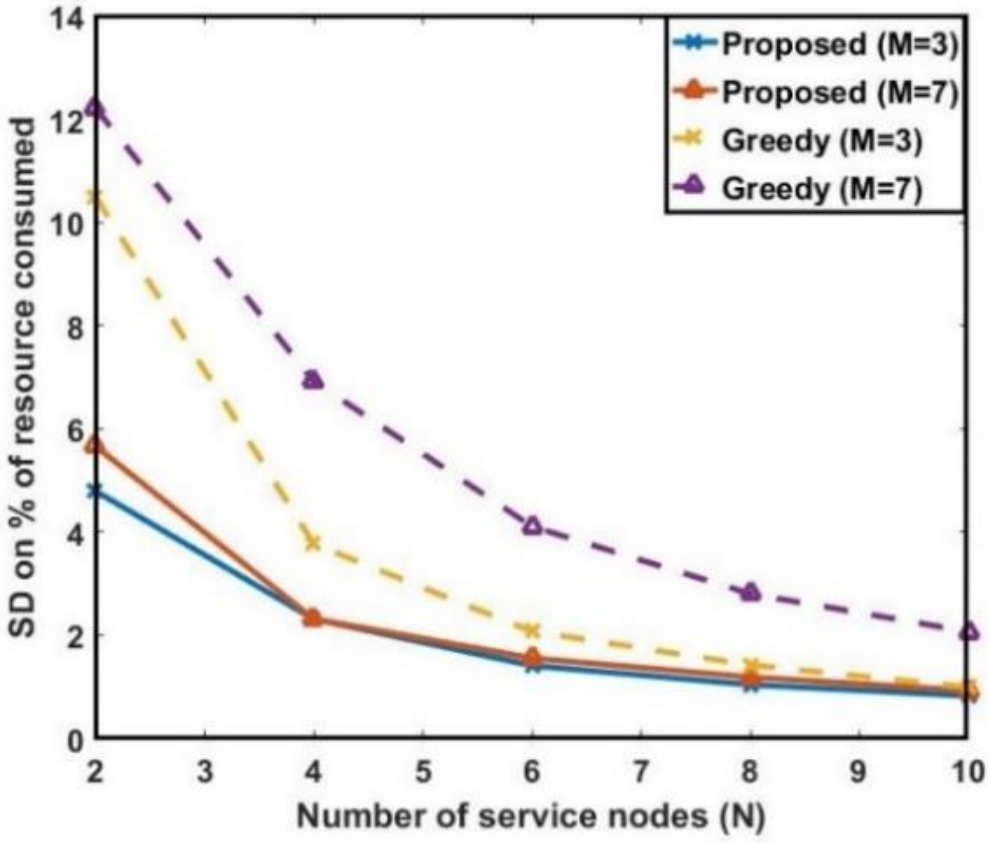

Figure 4.3 Impact of service nodes on standard deviation of resource consumed with full knowledge of service node resource capacity.

Figure 4.4 shows the impact of the number of service tasks *M* from client on the SD of both schemes for a given number of service nodes: *N*=4, and *N*=8. The result shows that the SD of

both schemes increase with *M*, but the rate of increase of the proposed scheme is much slower due to its more even spreading of the load among the service nodes. Similar to the result in Figure 4.3, it is also observed that the performance of both schemes are better when *N* is increased from 4 to 8.

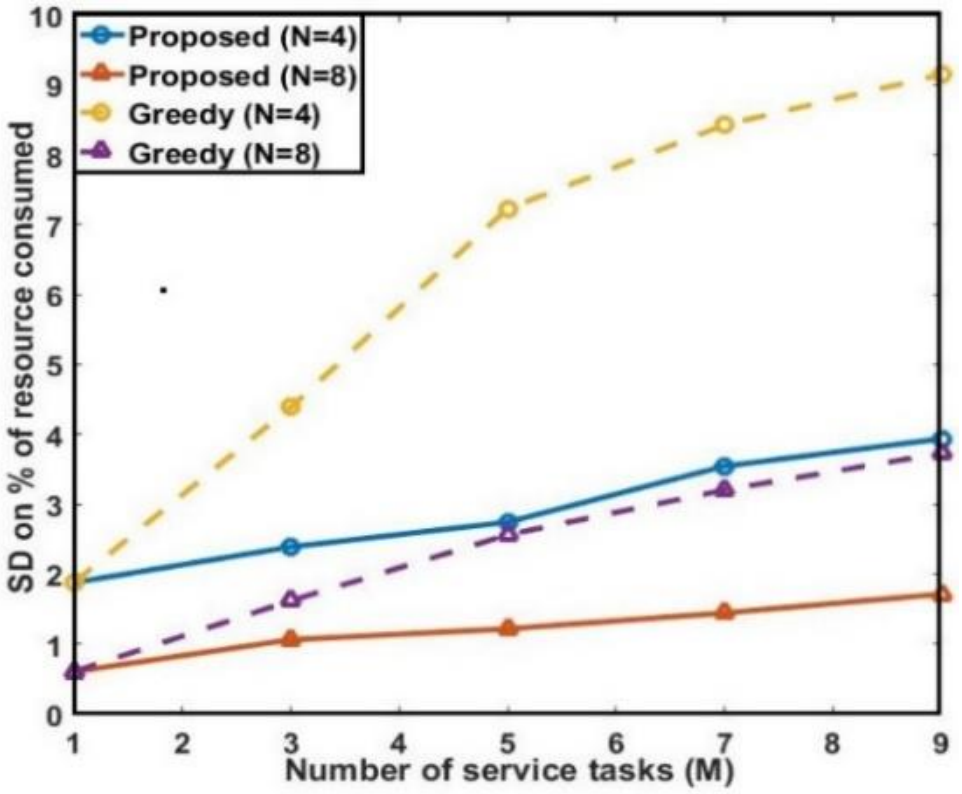

Figure 4.4 Impact of service tasks on standard deviation of resource consumed with full knowledge of service node resource capacity.

Next, we analyze the results of the second case. As in Figure 4.3, Figure 4.5 shows the impact of *N* on the SD of both schemes for *M*=3 and *M*=7, but under the scenario that FAP has no knowledge about the resource capacity of individual service nodes, which may be more practical for implementation. The result shows that in comparison with the greedy scheme, the SD of the proposed scheme increases only marginally even when the number of service tasks is increased from 3 to 7, which reflects its resiliency in maintaining a load balanced performance. The other performance trends remain similar to those in Figure 4.3.

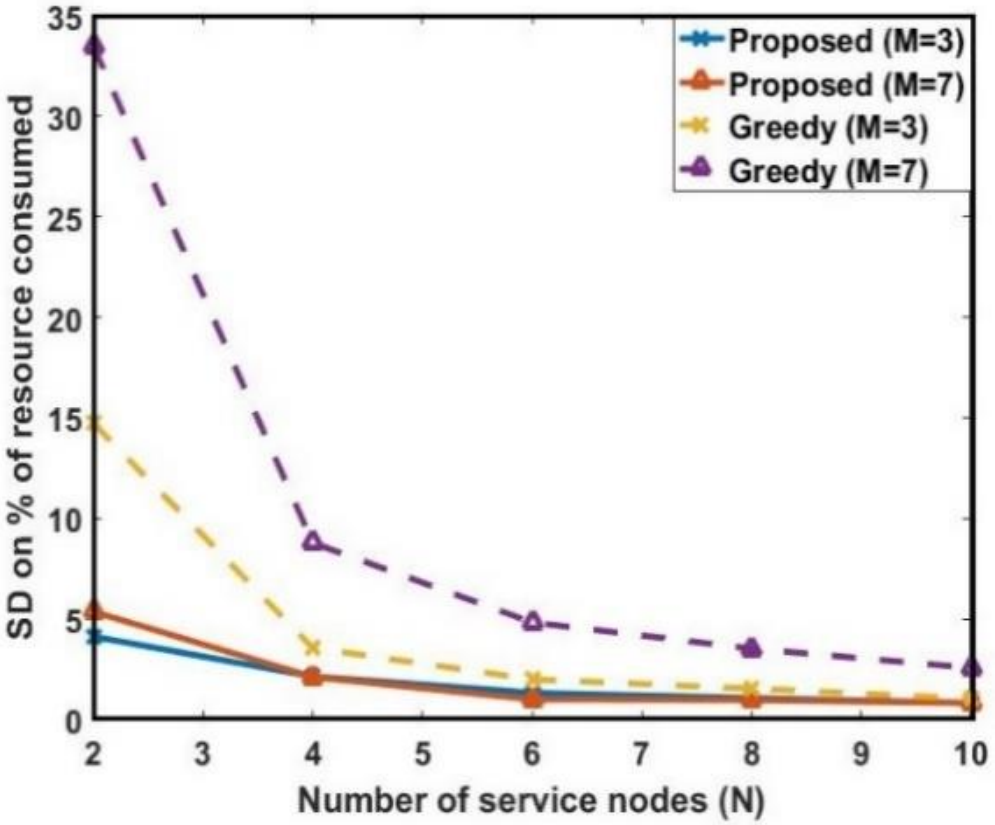

Figure 4.5 Impact of service nodes on standard deviation of resource consumed with no knowledge of service node resource capacity.

Similarly, Figure 4.6 shows the impact of *M* on the SD of both schemes given *N*=4, and *N*=8. Generally, there is no significant difference in performance of the proposed scheme, while SD for greedy scheme is higher than that in Fig. 4, particularly with low number of service nodes, e.g. *N*=4. The other performance trends remain similar to those in Figure 4.4.

Table 4.2 shows a comparison of mapping failure rates between the proposed and greedy schemes. As defined earlier, the mapping chosen by a scheme fails when the resource demand of a task exceeds the resource capacity of the service node assigned by the mapping to process the task.

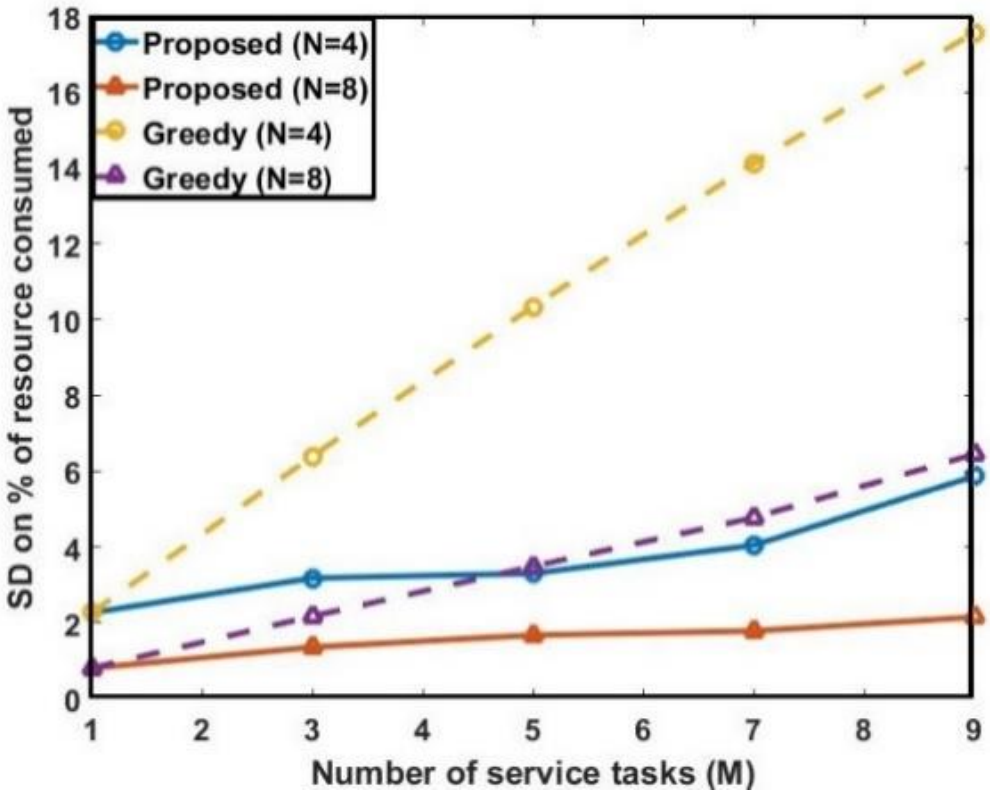

Figure 4.6 Impact of service tasks on standard deviation of resource consumed with no knowledge of service node resource capacity.

Table 4.2 Failure Rate Comparison

Scheme	Mapping Failure Rate (%)					
	No. of Service Nodes (*N*) (with *M*=5)			No. of Service Tasks (*M*) (with *N*=6)		
	N=2	*N*=6	*N* =8	*M*=1	*M*=5	*M*=9
Proposed	2	0	0	0	0	6
Greedy	56	41	39	0	41	93

The result shows that the failure rate of the proposed scheme is consistently lower than the greedy scheme for all considered scenarios (comprising of 5 different combinations of *N* and *M*). The proposed and greedy scheme suffers a failure rate of up to 6%, and 93%, respectively. For both schemes, the trends of the failure rate are consistent with those of SD in Figures 4.5 and 4.6, i.e.,

the high failure rate occurs under the same circumstances (defined by *N* and *M*) where SD is high, and vice-versa. The low failure rate of the proposed scheme is encouraging, particularly given the fact that it was achieved without the need for FAP to know the resource capacity of individual service nodes.

Finally, Table 4.3 compares both the failure rate and SD between the proposed and greedy schemes as the number of service tasks (*M*) is further increased for a given number of service nodes. The results show that the failure rate of the greedy scheme reaches 100% when the number of service tasks increases to 15, while it takes a larger load of 25 service tasks for the proposed scheme to suffer a similar failure. The proposed scheme also widens its SD performance gap over its greedy counterpart, demonstrating its better scalability.

Table 4.3 Failure rate and standard deviation comparison

M with *N*=6	Mapping Failure Rate (%)		Standard deviation (%)	
	Proposed	Greedy	Proposed	Greedy
15	34	100	4.11	16.89
20	76	100	4.74	22.21
25	99	100	6.21	27.99

4.5. Chapter Summary

In this chapter, the concept of OF-RAN has been proposed and one if its key problems of service task assignment with load balancing has been studied. We have formulated an optimization problem for mapping a linear task graph to an edgeless service graph with the objective function to minimize the maximum resource costs. A service load balancing algorithm for the v-FAPs is then proposed. To our knowledge, no similar research has been done for F-RAN with v-FAPs. Numerical results show that the proposed scheme can achieve significantly more balanced loads amongst service nodes and lower failure rates due to overloading as compared to a greedy minimum cost scheme.

CHAPTER V: Multi-objective Optimization of Task Node Assignment Optimization in OF-RAN

5.1. Introduction

With the emergence of IoT where a large number of smart devices will need to communicate with each other, the next-generation cellular networks shall play a pivotal role in providing fast and reliable communication services, not only to humans but also the IoT devices [79].

The notion of C-RAN was developed to support these goals, by enabling flexible splitting of RAN functionalities between RAPs and the cloud, based on the availability of cloud resources [80]. However, the C-RAN may suffer from a heavy workload on its centralized BBU pool, coerced backhaul capacity, and difficulty in meeting the low latency requirements of delay-sensitive applications [81]. In order to address these challenges, the F-RAN was proposed. Unlike Cloud-RAN, the F-RAN utilizes resources from devices on the network edge to provide localized RAN services to its end-users [82]. The F-RAN deploys geo-distributed fog access points (FAP), which serve the user devices directly or with the assistance of other devices. The F-RANs, however, have only considered using dedicated fog servers, existing RRHs or macrocell base stations extended with fog functionalities as FAPs. They do not leverage on the presence of other edge devices such as WiFi access points, femtocell base stations, and resourceful user devices such as high-end smart phones, tablets, and smart TVs, to collectively perform the role of the FAP [83].

Recently, we have proposed the idea of OF-RAN [26], conceived from the concepts of F-RAN and oppnet. The latter is a type of mission-oriented ad-hoc network formed to utilize local resources opportunistically [8]. It composes a small set of 'seed' nodes which, when required to accomplish a mission, can recruit locally available 'helper' nodes to establish an oppnet for that mission. The FAPs of the current F-RAN resemble the seed nodes of our proposed OF-

RAN architecture. Each seed node can recruit local resourceful user devices as helper nodes, which it manages collectively as a v-FAP. Each v-FAP can be formed dynamically by the seed node to serve a resource-limited user or IoT device. In this chapter, these resource-limited devices, and helper nodes, are referred to as the clients, and service nodes, respectively. The processing to be done by a v-FAP for a client is referred to as a service task, which previously would have been performed by the dedicated FAPs in F-RAN or by the cloud in C-RAN.

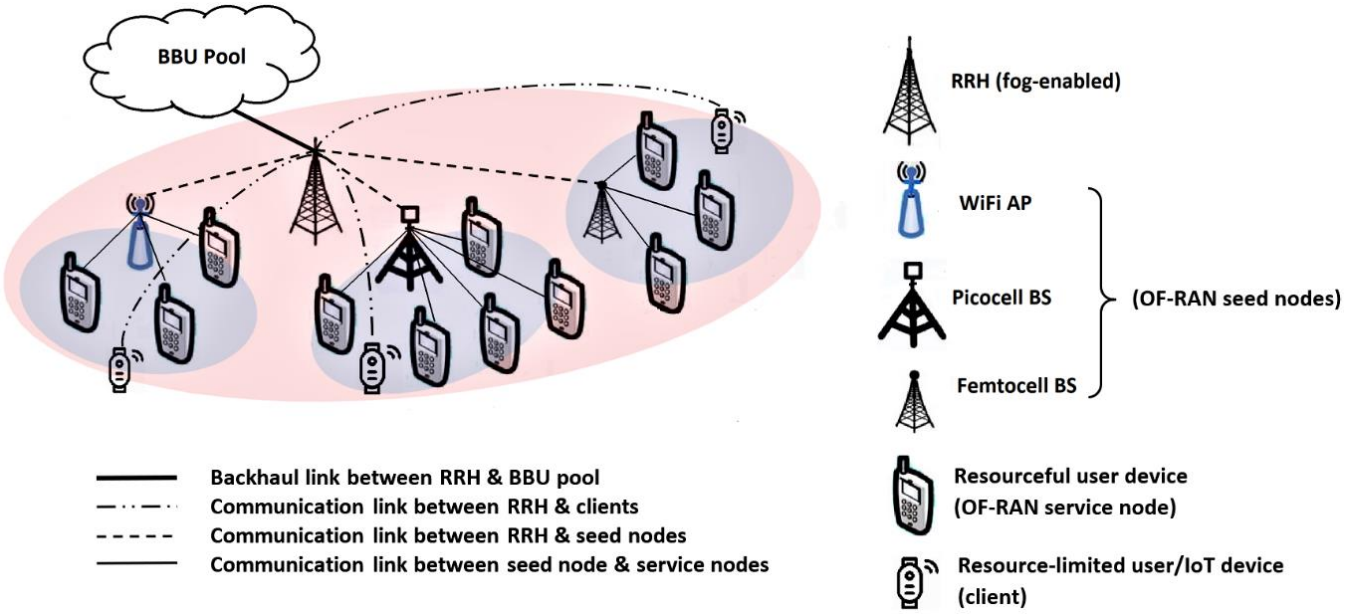

Figure 5.1 OF-RAN architecture for TNA

Figure 5.1 illustrates an OF-RAN with three v-FAPs (shaded in blue), each serving a client through a seed node and multiple service nodes. Initially, the client sends a service request to the fog-enabled RRH (as in current F-RAN). If the RRH is too busy to serve the client, it can instruct a local seed node to serve on its behalf. If the seed node does not have sufficient resources to serve the client, the seed node can form a v-FAP by recruiting local resourceful user devices as service nodes. Considering real-world heterogeneity in the service node's resource capacities and client task's resource demands, a significant problem that arises in OF-

RAN is the proper assignment of the client task to service nodes in a v-FAP, i.e., TNA, which best meets the desired performance objectives.

In one of our previous works [84], a similar TNA problem is solved as a single-objective optimization problem with the fairness objective of balancing the workloads at the service nodes. However, in the real-world, there are often multiple objectives to be considered when finding the best possible solution. Hence, the main contributions of this chapter are as follows: we reconsider the TNA problem in our OF-RAN and present a novel solution approach that balances the trade-offs among conflicting objectives. More specifically, we formulate and solve the TNA problem as a tri-objective optimization problem that seeks to minimize energy and latency of the v-FAP, while maximizing fairness amongst its service nodes by balancing their resources consumption.

The rest of the chapter is organized as follows. Section 5.2 presents the system model and formulates the problem. Section 5.3 presents the designed optimization framework for our specific TNA problem. The simulation environment and results are discussed in Section 5.4, and Section 5.5, respectively. Finally, Section 5.6 concludes the chapter.

5.2. System Model and Problem Formulations

This section presents the v-FAP system model for the proposed OF-RAN and formulates the TNA problem as a tri-objective optimization problem.

5.2.1. System Modelling

Figure 2 shows an OF-RAN seed node lies in the coverage of a fog-enabled RRH and is surrounded by resourceful user devices that it can recruit as service nodes to form a v-FAP (shaded in blue) for a resource-limited user or IoT client device with computation-intensive and delay-sensitive tasks to be performed. The communication links between the seed node

and RRH can be wired or wireless, while only wireless links exist between the seed node and service nodes, between the seed node and client, and between the client and RRH. As the RRH coordinates the transmission and reception between various nodes involved in the v-FAP formation, we consider the communication environment as an interference-limited environment.

As shown in Figure 5.2, a client wishing to be served firstly sends a request (1) including its task requirements to its associated RRH. If the RRH is busy, it sends a notification (2) to a seed node located near the client to serve on its behalf, which is also received by the client. The client then sends its tasks to the seed node (3). Based on the received tasks and task requirements, the seed node decides if it can serve the client on its own or should enlist the resources of its neighboring user devices by recruiting them as service nodes to collectively serve in a v-FAP. In the latter case, the seed node determines an optimal way of assigning the client's tasks to the service nodes so that the performance objectives are achieved. An optimal assignment defines the set of service nodes that constitute the v-FAP and the task(s) that each service node will perform. Based on the optimal assignment, the seed node sends the received tasks to the service nodes for processing (4), collates the processed tasks from the service nodes (5), and forwards them to the client (6).

We consider a seed node with N resourceful user devices in its neighborhood that can be recruited as service nodes. We also consider a client with M tasks to be performed externally. To reflect real-world scenarios, both the service nodes and tasks are assumed to be heterogeneous, each having different resource capacities, and resource demands, respectively. Two resource types are considered: energy resource and time resource, which refer to the amount of energy available, and time available, respectively, in a service node to serve the client.

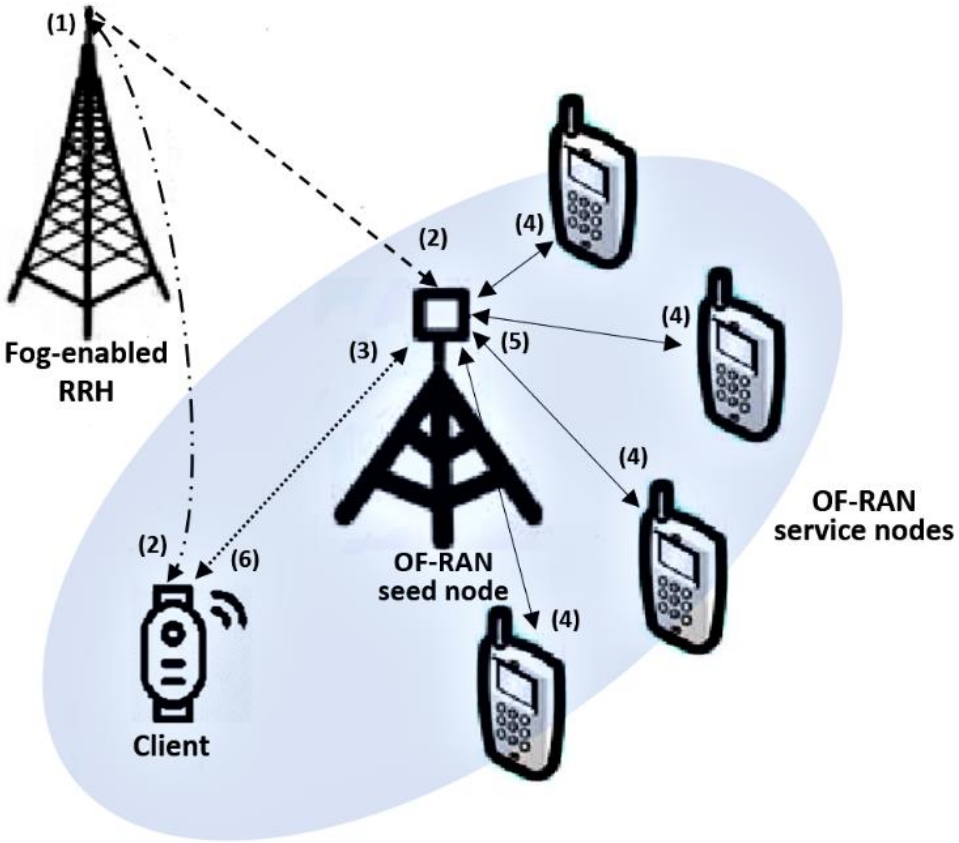

Figure 5.2 The sequence of operations of v-FAP

The energy demand for each task depends not only on its size (in bytes) and complexity (in CPU cycles), but also which service node the task has been assigned to, and the energy required by the assigned service node to perform the task, i.e. the task's energy demand can vary with different node assignment. A service node consumes energy for receiving, processing and transmitting a task. Thus, the total energy demand of a particular m^{th} task on the n^{th} service node, denoted as $e_{n,m}^{total}$, can be given by:

$$e_{n,m}^{total} = e_{n,m}^{rx} + e_{n,m}^{proc} + e_{n,m}^{tx} \quad (5.1)$$

where $n \in N$ and $m \in M$; $e_{n,m}^{rx}$ is the energy required by n^{th} node to receive m^{th} task from seed node; $e_{n,m}^{proc}$ is the energy required by n^{th} node to process m^{th} task; and $e_{n,m}^{tx}$ is the energy required

by n^{th} node to transmit processed m^{th} task to seed node. The $e_{n,m}^{rx}$, $e_{n,m}^{proc}$, and $e_{n,m}^{tx}$ can be further defined as:

$$e_{n,m}^{rx} = e_n^{rx} . s_m^{rx} \tag{5.2a}$$

$$e_{n,m}^{proc} = e_n^{proc} . s_m^{proc} \tag{5.2b}$$

$$e_{n,m}^{tx} = e_n^{tx} . s_m^{tx} \tag{5.2c}$$

where e_n^{rx} (in joules per byte) is the energy required by n^{th} node to receive one byte of task from the seed node; e_n^{proc} (in joules per CPU cycle) is the energy required by n^{th} node to process a period of one CPU cycle; e_n^{tx} (in joules per byte) is the energy required by n^{th} node to transmit one byte of processed task to the seed node; s_m^{rx} and s_m^{tx} is the size of the m^{th} task received, and processed m^{th} task transmitted, respectively in bytes; and s_m^{proc} is the processing complexity of the m^{th} task in CPU cycles.

Similarly, the time demand for each task can be constituted of processing time and communication time. Thus, the total time demand of a particular m^{th} task on the n^{th} service node, denoted as $t_{n,m}^{total}$, can be given by:

$$t_{n,m}^{total} = t_{n,m}^{rx} + t_{n,m}^{proc} + t_{n,m}^{tx} \tag{5.3}$$

where $t_{n,m}^{rx}$ is the time required by n^{th} node to receive m^{th} task from seed node; $t_{n,m}^{proc}$ is the time required by n^{th} node to process m^{th} task; and $t_{n,m}^{tx}$ is the time required by n^{th} node to transmit processed m^{th} task to seed node. The $t_{n,m}^{rx}$, $t_{n,m}^{proc}$, and $t_{n,m}^{tx}$ can be further defined as:

$$t_{n,m}^{rx} = \frac{s_m^{rx}}{r_n^{sd}} \tag{5.4a}$$

$$t_{n,m}^{proc} = \frac{s_m^{proc}}{c_n^{proc}} \tag{5.4b}$$

$$t_{n,m}^{tx} = \frac{s_m^{tx}}{r_{sd}^{n}} \tag{5.4c}$$

where c_n^{proc} is the processing speed of n^{th} node in CPU cycles per second; r_n^{sd} and r_{sd}^{n} is the data rate of communication from seed node to n^{th} node, and from n^{th} node to seed node, respectively in bits per second, which can be given by the Shannon-Hartley capacity theorem using known channel bandwidth and received signal-to-noise ratio [85].

5.2.2. Problem Formulation

For a given M number of service tasks and N number of service nodes, denote X as the set of possible TNAs, P as the cardinality of set X (i.e. $P=|X|$), and X_j as the j^{th} TNA of X where $j=1,..,P$. Figure 5.3 shows an example of TNA, denoted by X_j, for $N=4$ and $M=8$. The X_j is given as a $N \times M$ binary matrix where a non-zero entry at an n^{th} column and m^{th} row, represents a certain m^{th} task is assigned to an n^{th} node. If $x_{n.m}$ is an element of the TNA matrix, then constraints (5.5) and (5.6) for an m_{th} task, and n_{th} node, respectively, must be satisfied in defining each TNA.

$$\sum_{n=1}^{N} x_{n,m} = 1 \tag{5.5}$$

$$\sum_{m=1}^{M} x_{n,m} \leq M \tag{5.6}$$

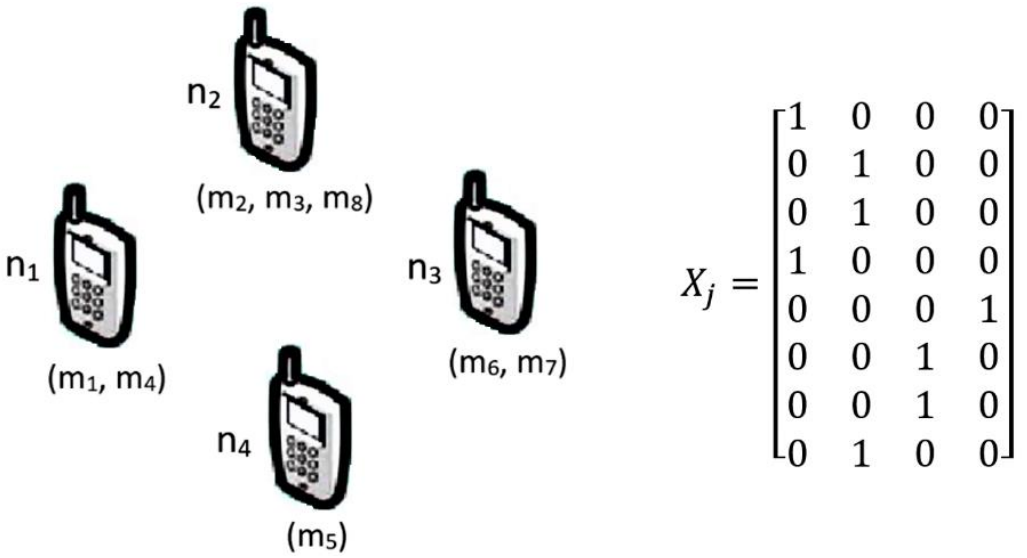

Figure 5.3 TNA example for N=4 service nodes and M=8 service tasks

The TNA is formulated as a tri-objective problem with the goals of minimizing energy consumption and service latency of the v-FAP, while maximizing fairness (load balancing) amongst its service nodes by minimizing their maximum load. In this work, the load of a service node is expressed as a fraction of its resource capacity consumed for performing its assigned task(s). A service node with a high resource capacity can thus take on a high absolute load, and vice-versa.

Each node $n \in N$ is specified with u_n and v_n, representing the per unit cost of using its energy, and time, respectively. The u_n can be made inversely proportional to the available energy capacity of the n^{th} node, i.e., higher its energy capacity, lower the cost of using its energy. Similarly, v_n can be made inversely proportional to the processing and communication speed of the n^{th} node, i.e., higher its processing and communication speed, lower the cost of using its time.

For a particular client, the total energy cost $E(X_j)$ of using the j^{th} TNA from a set X of possible TNAs for assigning M tasks to N nodes in a v-FAP is given by:

$$E(X_j) = \sum_{n=1}^{N} D_n.u_n \quad j = 1 \dots P \tag{5.7}$$

$$D_n = \sum_{m=1}^{|M_n|} R_{n,m} \quad \leq Te_n \tag{5.8}$$

where D_n is the total energy demand by the set of task(s) $M_n \subseteq M$ assigned to n^{th} *node*, $R_{n,m} = \{e_{n,1}^{total}, \dots, e_{n,|M_n|}^{total}\}$ is the set of energy demands by individual task(s) in M_n where $e_{n,m}^{total}$ is defined in (1), and Te_n is the total available energy in the n^{th} node to serve its client.

Similarly, the total time cost $T(X_j)$ of using the j^{th} TNA from a set X of possible TNAs for assigning M tasks to N nodes in a v-FAP is given by:

$$T(X_j) = \sum_{n=1}^{N} G_n.v_n \qquad j = 1 \dots P \tag{5.9}$$

$$G_n = \sum_{m=1}^{|M_n|} Z_{n,m} \quad \leq Tt_n \tag{5.10}$$

where G_n is the sum of the time demand by the set of task(s) $M_n \subseteq M$ assigned to n^{th} *node*, $Z_{n,m} = \{t_{n,1}^{total}, \dots, t_{n,|M_n|}^{total}\}$ is the set of total time demands by each task in M_n with $t_{n,m}^{total}$ as defined in (5.3), and Tt_n is the total available time of the n^{th} node to serve its client.

The maximum node usage cost $L(X_j)$, i.e., the maximum cost of using a service node in the j^{th} TNA to serve a client, is given by:

$$L(X_j) = \max_{1 \dots N} \{\alpha. D_n. u_n + \beta. G_n. v_n\} \tag{5.11}$$

where α and β are normalization factors for scaling the cost into the interval [0, 1]. For an n^{th} service node, the per unit cost of using its energy (u_n) and its time (v_n) are inversely proportional to its available energy, and processing/communication speed, respectively. The $L(X_j)$ also reflects the highest fractional use of a service node's resources, i.e., the maximum load on a service node in the j^{th} TNA.

In order to select an optimal TNA that not only minimizes $E(X_j)$ and $T(X_j)$, but also maximizes fairness amongst the service nodes by minimizing their maximum load, our tri-objective optimization problem can be defined as:

$$\underset{1....P}{\text{minimize}} \quad \{E(X_j), T(X_j), L(X_j)\} \tag{5.12}$$

5.3. MOEA/D Framework for Solving the TNA problem

Since our TNA problem is dealing with conflicting objectives, it cannot be solved by optimizing each objective individually while treating all other objectives as constraints, as there is no single solution that can optimize all objectives simultaneously. Hence, we designed a solution framework based on the MOEA/D algorithm, which gives a promising (pareto-optimal) set of solutions with reasonable tradeoffs between the objectives [86], i.e. all solutions in the set is non-dominant and the decision-maker can select the best TNA from the set based on the desired outcome.

A multi-objective problem (MOP) can be defined as:

$$\underset{1....P}{\text{minimize}} \quad F(X_j) = \left(f_1(X_j), f_2(X_j), \ldots, f_k(X_j)\right) \tag{5.13}$$

where k is the number of objectives in our problem, and parameters X_j, P, and j, are as defined in Section 3.2. Here, $F : \omega \rightarrow R^k$ where ω is the decision variable space which maps to the objective function space R^k. Hence, $\{f_i(X_j) \mid X \in \omega;\ i = 1..k\}$ is the attainable objective set, while $f_1(X_j)$, $f_2(X_j)$, and $f_3(X_j)$ are corresponding to $E(X_j)$, $T(X_j)$, and $L(X_j)$ in (5.12), respectively.

5.3.1. Decomposition

It is known that the pareto-optimal solution to any MOP can be the optimal solution to a number of scalar (single objective) optimization sub-problems whose objective is the weighted aggregation of all the f_i's [87]. The decomposition of a MOP into a set of sub-problems are commonly based on the weighted-sum and Tchebycheff approaches. We adopted the Tchebycheff approach as it produces more diverse pareto fronts, including convex, concave, mixed, and other geometries. The approach uses a weight vector $\lambda^j = (\lambda_1^j, .. \lambda_k^j)$ consisting of k weights (one for each objective) for the j_{th} sub-problem where $\sum_{i=1}^{k} \lambda_i^j = 1$ and $\lambda_i^j \geq 0$. Our MOP is decomposed into P sub-problems. For each j_{th} sub-problem, different weights λ_i^j are applied to the objective function f_i as shown in (14). The X_j is the solution variable (chromosome) for the j_{th} sub-problem representing the TNA to be optimized, $z = (z_1, .., z_k)$ is the set of reference points holding the minimum of each of the k objectives values f_i's in the decision variable space, i.e. $\min\{f_i(X_j) \mid X \in \omega;\ i = 1..k\}$, $\lambda = (\lambda^1, .., \lambda^P)$ is the set of weight vectors for the P sub-problems, and $g(X \mid \lambda, z)$ determines the fitness of each X_j in X by obtaining the maximum of all k weighted f_i's for that assignment. The fittest assignment is then the one with the minimum fitness value:

$$\text{minimize} \quad g(X \mid \lambda, z) = \max_{1 \leq i \leq k} \{\lambda_i^j . |f_i(X_j) - z_i|\} \tag{5.14}$$

5.3.2. MOEA/D Framework

An initial internal population (IP) consisting of P possible TNAs (parent chromosomes) is generated. For each j_{th} of the P sub-problems, a set of T closest neighboring sub-problems is determined based on T shortest Euclidean distances between its weight vector λ^j and those of other $(P - 1)$ sub-problems. Two parent chromosomes P_{c1} and P_{c2} are then randomly selected from the solutions (TNAs) of the T neighboring sub-problems to generate an offspring or new solution Y (new TNA). An update phase follows, where the reference points set z is updated if the offspring

Y results in a new minimum of each of the k objectives values f_i's; and the T closest neighbors of the j_{th} sub-problem are updated if the offspring Y provides a better solution in terms of fitness.

For every generation, the IP is also updated to reflect the best solutions found so far for each sub-problem. All non-dominated solutions found during the search are placed in an external population (EP). A solution v is said to dominate another solution v' if and only if it is equal or better than v' in all objectives, and there is at least one objective where v is strictly better, i.e. v is non-dominated by v'. Thus, in a set of non-dominated solutions, no solution can dominate the others in all objectives. The EP is updated for every generation and the pareto front is plotted using the final set of non-dominated solutions. Algorithm 5.1 shows our MOEA/D framework for the TNA problem.

Algorithm 5.1 MOEA/D framework for the TNA problem

Input:
    TNA parameters (M, N, D_n, u_n, G_n, v_n);
    P: population size and number of sub-problems;
    T: neighborhood size;
    λ: set of weight vectors for the P sub-problems;
    G_{max}: maximum number of generations (beyond which no further addition of non-dominated solutions to the EP is normally observed);
Output: EP
Step 1) Initialization
    Set $EP = \emptyset$; $G = 0$; $z = \emptyset$;
    Generate an initial $IP = \{X_1, .., X_P\}$ randomly subject to the constraints in Equations (5) and (6);
    Determine the T closest neighborhood for each j^{th} of P sub-problems;
Step 2) Reproduction and update
    for $j = 1..P$ **do**
        Randomly select two T closest neighbor solutions and generate a new solution Y using the genetic operators;
        Use Y to update z, T closest neighbor solutions, IP and EP;
    end
Step 3) Stopping criterion
    if $G = G_{max}$ **then**
        Stop and output EP;
    else
        Increment G and go to Step 2);
    end

5.3.2.1. Chromosome Encoding

In MOEA/D, the solutions are represented as chromosomes. For our specific TNA problem, the TNAs are encoded as chromosomes, and then MOEA/D is employed to evolve them into optimal TNAs subject to our energy, latency, and fairness (load balancing) objectives. Given the discrete nature of our problem, decimal encoding is introduced to express the TNAs as chromosomes. Each chromosome (TNA) has a certain length (number of service tasks) and carries multiple genes (service nodes). Therefore, each gene's value is the index of a service node, while its locus (position) in the chromosome denotes the index of a service task assigned to the service node represented by the gene.

Figure 5.4 shows an example of our chromosome encoding for the case of 4 service nodes and 8 service tasks (N=4, M=8). The encoded chromosome represents a specific TNA, where tasks 1 and 4 (m1 and m4) are assigned to node 1 (n=1); tasks 2, 3, and 8 are assigned to node 2; tasks 6 and 7 are assigned to node 3; and finally, task 5 is assigned to node 4.

Locus (m)	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
Genic Value (n)	1	2	2	1	4	3	3	2

Figure 5.4 Example of proposed chromosome encoding

5.3.2.2. Crossover Operators

In order to evolve the chromosomes (TNAs), the MOEA/D applies a crossover operator to a pair of parent chromosomes and generate an offspring (new TNA). We consider three crossover operators: *one-point, two-point* and *uniform* crossover as described below, and then determine which is the most appropriate in the next section.

- ***One-point crossover***: Two parent chromosomes (Pr_1, Pr_2) of length *M* are selected, and a random crossover point is chosen between 1 and *M*. Each chromosome is then sliced into

two segments which are exchanged to produce offspring, from which an offspring Y is randomly selected.

- ***Two-point crossover***: The process is similar to one-point crossover, except two instead of one random crossover points are chosen for segmenting the chromosomes.
- ***Uniform crossover***: Unlike above, offspring here are produced from an exchange of genes uniformly selected from two parent chromosomes: from odd-index locus of Pr_1 and even-index locus of Pr_2, and vice-versa; from which an offspring Y is randomly selected.

5.4. Simulation Environment

The designed framework is implemented in MATLAB and evaluated under varying number of service nodes (N) and service tasks (M). The impacts of different crossover operators on the produced solutions are also investigated. Finally, a comparison is made with solutions obtained using the non-dominated sorting genetic algorithm (NSGA) II, which is a competing alternative to MOEA/D. All obtained solutions (TNAs) are evaluated for their incurred objective costs, i.e. total energy cost E, total time (latency) cost T, and maximum node usage cost L, as defined by equations (5.7), (5.9), and (5.11), respectively.

The cost L of a TNA reflects the maximum load that it assigns to its service nodes. Thus, minimizing L is also maximizing fairness (load balancing) amongst the service nodes in a TNA. However, instead of L, it is more intuitive to show the standard deviation (S.D.) of the loads between service nodes in each TNA in order to reflect fairness. The solutions obtained are plotted as a three-dimensional pareto-front, one dimension for each objective. All objective costs on the pareto-front are normalized to the interval [0,1]. Table 5.1 shows the simulation settings used.

Table 5.1 Simulation Settings

Parameter	Value
Number of service nodes (*N*)	2, 4, 6 (default=4)
Number of service tasks (*M*)	4, 8,12 (default=8)
Crossover Operator	One-point; Two-point and Uniform Crossover
Population Size	$\frac{(N+M-1)!}{M!\,(N-1)!}$
Neighborhood Size	25

The population size is the number of ways to assign *M* tasks to *N* nodes, which is equivalent to the number of ways to choose *k* from *n* items found by binomial coefficient $\binom{n}{k}$ where *n*=(*N*+*M*-1) and *k*=(*N*-1). To compare solutions from MOEA/D and NSGA-II, a statistical analysis using set coverage (C) metric is performed. For two solution sets *A* and *B*, the C-metric is defined as the percentage of solutions in *B* dominated by at least one solution in *A*:

$$C(A,B) = \frac{|\{b \in B | \exists a \in A : a \ dominates \ b\}|}{|B|} \tag{5.15}$$

5.5. Results and Discussion

Figure 5.5 shows the pareto-front obtained under varying number of service nodes (*N*=2, 4, 6), while maintaining the number of service tasks to its default case (*M*=8). It firstly shows that the number of pareto-points (the set of pareto-optimal solutions) expectedly increases with *N* as more service nodes leads to a larger population size (of TNAs) and more diverse genic value of the parent chromosomes. Moreover, it decreases the standard deviation of the loads between the service nodes, resulting in a greater fairness as more nodes share the workload. Table 5.2 shows the mean energy and latency costs of the obtained TNAs decrease as *N* increases. This may be attributed to the wider selection choices of service nodes with different amount of energy and processing resources.

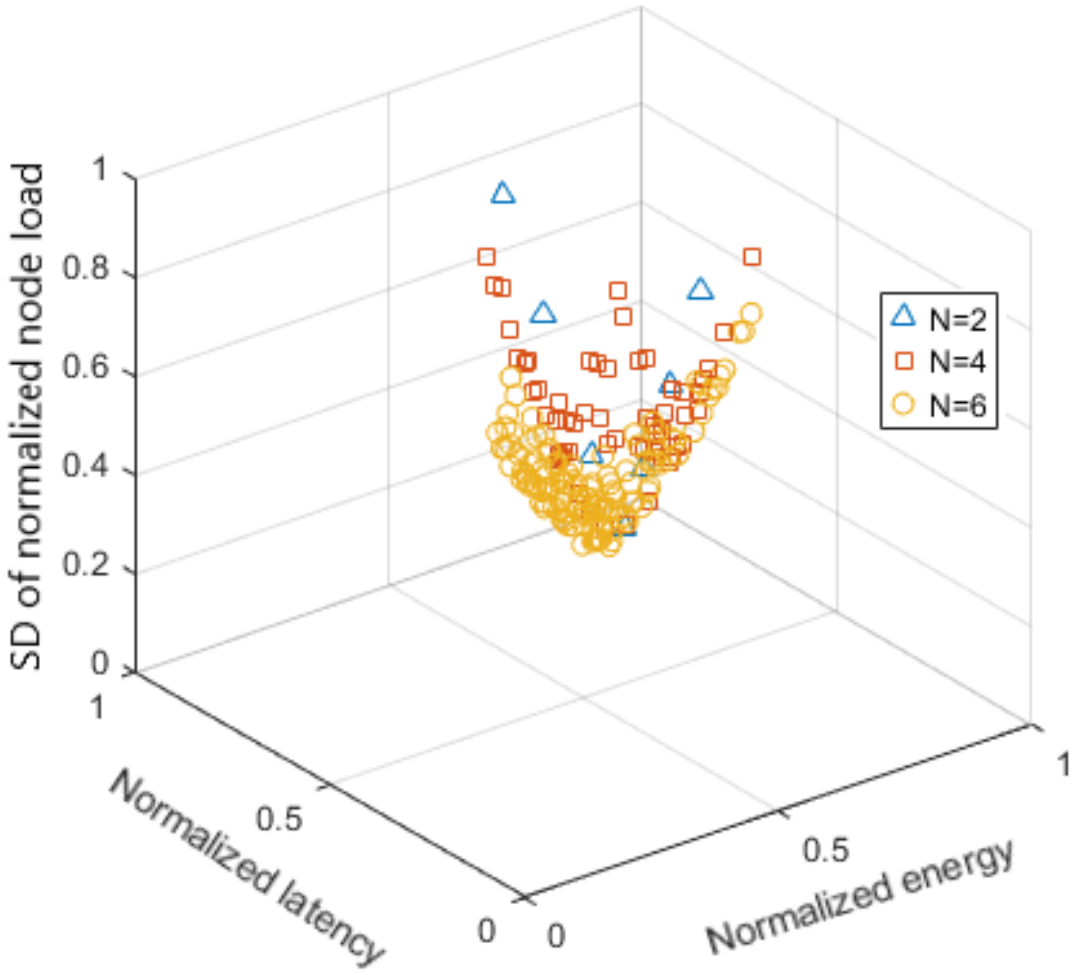

Figure 5.5 Pareto-fronts obtained under varying number of service nodes (N=2, 4, 6) and a default number of service tasks (M=8)

Table 5.2 Mean and Max values of the objective costs under varying N

N	Normalized Energy		Normalized Latency		S.D. of Normalized Node Load	
	Mean	Max	Mean	Max	Mean	Max
2	0.8329	0.9377	0.8627	0.9746	0.3690	0.7322
4	0.8145	1.00	0.8348	1.000	0.3257	0.6167
6	0.7562	1.00	0.8059	0.9004	0.2244	0.5036

Figure 5.6 shows the pareto-fronts obtained under varying number of service tasks (M=4, 8, 12), while maintaining the number of service nodes to its default value (N=4). The number of pareto-

points clearly increases with *M* due to a greater population size and diversity of parent chromosomes arising from longer chromosome lengths. In addition, the pareto-fronts can be seen to shift towards higher energy and latency costs, which are confirmed by their higher mean and maximum values as shown in Table 5.3 This is because as *M* increases, the demand for energy and time resources on each service node in a TNA increases, resulting in higher overall energy and latency costs of the TNAs.

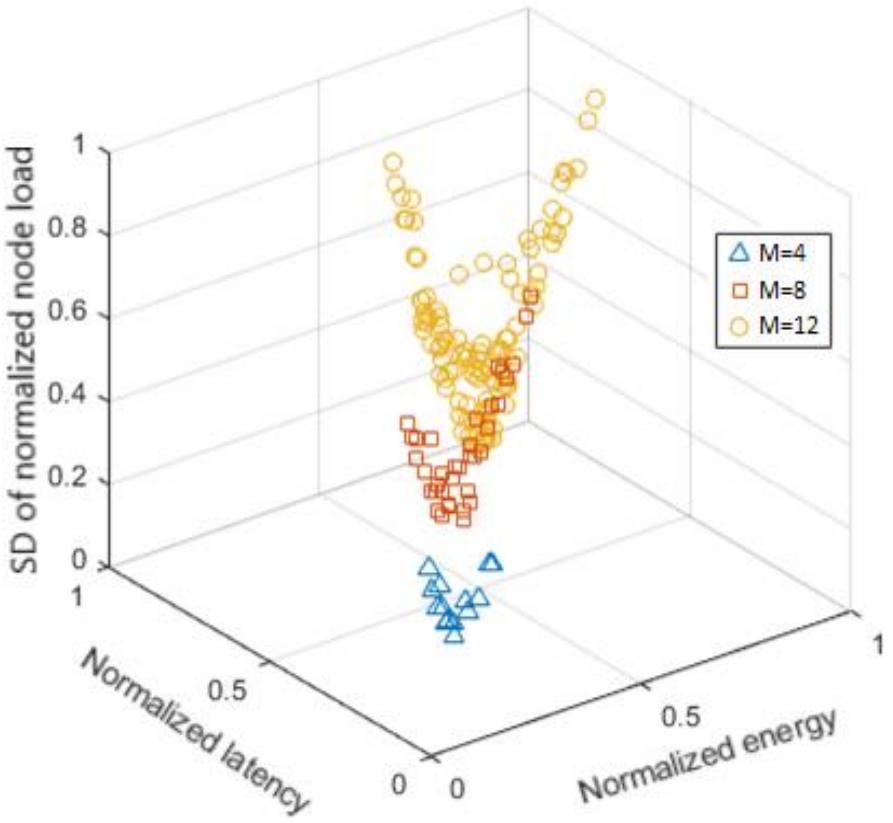

Figure 5.6 Pareto-fronts obtained under varying number of service tasks (*M*=4, 8, 12) and a default number of service nodes (*N*=4)

Table 5.3 Mean and Max values of the objective costs under varying *M*

M	Normalized Energy		Normalized Latency		*S.D.* of Normalized Node Load	
	Mean	Max	Mean	Max	Mean	Max
4	0.2617	0.3339	0.2650	0.2790	0.1640	0.2539
8	0.5489	0.6778	0.6242	0.6684	0.2760	0.6087
12	0.7872	1.00	0.8713	0.9253	0.3550	0.8722

The impact of utilizing different crossover operators: single-point, two-point, and uniform crossover, on the obtained pareto-fronts under a default number of service nodes and service tasks (N=4, M=8) is shown in Figure 5.7 It is found that while uniform crossover produced fewer pareto-points (55) than one-point (58) and two-point (64) crossover, the solutions obtained are more optimal than those under one-point and two-point crossover. This can be observed from their closer proximity to the most optimal point, which is the point-of-origin (0,0,0). This observation is also statistically verified by a C-metric comparison of the solutions as shown in Table 5.4.

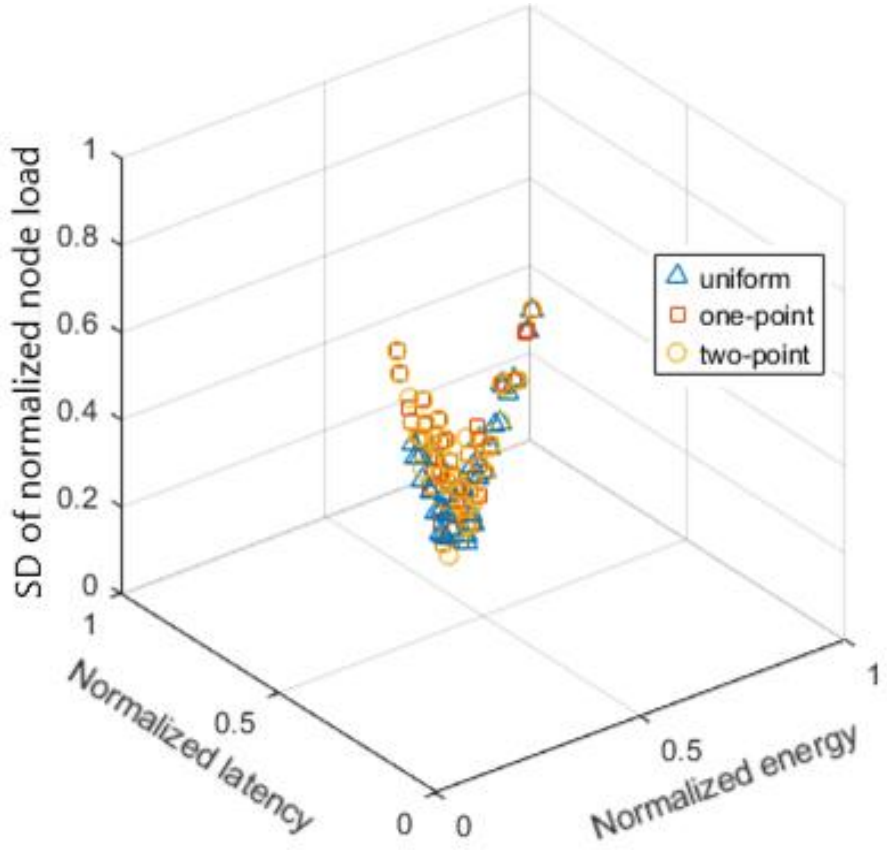

Figure 5.7 Pareto-fronts obtained under different crossover operators and a default number of service nodes (N=4) and service tasks (M=8)

Table 5.4 C-Metric of solutions obtained under one-point, two-point and uniform crossover

C-Metric	(%)
C (Uniform, One-point)	100
C (One-point, Uniform)	81.81
C (Uniform, Two-point)	100
C (Two-point, Uniform)	81.81
C (One-point, Two-point)	98.44
C (Two-point, One-point)	98.27

Figure 5.8 compares the obtained solutions by MOEA/D with those obtained by a competing technique, NSGA-II, under a default number of service nodes and service tasks (N=4, M=8). The pareto-points of MOEA/D are found to have slightly lower values (closer to the point-of-origin), suggesting the solutions are more optimal than those from NSGA-II.

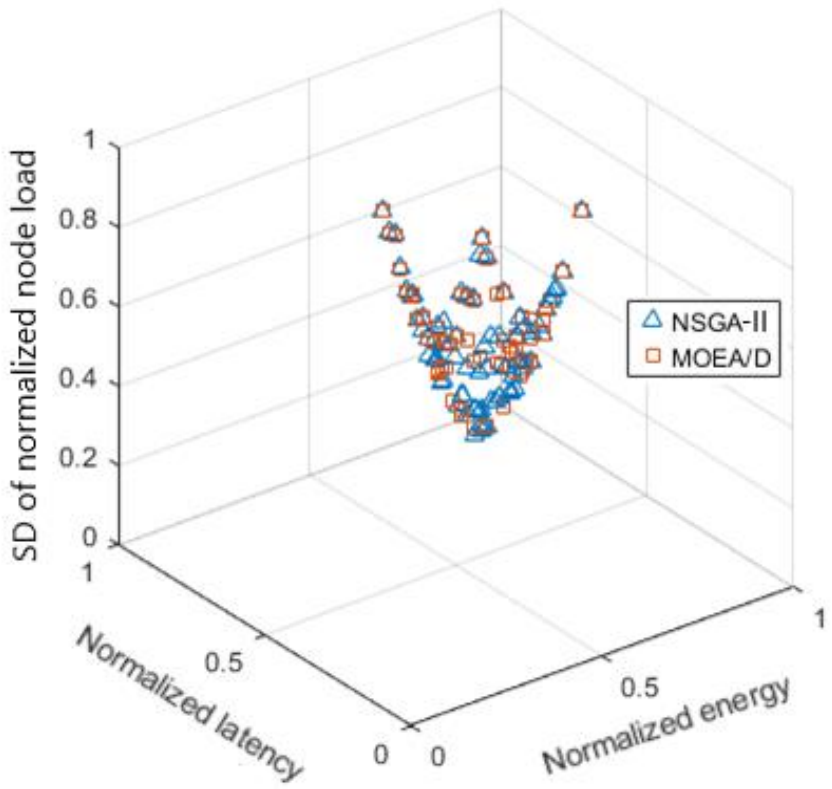

Figure 5.8 Pareto-fronts obtained from MOEA/D and NSGA-II under default number of service nodes (N=4) and service tasks (M=8)

This is confirmed by the average C-Metric values obtained over 20 trials under MOEA/D and NSGA-II as shown in Table 5.5. It can be seen that MOEA/D has a higher percentage of non-dominated solutions than NSGA-II. The 95% confidence intervals (CI) of the C-Metric values further show that the C-Metric range of both algorithms do not overlap, indicating their performance difference are statistically significant.

Table 5.5 C-Metric of solutions obtained under MOEA/D and NSGA-II

C-Metric	**(%)**	**95% CI**
C (MOEA/D, NSGA-II)	90.22	[77.02, 103.42]
C (NSGA-II, MOEA/D)	56.47	[38.10, 74.85]

In order to quantify the statistical significance, a two-sample t-test is performed for the two average C-Metric values as shown in Table 5.6. For a significance level of $\alpha = 0.05$, it is observed that $P(T <= t) < \alpha$ with $t_{stat} > t_{critical}$ for both one-tailed and two-tailed t-tests, which justifies the rejection of the null hypothesis that the performances of both algorithms based on their C-Metric values are equivalent; and accept the alternative hypothesis that the performances of both algorithms are statistically significantly different.

Table 5.6 Two-sample t-test for C (MOEA/D, NSGA-II) and C (NSGA-II, MOEA/D)

Parameters	Values
t_{stat}	2.5287
α	0.05
$P(T <= t)\ one\text{-}tail$	0.01023
$t_{critical}\ one\text{-}tail$	1.7291
$(T <= t) two\text{-}tail$	0.02046
$t_{critical}\ two\text{-}tail$	2.0930

5.6. Chapter Summary

In this chapter, the TNA problem in OF-RAN is reformulated and solved as a tri-objective optimization problem using MOEA/D. The objectives are to minimize the energy and latency costs of the v-FAP, while maximizing fairness among its service nodes by minimizing their

maximum load. The impact of different number of service nodes, number of service tasks, and crossover operator on the obtained solutions (TNAs) are investigated.

It is found that a higher number of service nodes in a v-FAP increases the number of solutions, while decreases the energy cost, latency cost, and the load standard deviation of the service nodes (suggesting better load balancing). On the other hand, a higher number of service tasks not only increases the number of solutions, but also increases the energy cost, latency cost, and the load standard deviation of the service nodes. Among the considered crossover operators, the uniform crossover produces fewer total number of solutions, but a larger number of non-dominant solutions than the one-point and two-point crossover. Finally, the solutions obtained from MOEA/D are found to be more optimal than those obtained from NSGA-II for our OF-RAN's TNA problem.

CHAPTER VI: Analytical Performance Modelling of OF-RAN

6.1. Introduction

The advancements and convergence in wireless, computing, sensor and actuation technologies have enabled a plethora of smart devices collectively known as the IoT [88]. These devices are deployed ubiquitously and in large numbers for a diverse range of applications, leading to massive data generation and an exponential growth in demand for transmission and computation resources. In order to meet these challenges, network architectures such as C-RAN [89] and F-RAN [82] have been introduced. Although C-RAN has immense computation resources in the cloud, it suffers from a number of drawbacks, such as heavy workload at the centralized BBU pool, stringent backhaul capacity constraint, and difficulty in catering to delay sensitive applications [4]. F-RAN, on the other hand, deploys FAPs at network edge to provide cloud-like services to IoT devices. The FAPs can be deployed as new dedicated entities in an existing infrastructure, or on existing entities of an infrastructure such as a small cell base station augmented with fog functionality [5].

Recently, we have proposed the OF-RAN [26], which is evolved from the concepts of F-RAN and oppnets [8]. The latter are mission-oriented ad hoc networks setup to utilize opportunistically available local resources. Each oppnet grows from a 'seed' node, which recruits one or more available local 'helper' nodes to assist with a specific mission. In our proposed OF-RAN, the *seed node* and *service nodes* are equivalent to the FAP of F-RAN, and helper nodes of oppnet, respectively. A seed node recruits locally available resourceful user devices such as high-end smart phones as service nodes that function collectively as a v-FAP to serve a resource-limited client such as an IoT device. The resourceful user devices can be incentivized as in [90] to lease their resources (e.g. computing, storage, and energy resources) for serving resource-limited

clients and be remunerated based on their performance (e.g. in terms of timeliness and reliability). The computation to be offloaded from a client to a v-FAP is referred as *service task*.

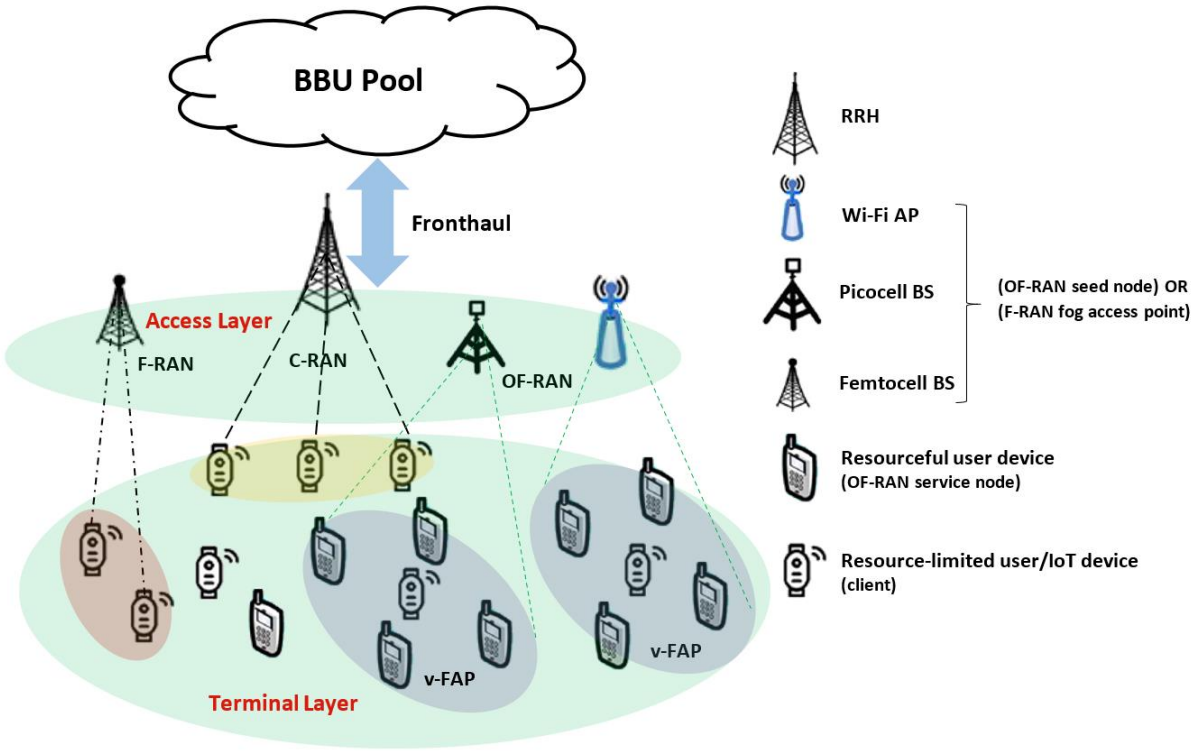

Figure 6.1 System architecture of co-existing RANs.

We consider a scenario shown in Figure 6.1 where OF-RAN, F-RAN, and C-RAN co-exist in the access layer to serve a terminal layer composing of both resourceful and resource-limited user/IoT devices. Our proposed OF-RAN can play a complementary role to F-RAN and C-RAN by harnessing resourceful terminal devices to deal with the computation workloads from a large number of offloading clients simultaneously in a time- and energy-efficient manner. A resource-limited client can offload its task in three ways: (i) offload to C-RAN by transmitting the task to the BBU in the cloud through a RRH; (ii) offload to F-RAN by transmitting the task to the FAP; (iii) offload to OF-RAN by transmitting the task to the v-FAP. In F-RAN or OF-RAN offloading where multiple FAPs or v-FAPs are available, the RRH could assist the client in selecting the most appropriate FAP or v-FAP for offloading. To provide insights into the

complementary nature of these RAN architectures, we develop an analytical model to evaluate their performances in terms of the energy consumption, completion delay, and failure rate, under various offloading scenarios.

The rest of the chapter is organized as follows. Section 6.2 presents the system model. Section 6.3 develops the analytical models for the three RAN architectures under consideration. Sections 6.4 and 6.5 discuss the simulation environment, and the results, respectively. Finally, Section 6.6 concludes with the chapter summary.

6.2. System Model

6.2.1. Network Model

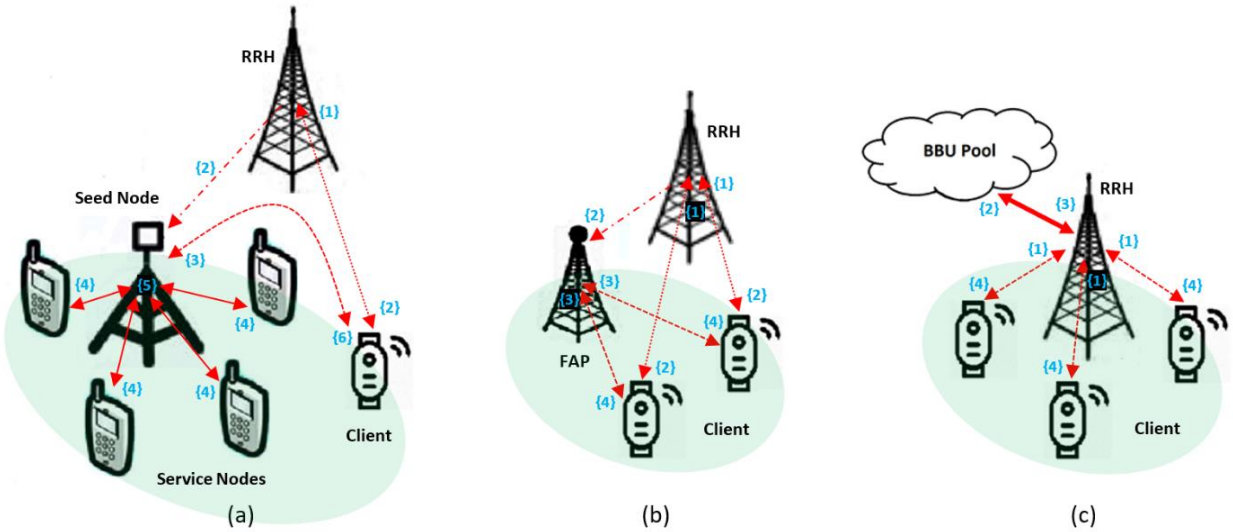

Figure 6.2 Network model for: (a) OF-RAN; (b) F-RAN; and (c) C-RAN.

Figure 6.2 shows the considered network model for our proposed OF-RAN as well as existing F-RAN and C-RAN architectures. In the OF-RAN, a resource-limited client offloads its task by first sending a request {1} including the task requirements to its associated RRH, which in tum notifies the client {2} to offload its task to an available seed node within its neighborhood. The client then sends its task {3} to this seed node for processing. Upon receiving, the seed node

firstly determines the optimal TNA based on the performance objectives [91]. The optimal TNA defines a set of suitably selected service nodes for the v-FAP, and appropriately sized sub-tasks to be assigned to each service node. Based on this assignment, the seed node sends the sub-tasks to the service nodes for processing {4}, collates the processed sub-tasks from the service nodes {5}, and forwards them to the client {6}. To emulate a real-world scenario, we consider both service nodes and service tasks to be heterogeneous, each having a different computation capacity, and complexity, respectively.

We further consider that the service nodes are registered RAN users that can be trusted to assist the resource-limited clients when called upon. This trust can be facilitated by a blockchain-enabled OF-RAN architecture [92] in which the smart contract is used to implement an algorithm for distributed formation and management of v-FAPs among trustless user devices acting as service nodes.

In the F-RAN, a client similarly offloads its task by first sending a request {1} to its associated RRH, which acts as a F-RAN controller [93] in charge of receiving offloading requests and distributing them to the FAPs. The RRH then notifies the client {2} to offload its task to an available FAP within its neighborhood. The client then sends its task {3} to this FAP for processing. On completion, the FAP forwards the processed task to the client {4}. However, unlike in OF-RAN where each v-FAP only serves a single client, the FAP in F-RAN may serve multiple clients at a time.

On the other hand, a client in C-RAN offloads its task {1} to the associated RRH, which in turn sends it to BBU pool in the cloud {2} for processing. The RRH receives the processed task {3} from BBU pool, and then forwards it to the client {4}. The wireless access links between RRH, clients, FAPs, seed nodes, and service nodes are considered to be using mmWave, while the wired fronthaul link between the RRH and BBU pool in the cloud is using an optical fiber.

6.2.2. Path Loss Model

The path loss model calculates the power loss of a signal as it travels through space. For the wireless access links in this chapter, the close-in (CI) free-space reference distance model [94] proposed for 5G systems is used to calculate the path loss. Compared to other path loss models such as 3GPP's alpha-beta-gamma (ABG) model, the CI model offers computation simplicity yet better accuracy in path loss prediction across a wide range of frequencies and distances [95]. The path loss $PL_{u,v}$ in decibel (dB) of a link from node u to node v is given by (6.1), where f is the signal frequency, d_0 is the close-in free space reference distance in meters, c is the speed of light in meters per second, α is the path loss exponent, $d_{u,v}$ is the distance between node u and node v in meters, and X_σ is the shadowing component in dB described by a zero-mean Gaussian random variable with standard deviation σ.

$$PL_{u,v}(dB) = 20\ log_{10}\left(\frac{4\pi f d_o}{c}\right) + 10\alpha\ log_{10}\left(\frac{d_{u,v}}{d_o}\right) + X_\sigma \tag{6.1}$$

The corresponding received signal power P_v^{rx} in decibel-milliwatts (dBm), and data rate $R_{u,v}$ in bits per second (bps), based on the determined path loss, are given by (6.2), and (6.3), respectively, where P_u^{tx} is the transmitted signal power in dBm of node u, b is the channel bandwidth, and P_v^{no} is the average noise power in dBm at the receiver node v.

$$P_v^{rx}(dBm) = P_u^{tx}(dBm) - PL_{u,v}(dB) \tag{6.2}$$

$$R_{u,v} = b\ log_2\left(1 + \frac{P_v^{rx}}{P_v^{no}}\right) \tag{6.3}$$

6.3. Analytical Model

This section presents the analytical model for evaluating the offloading performance of the OF-RAN, F-RAN, and C-RAN as illustrated in Figure 6.2 In this model, expressions are obtained for three system level performance metrics, namely total delay, total energy consumption, and offloading failure.

6.3.1. Delay

In OF-RAN, the total delay D_{OF}^{total} incurred while offloading a task from the client to a v-FAP comprising of one seed node and N service nodes, consists of transmission, propagation, and processing delays. As shown in (6.4a), the transmission delay includes the time for sending a request of size ∂ from client to RRH, a notification of size φ from RRH to client, a task of size T (or T' after processing) between client and seed node, and N sub-tasks each of size M_n (or M_n' after processing) between seed node and N service nodes, where n is the index of a service node and $\sum_{n=1}^{N} M_n = T$.

The propagation delay between a transmitting node u and receiving node v is given by the ratio of their distance $d_{u,v}$ and the speed of light c. The processing delay incurred by a service node $Sv_{(n)}$ for a sub-task of size M_n is given by the ratio of the number of floating-point operations (FLOPs) required by the sub-task (depending on M_n in bits and task complexity γ in FLOPs per bit) and the computation capacity C_{Sv} of the service node in FLOPs per second (FLOPS). Without loss of generality, we assume all service nodes have the same computation capacity C_{Sv}, which can be found using (6.4b) where δ_{Sv} is the service node's performance in FLOPs per cycle per core, β_{Sv} is the number of cores, and ζ_{Sv} is the processor frequency in hertz (or cycles per second). For a seed node with N or more antennas, it can transmit all N sub-tasks at the same time using one antenna for each service node. Thus, all N sub-tasks can be processed in parallel by the service nodes. Likewise, the seed node can simultaneously receive the processed sub-tasks from

all N service nodes. Hence, the delay between the seed node and service nodes, which include the time for transmission, propagation, and processing of all N sub-tasks, is the maximum of all pair-wise delays between the seed node and each service node.

$$D_{OF}^{total} = \left(\frac{\partial}{R_{C,R}} + \frac{d_{C,R}}{c}\right) + \left(\frac{\varphi}{R_{R,C}} + \frac{d_{R,C}}{c}\right) + \left(\frac{T}{R_{C,S}} + \frac{d_{C,S}}{c}\right) + \max_{n=1\ldots N}\left\{\left(\frac{M_n}{R_{S,Sv(n)}} + \frac{d_{S,Sv(n)}}{c}\right) + \frac{\gamma M_n}{C_{Sv}} + \left(\frac{M_n'}{R_{Sv(n),S}} + \frac{d_{Sv(n),S}}{c}\right)\right\} + \left(\frac{T'}{R_{S,C}} + \frac{d_{S,C}}{c}\right) \quad (6.4a)$$

$$C_{Sv} = \beta_{Sv}\delta_{Sv}\zeta_{Sv} \quad (6.4b)$$

In F-RAN, the total delay D_{FR}^{total} incurred while offloading a task from the client to a FAP is similarly derived as shown in (6.5a). The processing delay incurred by a FAP for a task of size T is simply given by the ratio of the number of FLOPs required by the task (depending on T and task complexity γ) and the computation capacity C_F of the FAP in FLOPS. Like OF-RAN, the C_F can be found using (6.5b) where δ_F, β_F, and ζ_F refers to the FAP's number of FLOPs per cycle per core, the number of cores, and processor frequency, respectively.

$$D_{FR}^{total} = \left(\frac{\partial}{R_{C,R}} + \frac{d_{C,R}}{c}\right) + \left(\frac{\varphi}{R_{R,C}} + \frac{d_{R,C}}{c}\right) + \left(\frac{T}{R_{C,F}} + \frac{d_{C,F}}{c}\right) + \frac{\gamma T}{C_F} + \left(\frac{T'}{R_{F,C}} + \frac{d_{F,C}}{c}\right) \quad (6.5a)$$

$$C_F = \beta_F\delta_F\zeta_F \quad (6.5b)$$

In C-RAN, the total delay D_{CR}^{total} incurred while offloading a task from the client to BBU is given by (6.6a), in which the transmission delay includes not only the time for sending a task of size T (or T' after processing) between the client and RRH, but also between the RRH and BBU via the optical fronthaul, where $c_{(op)}$ denotes propagation speed in the optical fiber. Like F-RAN, the processing delay incurred by a BBU for a task of size T is given by the ratio of the FLOPs

required by the task (depending on T and task complexity γ) and the computation capacity C_B of the BBU in FLOPS. Similarly, the C_B can be found using (6.6b), where δ_B, β_B, and ζ_B refers to the BBU's number of FLOPs per cycle per core, the number of cores, and processor frequency, respectively.

$$D_{CR}^{total} = \left(\frac{T}{R_{C,R}} + \frac{d_{C,R}}{c}\right) + \left(\frac{T}{R_{R,B}} + \frac{d_{R,B}}{c_{(op)}}\right) + \frac{\gamma T}{C_B} + \left(\frac{T'}{R_{B,R}} + \frac{d_{B,R}}{c_{(op)}}\right) + \left(\frac{T'}{R_{R,C}} + \frac{d_{R,C}}{c}\right) \quad (6.6a)$$

$$C_B = \beta_B \delta_B \zeta_B \quad (6.6b)$$

6.3.2. Energy

In OF-RAN, the total energy E_{OF}^{total} incurred while offloading a task from the client to a v-FAP is constituted of communication energy and processing energy, as shown in (6.7a). The communication energy includes the energy for sending a request of size ∂ from client to RRH, a notification of size φ from RRH to client, a task of size T (or T' after processing) between client and seed node, and N sub-tasks each of size M_n (or M_n' after processing) between seed node and N service nodes, where n is the index of a service node and $\sum_{n=1}^{N} M_n = T$.

The processing energy depends on the service node's energy efficiency E_{Sv} in joules per cycle, the size of each sub-task M_n in bits, and the OF-RAN computation intensity ω_{OF} in CPU cycles per bit. The ω_{OF} can be found using (6.7b) where γ is the task complexity in FLOPs per bit, δ_{Sv} is the service node's performance in FLOPs per cycle per core, and β_{Sv} is the number of cores.

$$E_{OF}^{total} = \frac{\partial P_C^{tx}}{R_{C,R}} + \frac{\varphi P_R^{tx}}{R_{R,C}} + \frac{T P_C^{tx}}{R_{C,S}} + \sum_{n=1}^{N} \frac{M_n P_S^{tx}}{R_{S,Sv(n)}} + \sum_{n=1}^{N} \omega_{OF} M_n E_{Sv} + \sum_{n=1}^{N} \frac{M_n' P_{Sv(n)}^{tx}}{R_{Sv(n),S}} + \frac{T' P_S^{tx}}{R_{S,C}} \quad (6.7a)$$

$$\omega_{OF} = \frac{\gamma}{\beta_{Sv} \delta_{Sv}} \quad (6.7b)$$

In F-RAN, the total energy E_{FR}^{total} incurred while offloading a task from the client to a FAP is similarly derived as shown in (6.8a). The processing energy depends on the FAP energy efficiency E_F in joules per cycle, the size of task T in bits, and the F-RAN computation intensity ω_{FR} in CPU cycles per bit. Like OF-RAN, the ω_{FR} can be found using (6.8b) where γ is the task complexity, δ_F is the FAP performance in FLOPs per cycle per core, and β_F is the number of cores.

$$E_{FR}^{total} = \frac{\partial P_C^{tx}}{R_{C,R}} + \frac{\varphi P_R^{tx}}{R_{R,C}} + \frac{T P_C^{tx}}{R_{C,F}} + \omega_{FR} T E_F + \frac{T' P_F^{tx}}{R_{F,C}} \tag{6.8a}$$

$$\omega_{FR} = \frac{\gamma}{\beta_F \delta_F} \tag{6.8b}$$

In C-RAN, the total energy E_{CR}^{total} incurred while offloading a task from the client to BBU is given by (6.9a), in which the communication energy includes not only the energy for sending a task of size T (or T' after processing) between the client and RRH, but also between the RRH and BBU via the optical fronthaul, where $P_{R(op)}^{tx}$ and $P_{B(op)}^{tx}$ denotes the optical transmit power of RRH, and BBU, respectively. Similarly, the processing energy depends on the BBU energy efficiency E_B in joules per cycle, the size of task T in bits, and the C-RAN computation intensity ω_{CR} in CPU cycles per bit given by (6.9b).

$$E_{CR}^{total} = \frac{T P_C^{tx}}{R_{C,R}} + \frac{T P_{R(op)}^{tx}}{R_{R,B}} + \omega_{CR} T E_B + \frac{T' P_{B(op)}^{tx}}{R_{B,R}} + \frac{T' P_R^{tx}}{R_{R,C}} \tag{6.9a}$$

$$\omega_{CR} = \frac{\gamma}{\beta_B \delta_B} \tag{6.9b}$$

6.3.3. Failure

The percentage of offloading failure is another performance metric evaluated. Two possible factors of failure considered are: (i) link failure; and (ii) completion time failure. In OF-RAN, there are wireless links between the client, RRH, seed node, and service nodes of a v-FAP. A wireless link from a transmitting node u to a receiving node v (where u and v can be the client,

RRH, seed node, or service node) is considered to fail when the received power P_v^{rx} is below the receiver sensitivity τ_v.

Even when all the links are successful, an offloading can still fail when the total delay D_{OF}^{total} incurred to complete a task is longer than the completion time requirement ϕ of the task. Hence, for a given client C, the offloading is deemed to have failed when either a link or completion time failure occurs, as shown by the failure conditions given in (6.10):

$$(\forall\, v \in V,\ P_v^{rx} < \tau_v) \vee \left(D_{OF}^{total} > \phi\right) \tag{6.10}$$

where V is the set of receiving nodes for wireless links used in offloading for C in OF-RAN.

In F-RAN, there are wireless links between the client, RRH, and FAP. Similarly, the offloading is deemed to have failed when the failure conditions in (6.11) are satisfied, where D_{FR}^{total} is the total delay incurred to complete a task for the client in F-RAN.

$$(\forall\, v \in V, \quad P_v^{rx} < \tau_v) \vee \left(D_{FR}^{total} > \phi\right) \tag{6.11}$$

In C-RAN, there are not only wireless links between the client and RRH, but also optical fiber links between the RRH and BBU. An optical fiber link is considered to fail when a random probability ψ representing the state of the link is below an expected failure rate ψ_{fail} of the link. Consequently, the offloading failure conditions can be given by (6.12), where D_{CR}^{total} is the total delay incurred to complete a task for a client in C-RAN.

$$\left[(\forall\, v \in V, \quad P_v^{rx} < \tau_v) \vee \left(\psi < \psi_{fail}\right)\right] \vee \left(D_{CR}^{total} > \phi\right) \tag{6.12}$$

Table 6.1 lists the notations used in the analytical model and their definitions.

Table 6.1 Notations and Definitions

Notation	Definition
$PL_{u,v}$	path loss of link from node u to node v in dB
f	signal frequency
d_0	close-in free space reference distance in meters
c, $c_{(op)}$	propagation speed in free space, and optical fiber, respectively in meters per second
α	path loss exponent
$d_{u,v}$	distance between node u and node v in meters
X_σ	shadowing component in dB with standard deviation σ
$R_{u,v}$	data rate of link from node u to node v in bits per second (bps)
b	channel bandwidth
P_u^{tx}	transmitted signal power of node u in dBm
P_v^{rx}	received signal power of node v in dBm
P_v^{no}	average noise power at node v in dBm
T, T'	size of original, and processed task, respectively in bits
D_{OF}^{total}, D_{FR}^{total}, D_{CR}^{total}	total delay incurred in OF-RAN, F-RAN, and C-RAN, respectively in seconds
M_n, M_n'	size of original, and processed sub-task, respectively assigned to n_{th} service node in bits
N	number of service nodes in a v-FAP
η	number of clients
∂	size of request from OF-RAN/F-RAN client to RRH in bits
φ	size of notification from RRH to seed node/FAP in bits
γ	task complexity in floating-point operations (FLOPs) per bit
E_{OF}^{total}, E_{FR}^{total}, E_{CR}^{total}	total energy consumed in OF-RAN, F-RAN, and C-RAN, respectively in joules
E_{Sv}, E_F, E_B	processing energy efficiency of a service node in OF-RAN, FAP in F-RAN, and BBU in C-RAN, respectively in joules per cycle
ω_{OF}, ω_{FR}, ω_{CR}	computation intensity of OF-RAN, F-RAN, and C-RAN, respectively, in cycles per bit
δ_{Sv}, δ_F, δ_B	processor performance of service node, FAP, and BBU, respectively, in FLOPs per cycle per core
β_{Sv}, β_F, β_B	number of processor cores in a service node, FAP, and BBU, respectively

C_{Sv}, C_F, C_B	computation capacity of a service node, FAP, and BBU, respectively in FLOPs per second (FLOPS)
τ_v	receiver sensitivity of node v in dBm
ψ	probabilistic state of optical fiber link between RRH and BBU in C-RAN
ψ_{fail}	expected failure rate of optical fiber link between RRH and BBU in C-RAN
ϕ	task completion time requirement in seconds
ζ_{Sv}, ζ_F, ζ_B	processor frequency of service node, FAP, and BBU, respectively, in hertz (or cycles per second)

[1] Nodes u and v can be any transmitting node, and receiving node, respectively, in the RAN
[2] u or v can be replaced by C (client), S (seed node), Sv (service node), F (FAP), R (RRH), or B (BBU)

6.4. Simulation Environment

The analytical model developed in Section 6.3 is implemented in MATLAB to evaluate the offloading performance of all three RAN architectures under varying task complexity (γ) and number of clients (η). For the proposed OF-RAN, the impact of varying number of service nodes (N) in a v-FAP is also investigated. Table 6.2 lists the simulation parameters and their realistically chosen values based on the real-world devices or operation settings.

All the wireless links between RRH, clients, FAPs, seed nodes, and service nodes operate at 38 GHz, which is one of the 5G mmWave frequencies. Each wireless node pair communicates over a line-of-sight (LOS) channel with a path loss exponent slightly higher than 2 (free space path loss exponent) and a bandwidth of 500 MHz. The parameters $d_{u,v}$, T, M_n, and ϕ are assigned with random values uniformly distributed on a range as shown in Table 2. Unless otherwise specified, the default values of the following parameters are used: γ=6250; η=15, and N=4. All results are averaged over 100 simulations and their 95% confidence interval are shown when the margins of error are more than 5% of the mean value, as otherwise they are hardly visible in the graphs.

Table 6.1 Simulation Parameters

Parameter	Value	Unit
$P_C^{tx}, P_S^{tx}, P_{Sv}^{tx}, P_F^{tx}, P_R^{tx}$	30	dBm
$P_{R(op)}^{tx}, P_{B(op)}^{tx}$	2, 5	dBm
P_v^{no}	–82	dBm
τ_v	–79	dBm
$d_o, d_{u,v}$	1, 1–100	meter(s)
T, T'	52,000–68,000	bits
M_n, M_n'	13,000–17,000	bits
N	1–10	
γ	2,500–10,000	FLOPs/bit
η	1–30	
∂, φ	8,000	bits
σ	3.2	dB
f	38	GHz
b	500	MHz
α	2.05	
ψ_{fail}	0.998	
ϕ	5–50	milliseconds
E_{Sv}, E_F, E_B	$1\times10^{-10}, 5\times10^{-10}, 1.46\times10^{-8}$	joules/cycle
$\zeta_{Sv}, \zeta_F, \zeta_B$	2.4	GHz
$\beta_{Sv}, \beta_F, \beta_B$	1, 8, 24	
$\delta_{Sv}, \delta_F, \delta_B$	8, 16, 16	FLOPs/cycle
$c, c_{(op)}$	$3\times10^8, 2\times10^8$	meters/second
$R_{R,B}, R_{B,R}$	15–25	Gbps

6.5. Results and Discussion

6.5.1. Effect of varying N

Table 6.3 shows the OF-RAN performance in terms of the total delay, total energy consumption, and offloading failure under the effect of varying number of service nodes (N) in a v-FAP.

The results are obtained for a default η=15 clients and task complexity γ=6250 FLOPs per bit. The total failures are further broken down into link and completion time failures. In addition, their 95% confidence interval (CI) are shown as the calculated margins of error are mostly not negligible (> 5%).

It can be observed that the total delay decreases as N increases. This is because a larger N splits the service task into smaller sub-tasks, resulting in each service node to incur a smaller processing delay. Since all service nodes are processing in parallel, and the processing delay is more dominant than the transmission and propagation delays in the considered scenario, the total delay for servicing a client is largely dependent on the maximum processing delay among the service nodes in a v-FAP. Hence, increasing N decreases this maximum delay, which in tum decreases the total delay.

On the other hand, the total energy consumption is found to be relatively unaffected by N. This is again due to the total energy being dominated by processing energy over communication energy. The processing energy is dependent on the total service task size, i.e. sum of all sub-task sizes, the service node's energy efficiency and OF-RAN computation intensity, which do not change with N. The minute changes in total energy are attributed to small differences in communication energy caused by some randomness in the path loss and consequently data rate of the links between nodes.

Table 6.2 Effect of N on the OF-RAN performance (γ=6250, η=15)

N	Delay (ms)	Energy (mJ)	Failure (%)			
			Link failure	Completion time failure	Total	CI
1	25.68	71.91	4.34	33.13	37.47	± 1.46
2	20.83	72.02	7.40	20.13	27.53	± 1.94
3	16.89	72.29	10.27	12.13	22.40	± 2.22
4	13.45	72.07	12.07	6.13	18.20	± 2.27
5	11.14	72.13	17.47	3.53	21.00	± 2.21
6	9.486	72.23	19.66	2.47	22.13	± 2.46
7	8.02	72.36	21.73	0.27	22.00	± 2.39
8	7.069	72.19	22.93	0.13	23.07	± 2.70
9	6.283	72.18	26.46	0.07	26.53	± 2.70
10	5.612	72.04	29.27	0	29.27	± 2.51

However, N has a significant impact on the type of failure occurrence. As seen in Table 6.3, increasing N increases the proportion of link failures, but decreases that of completion time failures. The reason is that a higher N increases the number of links but decreases the total delay that in turn reduces the number of completion time failure. The total failure rate is minimized when N=4, which explains our choice of setting the default number of service nodes in a v-FAP to this value.

Since the service nodes are only used in OF-RAN, we do not evaluate the effect of N on other types of RAN. In the next two sections, we further evaluate the performance of OF-RAN under the effect of varying task complexity γ and number of clients η, and compare it with the performances of current F-RAN and C-RAN architectures.

6.5.2. Effect of varying γ

Figure 6.3 and Figure 6.4 show the total delay, and total energy consumption, respectively, of all three RAN architectures under varying task complexity (γ). Results are obtained under a default number of clients (η=15) for an average scenario, and a large number of clients (η=30) for a stress scenario. The 95% confidence interval of the results are found to have a margin of error between 0.2–3.3%, which are hardly visible and thus omitted in the graphs.

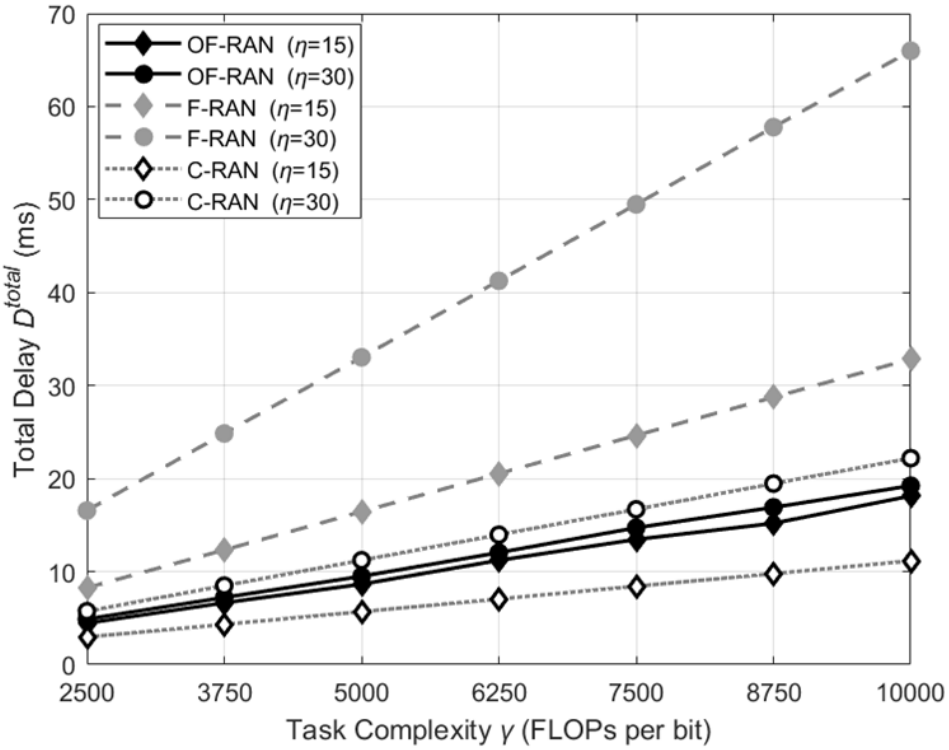

Figure 6.3 Effect of γ on total delay of the RANs.

Expectedly, both delay and energy consumption increase with γ, as higher-complexity tasks demand more processing time and energy. In Figure 6.3, it can be seen that C-RAN incurs the least delay in the average case (η=15), followed by OF-RAN and F-RAN. However, in the higher- stress case (η=30), the OF-RAN outperforms both C-RAN and F-RAN. This is because the computation capacity available to each client in C-RAN and F-RAN decreases with higherη due to the finite fixed capacity of the BBU, and FAP, respectively. On the contrary, OF-RAN can expand its computation capacity when needed by establishing more v-FAPs (one for each

new client) subject to the service nodes availability. This illustrates the inherent scalability of the OF-RAN architecture.

Fig. 6.4 shows that OF-RAN outperforms C-RAN and F-RAN in total energy consumption for both average and stress scenarios. This is despite the OF-RAN utilizing more nodes, i.e. service nodes, which can lead to higher communication energy consumption. The reason is due to OF-RAN's much lower consumption of processing energy, which dominates the total energy consumption. While the BBU and FAP (processing nodes in C-RAN, and F-RAN, respectively) have higher computation capacity, they are also more power-hungry and consume more energy per CPU cycle. On the other hand, being often battery-powered user devices, OF-RAN's service nodes are operating with better processing energy efficiency or less energy in joules per cycle.

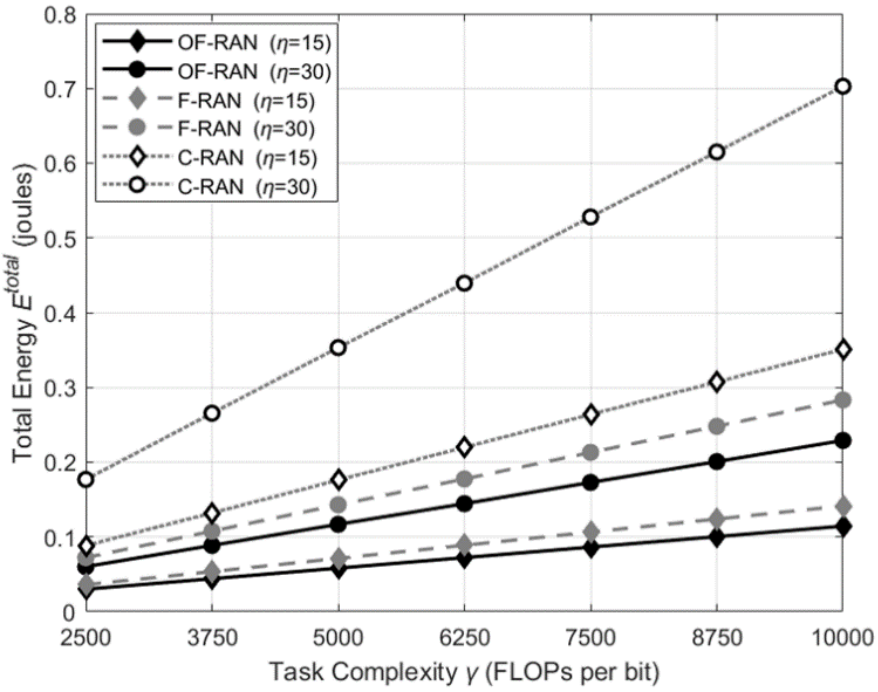

Figure 6.4 Effect of γ on total energy consumption of the RANs.

Figure 6.5 further shows impact of task complexity on the offloading failure, which is broken down into link and completion time failures. The results are shown for the stress scenario (η=30) with their 95% confidence interval as the margins of error are not negligible (> 5%).

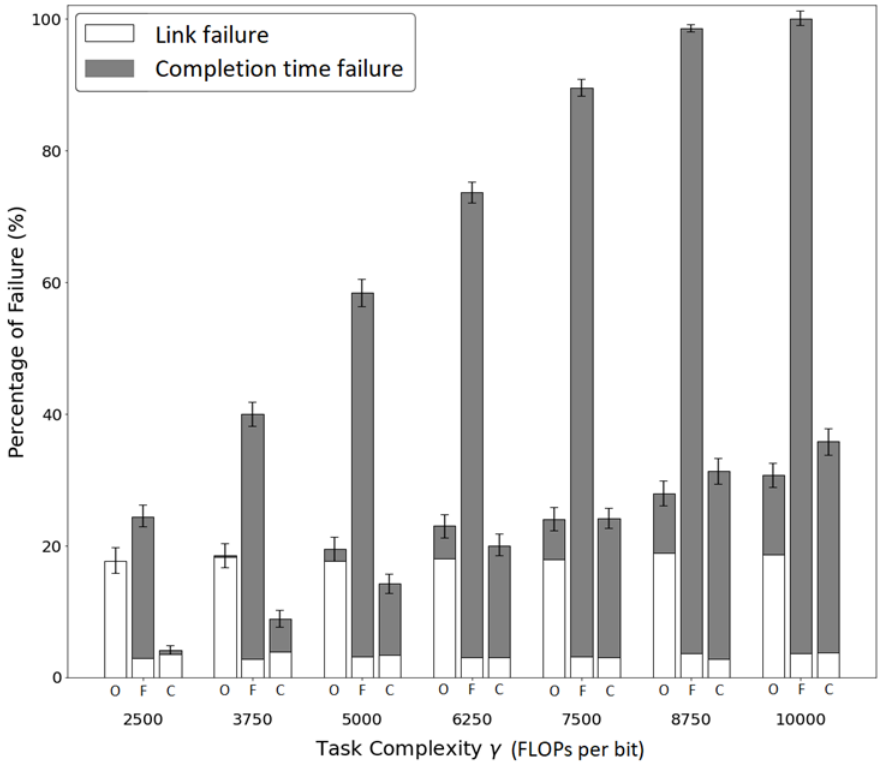

Figure 6.5 Effect of γ on failure rate of the OF-RAN (O), F-RAN (F) and C-RAN (C).

Expectedly, the failure rate of all RANs increases with task complexity, caused by an increase in completion time failures due to longer processing time. At low task complexity (γ=2500), C-RAN has the lowest failure rate, followed by OF-RAN and F-RAN. Both failures in C-RAN and OF-RAN are mainly due to link failures. However, as task complexity increases to γ=7500, OF-RAN begins to outperform as the number of its completion time failures increases at a slower rate than C-RAN and F-RAN. This is consistent with the observation in Fig. 3 where the delay of OF-RAN (predominantly processing delay) increases at a slower rate than C-RAN and F-RAN under the stress scenario. This illustrates once again that OF-RAN is better suited for stress scenarios with not only high number of clients but also high task complexity.

6.5.3. Effect of varying η

In this section, we present a more detailed analysis on the effect of varying number of clients (η) under average and stress scenarios defined by task complexity. Figure 6.6 and Figure 6.7 show the total delay, and total energy consumption, respectively, of all three RAN architectures, as η varies from 1 to 30. Results are obtained under a default task complexity (γ=6250) for an average scenario, and high task complexity (γ=1000 0) for a stress scenario. The corresponding 95% confidence intervals have a margin of error between 0.15–4.3%, which are again hardly visible and thus omitted in the graphs.

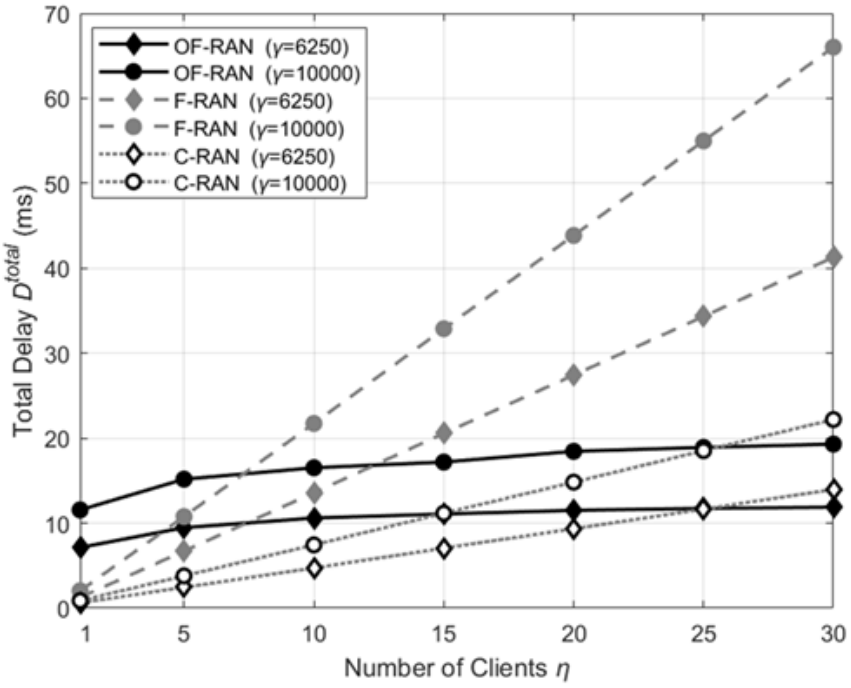

Figure 6.6 Effect of η on total delay of the RANs.

For low number of clients ($\eta \leq 10$), C-RAN has the lowest delay in both average and stress scenarios, which is attributed to its high computation capacity, resulting in much smaller processing time that dominates the total delay. F-RAN has a lower initial delay than OF-RAN, but it increases with η at a rate faster than OF-RAN and C-RAN. OF-RAN starts to outperform F-RAN at η=10, and then C-RAN at η=30, in both average and stress cases. Moreover, it exhibits a relatively flat delay

response to η, due to its ability to expand computation capacity when needed as explained in previous section. In terms of energy, OF-RAN consistently consumes the least for all η and in both average and stress cases. On the other hand, C-RAN consistently consumes the most, mainly due to its power-hungry BBUs that result in high processing energy consumption.

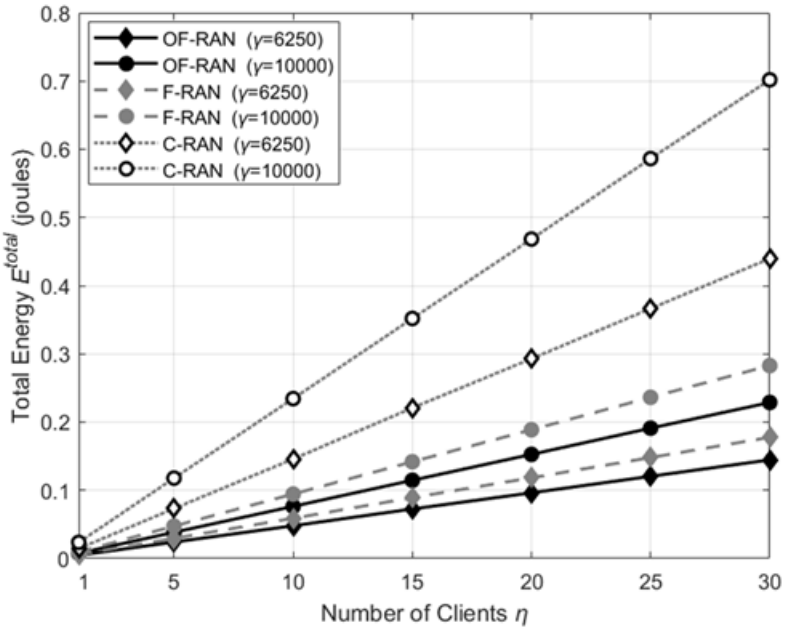

Figure 6.7 Effect of η on total energy consumption of the RANs.

Figure 6.8 shows the impact of η on the offloading failure. The results are shown for stress scenario (γ=10000) with their 95% confidence interval as the margins of error are non-trivial (> 5%). Similar to the delay result, the failure in OF-RAN is relatively unaffected by η. Moreover, it starts to outperform F-RAN at η=10, and then C-RAN at η=30. On the other hand, the failure in C-RAN and F-RAN increase with η due to more completion time failures. This is because a higher η reduces the computation capacity available to each client in these RANs, and the impact is greater on F-RAN since FAPs are not as computationally powerful as BBUs in C-RAN. Overall, the results show that the OF-RAN is a promising and scalable architecture.

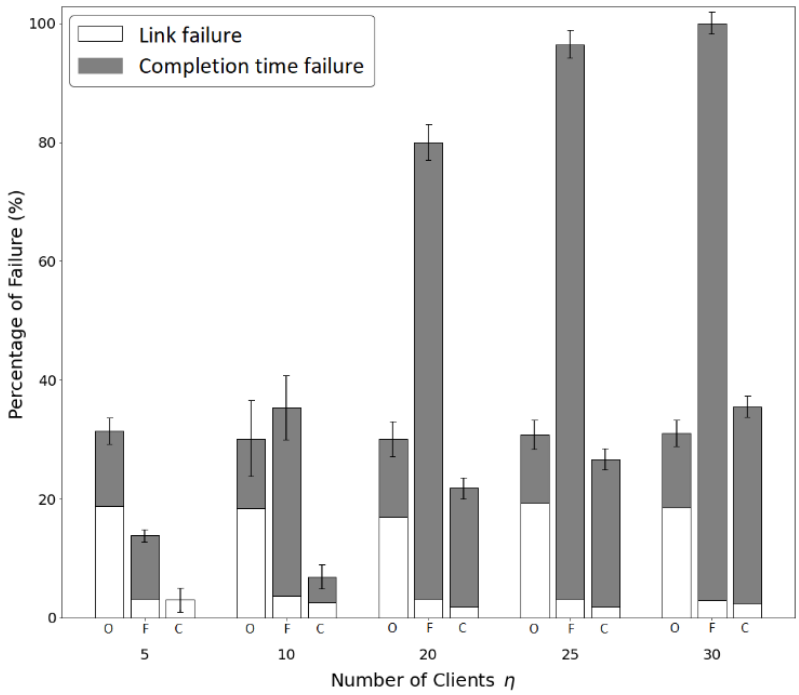

Figure 6.8 Effect of η on failure rate of the OF-RAN (O), F-RAN (F) and C-RAN (C).

6.6. Chapter Summary

This chapter analyses and compares the offloading performance of OF-RAN with that of existing C-RAN and F-RAN. For each RAN, we develop an analytical model to evaluate its offloading performance in terms of completion delay, energy consumption, and failure rate. The performances are evaluated under the effect of varying number of service nodes, number of clients, and task complexity.

The results show that there exist an optimal number of service nodes for which the failure rate of OF-RAN is minimized. OF-RAN also outperforms C-RAN and F-RAN in all three performance metrics under high-stress scenarios where the task complexity and number of clients are high. This illustrates the scalability of our OF-RAN, which can co-exist with and complement the C-RAN and F-RAN to support computation-intensive and delay-sensitive offloading services.

CHAPTER VII: Blockchain Enabled OF-RAN: Deep Learning Applications Case Study

7.1. Introduction

The exponential rise in data volume from a wide range of IoT devices has driven the development of DL based applications where big data is used for learning and training purposes. This has resulted in improved accuracy of various DL applications such as object detection, image recognition, and speech analysis. However, DL is computationally resource intensive if executed in IoT devices with low computation capacity. Hence, the need for them to offload DL tasks to more resourceful devices has increased significantly. Traditional offloading to the cloud via a C-RAN can be a potential solution, but it suffers from several drawbacks such as heavy workload at centralized baseband unit pool BBU, high data leakage, limited backhaul capacity, and difficulty in serving delay sensitive applications [19,98]. To resolve some of the above-mentioned issues, researchers have proposed F-RAN in which fog access points (FAPs) are deployed at network edge to serve the IoT devices. These FAPs can be entities already exist in an infrastructure but additionally equipped with fog functionalities, or newly deployed entities such as picocell or femtocell base stations in an existing infrastructure [4]. However, existing F-RANs does not utilize distributed edge devices in the proximity of FAPs such as femtocell BS, Wi-Fi access points and resource rich end-user devices.

Alternatively, the DL tasks can be offloaded to an opportunistic fog radio access network (OF-RAN) which we proposed in [26]. OF-RAN enhances the F-RAN by harnessing the concept of oppnets, which are a type of adhoc networks for utilizing available local resources in an opportunistic manner [8]. Each oppnet is established by a 'seed node', which assigns one or more 'helper nodes' to assist with a specific task. In our proposed OF-RAN, the *seed node* and

service node are equivalent to the FAP of F-RAN, and helper nodes of oppnet respectively. A seed node in OF-RAN recruits locally available resourceful user devices such as high-end smartphones and tablets as service nodes that collectively form a virtual FAP (v-FAP) to serve a resource-limited client such as an IoT device [26,84]. The resourceful user devices can be remunerated based on their performance in serving the client in terms of timeliness and reliability. However, the proposed OF-RAN still requires an effective way of managing service nodes and establishing trust between various entities in our scenario, such that we can automate the formation of V-FAPs involving multiple seed nodes and service nodes in a large-scale OF-RAN. Recently, blockchain has emerged as a promising candidate to provide distributed and secure solutions, along with the features of smart contacts for IoT automation. Consequently, we further proposed a blockchain-enabled OF-RAN [92] where each seed node has additional mining features. Such a seed node is thus also a *blockchain node*, which has a copy of the blockchain and collectively forms a blockchain network with the other seed nodes as shown in Figure 7.1.

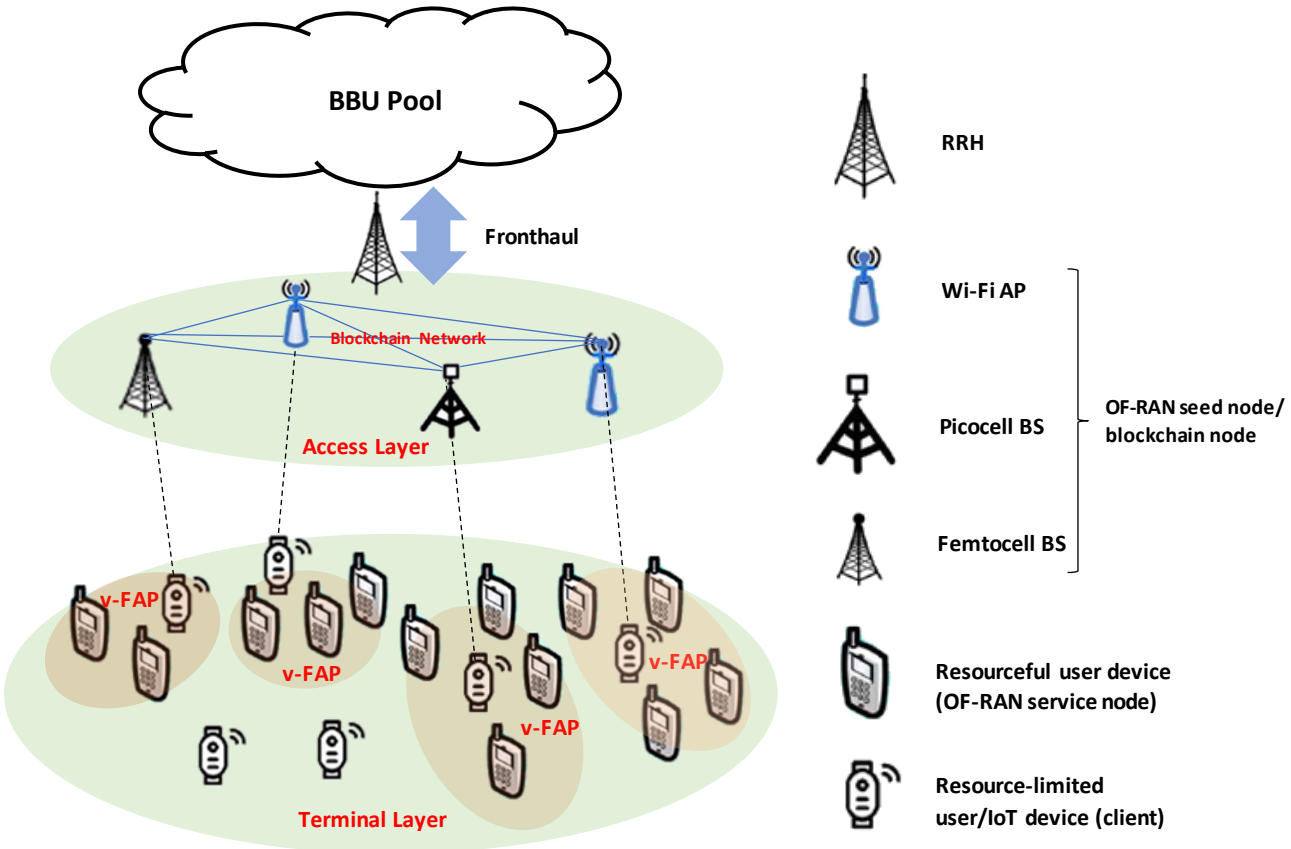

Figure 7.1 System architecture of blockchain-enabled OF-RAN

Here, the smart contract residing in each blockchain node is utilised for automated formation of v-FAP. Each block in the blockchain is composed of a lookup table that keeps information about the identity and performance of each service node in a v-FAP. During the operation of a v-FAP, the seed node mines a new block from the lookup table and sends it to the blockchain network as proof-of-work (PoW) before being appended to the blockchain. This information in the blockchain along with our proposed task-node assignment algorithm implemented as a smart contract script for OF-RAN [91] will facilitate the seed nodes in selecting their service nodes for the future formation of v-FAPs. In v-FAP formation, the blockchain node is mainly responsible for executing the smart contract scripts and helping in the selection of reliable service nodes using a lookup table. The computations performed in the v-FAP for an offloading application is independent of the blockchain computation performed by the seed node. This is to ensure the blockchain-enabled OF-RAN can still support delay-sensitive applications.

Federated learning is a promising new paradigm for distributed machine learning where DL models can be executed locally in a distributed manner and the results are sent to a server for aggregation [60]. To demonstrate how our OF-RAN can play a vital role in real life applications, we have designed and implemented a federated DL application. This federated DL approach can be incorporated into our blockchain-enabled OF-RAN, where a resource-limited client such as an IoT device can send its DL model parameters and training data to the nearest seed node. In turn, the seed node splits the training data and forwards them along with the model parameters to each service node in its v-FAP. The service nodes then train their respective local models and send the trained model parameters to the seed node for aggregation before returning the aggregated results to the IoT devices.

The key research contributions of this chapter are:

- Propose and investigate the use of our blockchain-enabled OF-RAN architecture to support federated DL for resource-limited IoT devices.
- Design and implement algorithms for federated DL and block generation by v-FAP nodes of our blockchain-enabled OF-RAN
- Build a v-FAP testbed using Raspberry PI devices to experimentally evaluate our algorithms and demonstrate the feasibility of our architecture.

The rest of the chapter is organized as follows. Section 7.2 presents the system model of the blockchain-enabled OF-RAN architecture. Section 7.3 details the process design for federated DL and block generation in the considered architecture. The experimental and simulation environments, and the results are discussed in Section 7.4, and 7.5, respectively. Finally, Section 7.6 concludes the chapter.

7.2. System Model

The system architecture proposed in this chapter describes a potential novel approach for decentralized big data analysis wherein the learning task is performed at the local models of service nodes, and the aggregation of model parameters and mining of a new block from updated lookup table is performed at the seed node. Figure 7.2 shows the functional block diagram of the proposed Blockchain-enabled OF-RAN in which the service nodes in a V-FAP interact with their associated seed node to carry out federated and secured DL tasks. The green arrows indicate the flow of operations between the seed node and service nodes. The individual entities used in the block diagram are explained as follow:

i. *Local Model*: Each service node in a v-FAP prepares a local learning model based on the initial parameters obtained from the seed node for a given client.
ii. *Service Task*: Refers to the DL task assigned to each service node.

iii. *Smart Contract*: Defines all the policies and rules for operating and governing our blockchain enabled OF-RAN. It has two main modules: learning and mining contracts. When the learning contract is initialized, each service node receives the initial parameters of its local model, trains it and then sends the updated local parameters to the seed node for aggregation. The mining contract is responsible for the formation of new blocks to update the lookup table information in the blockchain after the learning contract is executed.

iv. *Smart Contract Interface*: Connects service nodes and seed node to the smart contract. It automatically triggers smart contract operations and activities at the v-FAP, which includes communication between seed and service nodes, and sharing the initial model parameters of the client with the service nodes via the seed node. It uses secured shell interface (SSH) to secure its connectivity with the service nodes and seed node in a v-FAP.

v. *Blockchain Network*: Every seed node participating in our OF-RAN is also a blockchain node, and they collectively form a blockchain network. Each blockchain node is responsible for monitoring all task transactions between the nodes in a v-FAP. Whenever a new v-FAP is formed, a new block is created and propagated in the blockchain network as PoW [96].

vi. *Iteration*: Refers to a set of steps performed by the service nodes to obtain an effective local deep learning model.

vii. *Epoch*: Refers to a single operation of generating a new block by the seed node in blockchain.

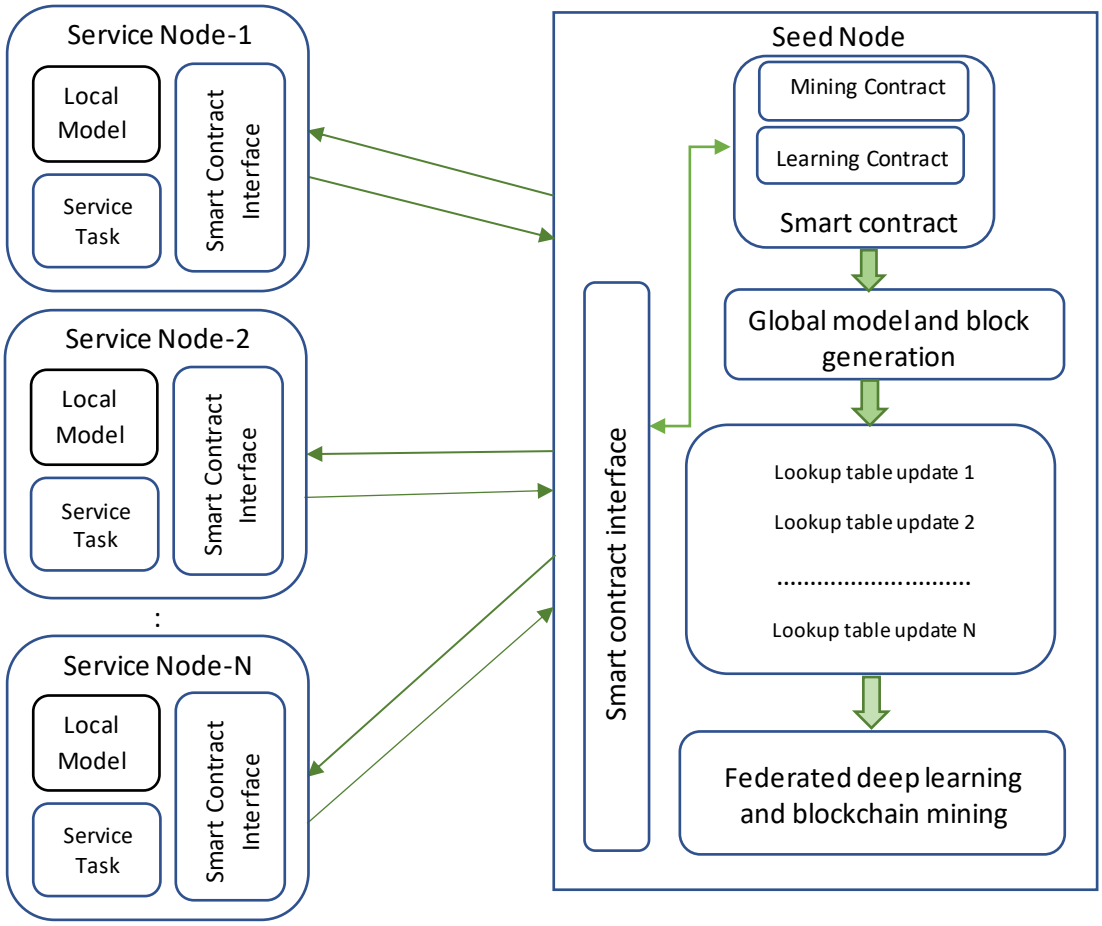

Figure 7.2 Functional Block Diagram of Blockchain-enabled OF-RAN

7.3. Proposed Federated DL and Blockchain Processes For OF-RAN

For a resource-limited client to offload its DL task to the proposed OF-RAN as shown in Figure 7.3, it first sends a service request {1} including the task requirements to its associated remote radio head (RRH), which in turn notifies the client {2} of an available nearby seed node to offload its task. The client then offloads its task {3} to this seed node in the form of data and initial model parameters. The seed node splits the data into mini batches for each service node in the v-FAP. Based on the task-node assignment (TNA) scheme proposed in our earlier work [91], the seed node distributes the mini batches and initial model parameters {4} to the service nodes for creating and training their respective local DL model. Upon training, the service nodes upload their updated model parameters {5} to the seed node for aggregation. Finally, the seed node returns the aggregated trained model parameters {6} to the client.

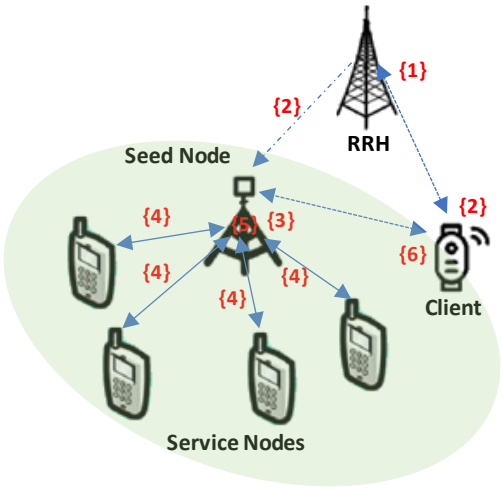

Figure 7.3 The sequence of operations in proposed Blockchain-enabled OF-RAN

Once the DL task is accomplished, the seed node updates its lookup table with new information about the identity and performance of each service node in its v-FAP, mines a new block from the updated information, and transmits it as PoW for appending to the blockchain. In the following sections, we explain in detail the federated DL process and its deployment in a blockchain environment. We also discuss the delay and accuracy invoked by federated DL and parameter impact of the blocks created by the seed nodes in our blockchain network.

7.3.1 Federated DL process

Federated DL using service nodes in a v-FAP relies on collection of their model's weight w parameters obtained by learning on training data sets. A training data sample i is described as a two-dimensional coordinate (x_i, y_i), wherein the DL model takes vector x_i as an input (such as pixels of an image) and gives a scalar output y_i (such as the label of the image). For each training data sample, the DL model computes a loss function $f_i(w)$, the result of which indicates the extent of model errors on the training data samples, and thus should be minimized in the learning process.

We consider a seed node with N resourceful user devices in its neighborhood that can be recruited as service nodes, and M is the set of training data from client. Denote the set of service

nodes in a v-FAP as $S = \{s_1, s_2, \ldots s_n\}$ where $n \in N$. Each s_j, $j = \{1..n\}$, receives a subset of training data m_j from the seed node, and $M = \sum_{j=1}^{n} m_j$. A loss function for s_j over its training data m_j can be defined as:

$$F_j(w) \triangleq \frac{1}{|m_j|} \sum_{i \epsilon m_j} f_i(w) \tag{7.1}$$

where $|m_j|$ returns the size of m_j. The global loss function for a v-FAP can be further defined as:

$$F(w) \triangleq \frac{\sum_{j=1}^{n} F_j(w)}{|M|} \tag{7.2}$$

The goal of the DL task is to find the optimal weight w' parameters that minimize $F(w)$:

$$w' \triangleq \arg\min F(w) \tag{7.3}$$

Denote $w_j^{(t)}$ as the local model parameters of each service node s_j at iteration t of learning. Here, $t = 0, 1, 2, .., T$, where T is the maximum number of iterations. Each s_j trains its local DL model using the subset of training data m_j. At $t = 0$, all service nodes in S initialize their local model parameters. At each subsequent iteration $t > 0$, each s_j updates its $w_j^{(t)}$ by minimizing the loss function using the gradient descent update rule in (7.4), where $\lambda > 0$ is the learning rate and $\nabla F_j(w_j^{(t-1)})$ is the average gradient on its training data at the previous local model parameters $w_j^{(t-1)}$:

$$w_j^{(t)} = w_j^{(t-1)} - \lambda \nabla F_j(w_j^{(t-1)}) \tag{7.4}$$

After T iterations, the updated local model parameters from each service node s_j is sent to the seed node where it is aggregated as a global model update, which in turn is sent once every P

epochs. For each epoch, a total of T iterations of local update are performed at each s_j. The local update of s_j at epoch p and iteration t is given by:

$$w_j^{(t,p)} = w_j^{(t-1,p)} - \frac{\lambda}{m_j}\left(\left[\nabla F_j\left(w_j^{(t-1,p)}\right) - \nabla F_j\left(w_j^{(p-1)}\right)\right] + \nabla F\left[w^{(p-1)}\right]\right) \quad (7.5)$$

where $p = 1,2,..P$, $w^{(p)}$ is the global update at epoch p, and $\nabla F\left(w^{(p)}\right) = \frac{1}{|M|}\sum_{j=1}^{n} m_j \nabla F_j\left(w^{(p)}\right)$ is the global gradient value at epoch p after T iterations are performed. Let $w_j^{(p)}$ be the local update of service node s_j at epoch p after T iterations. Then w^p is updated as:

$$w^{(p)} = w^{(p-1)} + \frac{1}{|M|}\sum_{j=1}^{n} m_j (w_j^{(p)} - w^{(p-1)}) \quad (7.6)$$

The federated DL process is detailed in Algorithm 7.1, where T iterations of local update are performed in each epoch p for all service node s_j in S. The final output of this process is $w^{(f)}$, which gives the final model update or parameter that produces a minimum value of global loss over an entire execution of local and global updates.

Algorithm 7.1: Federated DL process in OF-RAN

1. **Input**:
2. $S = \{s_1, s_2, .. s_n\}$: set of service nodes associated with a seed node
3. T : total number of iterations in an epoch
4. P: total number of epochs
5. ε: termination threshold
6. w_{cl}: initial weights from client
7. **Output**: final model update or parameter w^f
8. **Process**: initialize model parameters w^f, $w_j^{(t=0,p=1)}$, $w_j^{(p=0)} \leftarrow w_{cl}$ for all service nodes s_j
9. set $t \leftarrow 0$; $p \leftarrow 1$
10. while $(|w^f| - |w^{(p-1)}| \leq \varepsilon)$ do
11. set $t \leftarrow t + 1$
12. for each service node s_j in S do
13. compute local update $w_j^{(t,p)}$ using equation (7.5)
14. end for
15. if t is an integer multiple of T

16. set $w_j^{(p)} \leftarrow w_j^{(T,p)}$ for each s_j in S
17. compute global update $w^{(p)}$ using equation (7.6)
18. update $w^f \leftarrow \underset{w \in \{w^f, w^{(p)}\}}{\arg\min} F(w)$
19. set $p \leftarrow p + 1$
20. end if
21. end while

7.3.2. Blockchain Process in OF-RAN Architecture

To provide a secure and reliable exchange of model parameters between seed and service nodes in a v-FAP via the distributed ledger, the blockchain enabled v-FAP deploys federated DL in the blockchain network wherein the transactions verification and block generation are performed by the seed node. The transaction $Tx_j^{(p)}$ received by the seed node from each service node s_j in S during epoch p records the service node's updated local weight $w_j^{(p)}$, and task completion time $t_j^{(p)}$. The latter is also utilized as a performance indicator by the seed node for service node selection in future v-FAP formation. Each block generated by the seed node comprises a body and header. The body contains a subset of the transactions containing the task completion time $t^{(p)} = \{t_1^{(p)}, t_2^{(p)}, .. t_n^{(p)}\}$ received from each service node s_j in S and stored in the lookup table of the seed node. The header H contains information about the seed node's block generation rate β (a.k.a. block interval time) and a hash pointer φ to the previous block.

A *learning contract* is executed by the seed node through which each service node s_j iteratively trains its local model. At each epoch p, the service node envelops its updated local weight $w_j^{(p)}$ and recorded completion time $t_j^{(p)}$ into transaction $Tx_j^{(p)}$, which is then sent to the seed node.

Upon reception, a *mining contact* is executed through which the seed node verifies the transactions, extracts and aggregates the local weights $\{w_1^{(p)}, w_2^{(p)}, .. w_n^{(p)}\}$ to compute the global update $w^{(p)}$, as well as completion times $\left\{t_1^{(p)}, t_2^{(p)}, .. t_n^{(p)}\right\}$ to generate a new block

$B^{(p)} = \{t^{(p)}, \beta, \varphi\}$ for appending to the blockchain. Hence, the learning and mining contracts (collectively known as *smart contracts*) are iteratively executed to carry out federated DL and update the blockchain at each epoch in our blockchain-enabled OF-RAN for DL applications, as illustrated in Algorithm 7.2.

Algorithm 7.2: Smart Contracts for DL and Blockchain Processes in OF-RAN

1. **Learning Contract**: DL process
2. for each service node s_j in S do
3. envelop $Tx_j^{(p)} \leftarrow \left[w_j^{(p)}, t_j^{(p)}\right]$
4. upload $Tx_j^{(p)}$ to seed node
5. end for
6. **Mining Contract**: Blockchain process
7. receive $Tx^{(p)} = \left\{Tx_1^{(p)}, Tx_2^{(p)}, .. Tx_n^{(p)}\right\}$
8. for each $Tx_j^{(p)}$ in $Tx^{(p)}$ do
9. extract $w_j^{(p)}$ and $t_j^{(p)}$ from $Tx_j^{(p)}$
10. verify $w_j^{(p)}$
11. add $t_j^{(p)}$ to $t^{(p)}$
12. end for
13. compute global update $w^{(p)}$ using equation (7.6)
14. generate block $B^{(p)} \leftarrow \{t^{(p)}, \beta, \varphi\}$

7.4. Experimental and Simulation Environment

7.4.1. Experimental Environment

This section describes the experimental environment to evaluate our proposed system, including the emulation of a v-FAP using Raspberry Pi devices, and the implementation of federated learning and smart contracts for an object detection application used as a case study.

The v-FAP is emulated using Raspberry Pi 4 Model B single-board computers (4 GB RAM, 1.5 GHz CPU, Raspbian OS) as service nodes, and an Acer Aspire F15 laptop (8 GB RAM, 2.5 GHz CPU) as the seed node. The latter is configured as a WiFi hotspot for it to communicate with the service nodes. Python 3.7 and TensorFlow 2.3.0 are used for implementing the DL models in both seed and service nodes. Python is also used for implementing the smart

contracts in the seed node, which include learning and mining contracts. The secure communications between seed and service nodes are implemented using Parallel SSH protocol. Figure 7.4 shows one seed node (laptop) with four service nodes (Raspberry Pis) to emulate a v-FAP to perform offloaded DL tasks for an object detection application.

Object detection has wide applications in IoT such as for surveillance and crowd control, which require low latency and high accuracy. In this work, we use the MNIST dataset, which contains 33600 training instances and 8400 validation instances of 10 object classes. The whole training dataset is split into 1050 mini batches of batch size 32 (total dataset of 33600 rows × 785 columns is split into minibatches of 32 rows × 785 columns). The minibatches are equally divided among the service nodes in a v-FAP. The DL model of each service node is implemented as a Deep Neural Network (DNN) with 7 layers: 1 input layer, 5 hidden layers, and 1 output layer. Each hidden layer can have a number of neurons, and neurons at different layers can learn the hierarchical features of input training data, which represent different levels of abstraction. Each neuron has multiple inputs and one single output. Generally, the output of neuron i at layer $l-1$ connects to each input of neuron j at layer l. For the connection between two neurons, there is a weight assigned to it. For example, $w_{i,j}$ is a weight assigned to the connection between neuron i at layer at $l-1$ and neuron j at layer l. Each neuron i has a bias b_i. These weights and bias are model parameters that need to be learned during training. Each service node trains its own DNN model and then sends the trained weights and bias to the seed node.

Figure 7.4 Emulated v-FAP for offloaded DL tasks

7.4.2. Simulation Environment

To investigate the impact of the blocks generated by seed node on the blockchain network, we use the Bitcoin simulator [96] built on discrete-event network simulator NS-3 to simulate a blockchain network with realistic download speed, latency, and bandwidth distribution. This is a PoW based simulator with a set of consensus and network parameters such as network delays, block generation time, and block size. The main outputs of this simulator are stale (orphan) block rate and throughput. We use the obtained results from the simulator to determine an appropriate block size and block interval for our blockchain to balance a trade-off between security and throughput. The following defines some key terms used in both our experiment and simulation:

- *PoW mechanism*: This is the widest deployed consensus mechanism in existing blockchains where each mining node uses its computing power to solve the proof-of-work instance and construct the appropriate block. It entails finding a nonce value (an arbitrary number used

once in cryptographic communication) such that when hashed with extra block parameters, the hash value must be smaller than the target value. When such a nonce is formed, the miner creates a block and forwards it to its peers who can verify the PoW by computing the hash of the block and check whether it satisfies the condition to be smaller than the current target value.

- *Block interval*: The time interval between blocks being added to the blockchain. The smaller the interval, the higher the probability of stale blocks. In this work, the block interval depends on the average time incurred by all service nodes to compute and send their transactions to the seed node for a new block to be generated and updated.
- *Block size*: Depends on the number of transactions carried within a block. This size controls the throughput attained by the system. Large blocks incur slower propagation speed which in turn increases stale block rate and weaken the security of blockchain [96].
- *Stale block*: Refers to a block not included in blockchain due to concurrency or conflicts between miners. It triggers chain forks which is an inconsistent state that slows the growth of main chain and thus is detrimental to the blockchain security and performance.
- *Stale block rate*: The percentage of stale blocks among the total number of blocks mined.
- *Throughput*: The ratio of number of transactions in a block to block interval time [97]. This metric is expressed in units of transactions per second (tps).
- *Mean precision accuracy (MPA)*: The ratio or percentage of correctly predicted test instances to total number of test data instances using the global model for object detection. In this work, we evaluate the impact of increasing number of service nodes on this accuracy.
- *Latency*: The total time incurred (including computation and communication time) for one epoch operation of the federated DL process. Similarly, we evaluate the impact of increasing number of service nodes on this latency.

Table 7.1 shows the simulation parameter settings used in our blockchain simulation. The upper limit of block size is set to 4 MB due to a study in [96] that found larger block sizes led to higher stale block rates that degraded the network security. The block interval range of 2–7 minutes is selected based on our experimental findings on the time incurred by our emulated v-FAP with 1–4 service nodes. The transaction size is approximately the average size of information sent from our service nodes to seed node. The number of miners is the number of seed nodes in our OF-RAN, and the Bitcoin simulator requires a minimum of 16 miners to be configured.

Table 7.1 Simulation Parameter Settings

Parameters	Values	Unit
Block size	0.25–4	MB
Block interval	2–7	minutes
Transaction size	1	KB
Number of miners	16	

7.5. Results and Discussion

7.5.1. Effect of Varying Service Nodes

The number of service nodes N in a v-FAP can potentially impact the performance of the offloaded DL tasks for the object detection application. Table 7.1 shows the experimental results obtained from our emulated v-FAP in terms of the latency and mean precision accuracy (MPA) as defined in Section 7.5.2 under varying number of service nodes. The results are obtained for one epoch in which the maximum number of DL iterations is equal to the number of mini-batches.

It can be observed that the latency decreases as N increases. This is firstly because higher N splits the training data into smaller mini-batches, resulting in each service node to incur smaller computation delay. Furthermore, since the learning task in all service nodes are executed in parallel, increasing N decreases the maximum delay among service nodes in the v-FAP. It is

also observed that the overall latency is significantly dominated by computation rather than communication delay.

On the other hand, the MPA is determined in the seed node using the global model obtained after the aggregation of local updates from each service node in the v-FAP. The results show that as N increases, the MPA expectedly decreases but only marginally, which can be attributed to the reduced number of mini-batches per service node. Hence, we can infer that the MPA of the global model depends on the number of mini-batches used for training the local model of the service nodes.

There is inherently a trade-off between latency and accuracy of the global model. The seed node can thus select the number of service nodes in a v-FAP based on the requirements of the offloading client (e.g. an IoT device). For instance, when the learning task is delay sensitive, the seed node may utilize more service nodes to reduce latency. On the other hand, if high accuracy is more critical than low latency, the seed node can opt for forming a v-FAP with fewer number of service nodes.

Table 7.2 Experimental performance of offloaded DL tasks under varying number of service nodes (N)

N	No. of mini-batches	Latency (secs)	MPA (%)
1	1050	408.85	94.84
2	525	220.92	93.70
3	350	161.86	92.19
4	262	139.06	91.33

7.5.2. Effect of Varying Block size and Block Interval

The choice of block size and block interval can potentially impact the blockchain performance of our OF-RAN in terms of stale block rate and throughput. Their choices of settings are in tum further factors to consider when determining the appropriate number of service nodes in a v-FAP.

Figure 7.5 shows the stale block rate and throughput under varying block size δ. The results are obtained for a default block interval $\tau = 4.5$ mins. It can be seen that as the block size increases, the throughput increases. However, the stale block rate also increases, which can cause the blockchain to be more easily attacked, thus increasing its level of insecurity.

Similarly, Figure 7.6 shows the stale block rate and throughput under varying block interval and default block size $\delta = 2$ MB. The results show that as block interval increases, the stale block rate decreases, but throughput also decreases. Hence, we can make appropriate choices of δ and τ that meet our stale block rate and throughput requirements. Moreover, the latency incurred by the v-FAP shall set the lower limit of τ, since the seed node cannot generate a block before all transactions are received from its service nodes. Alternatively, the chosen δ and τ that meet the stale block rate and throughput requirements shall inform the choice of an appropriate number of service nodes N that should also meet the offloading client's latency and MPA requirements.

For instance, if the required stale block rate, throughput, latency, and MPA are ≤ 1%, ≥ 10 tps, ≤ 200 secs (3.33 mins), and ≥ 92%, respectively, then an appropriate choice of parameters for our system could be: $\delta = 2$ MB; $\tau = 3$ mins; and $N = 3$, for a resulting stale block rate of 0.8%, throughput of 11.5 tps, latency of 161.86 secs (2.81 mins), and MPA of 92.19% (refer to Figure 7.6 and Table 7.2).

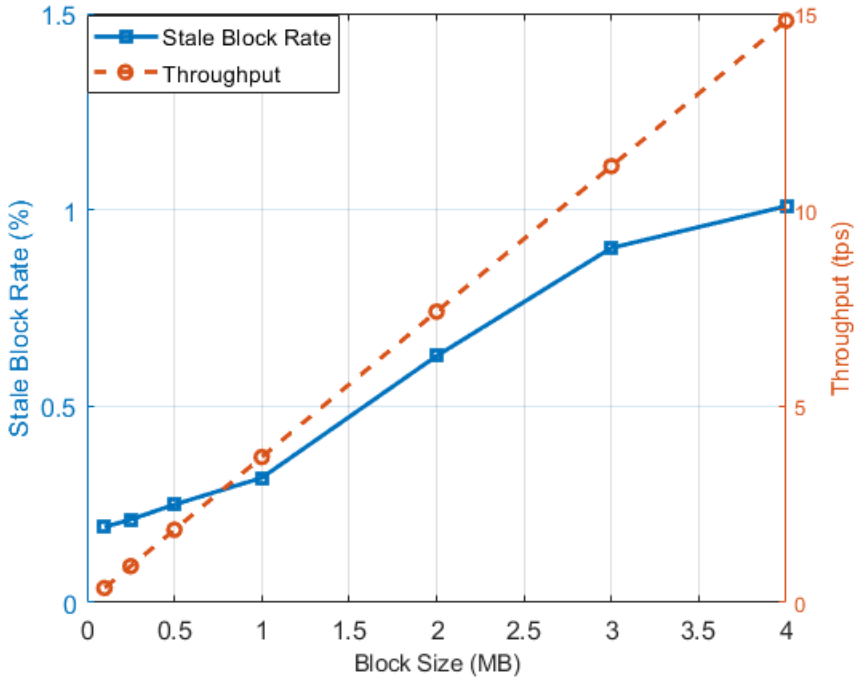

Figure 7.5 Effect of block size δ on stale block rate and throughput (τ=4.5 mins)

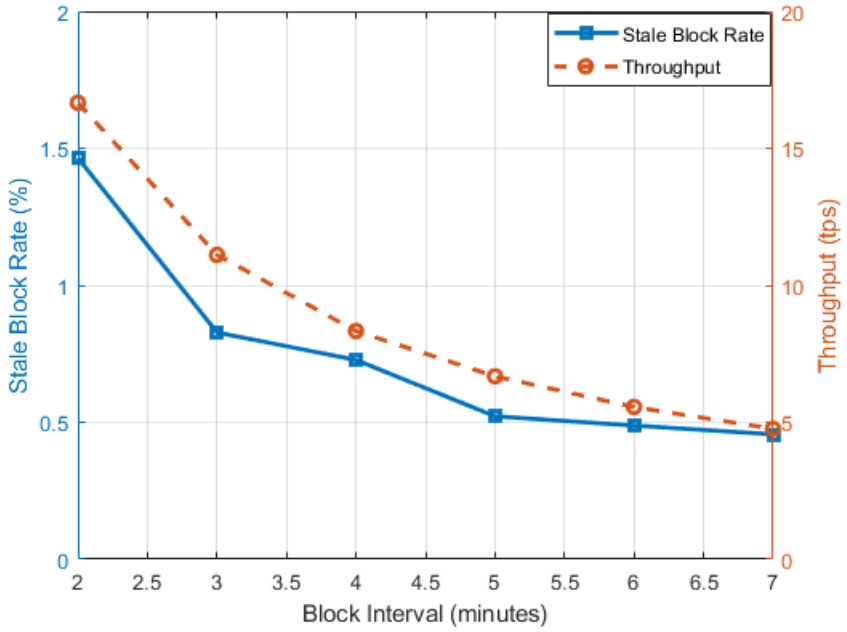

Figure 7.6 Effect of block interval τ on stale block rate and throughput (δ=2 MB)

7.6. Chapter Summary

This chapter proposes a blockchain-enabled OF-RAN for DL applications, which enables resource-limited devices such as IoT devices to offload their computationally intensive DL tasks to our v-FAPs, while leveraging on blockchain technology to provide secure and distributed management of our OF-RAN. The v-FAPs employ federated DL models that run locally in a distributed manner among the service nodes and local model updates are sent to the seed node for aggregation. Through smart contracts, our OF-RAN establishes a blockchain network of seed nodes to maintain a distributed ledger of all service nodes involved in the v-FAP formation.

An emulated v-FAP is implemented and utilized for an experimental evaluation of our proposed system for federated DL to support an object detection application. The experimental results validated the DL performance of our system in terms of latency and precision accuracy under the effect of varying number of service nodes in our v-FAP. A simulated blockchain network is also implemented and utilized for evaluating the blockchain performance of our system in terms of stale block rate and throughput under the effect of varying block size and block interval. Both experimental and simulation results demonstrated the efficacy of the system. An appropriate selection of settings for the block size, block interval, and number of service nodes to meet the various and often competing requirements of stale block rate, throughput, latency, and precision accuracy is also demonstrated.

CHAPTER VIII: Conclusion and Future Work

The thesis has undertaken an in-depth investigation of various challenges faced by the current RANs, with a focus on providing a secure and scalable solution. In the first three chapters of the thesis, we have presented the introduction, background and literature review. The next four chapters elaborated our research contributions in this thesis, which are summarized as follows:

8.1. Summary of Contributions

In Chapter IV, we address the research gaps 1 & 2 by proposing the idea of OF-RAN. Here we contribute the concept of a v-FAP formed by two or more local edge devices and monitored by physical FAPs for collaborative task processing. The proposed OF-RAN is a low-latency and high-scalable alternative to the current F-RAN and C-RAN architectures.

In Chapter V, we utilize a multi-objective optimization technique to resolve the research gap 3 i.e., the task-to-node assignment (TNA) issue in the OF-RAN. We formulate this as an MOP with a goal of finding optimal assignment minimizing energy and latency of the v-FAP, while maximizing fairness (or load balancing) amongst its service nodes by minimizing their maximum load. We used MOEA/D to solve the tri-objective optimization problem. Simulation results demonstrate that higher number of service nodes in a v-FAP increases the number of optimal solutions, while decreases the energy cost, latency cost, and the load standard deviation of the service nodes (suggesting better load balancing). On the other hand, a higher number of service tasks not only increases the number of solutions, but also increases the energy cost, latency cost, and the load standard deviation of the service nodes

In Chapter VI, we address research gap 4, in which we model and evaluate the performance of OF-RAN against the existing RAN architectures to understand how our architecture can be

used as alternative solution in a high stressed environment. It analyses the energy consumption, completion delay, and failure rate performances under the effect of varying scenarios. The analysis supports our hypothesis that our proposed OF-RAN is scalable and can complement existing C-RAN and F-RAN architectures.

In Chapter VII, we address research gap 5 by harnessing the inherent security of decentralization in blockchain technology to propose our blockchain enabled OF-RAN and demonstrating its efficacy through a DL application case study. Here, federated DL is modelled and executed at the v-FAP for resource-limited clients such as IoT devices. The effect of factors such as varying number of service nodes in a v-FAP, block size, and block interval on the proposed system's stale block rate, throughput, latency and precision accuracy are investigated.

8.2. Future Work

The analytical, simulation, and experimental works conducted in his thesis have demonstrated good potential for the proposed OF-RAN architecture. There are various directions that can be pursued for future work, some of which are enlisted below:

- Investigate the role of v-FAPs in optimal functional split between the cloud and OF-RAN for future radio access networks.
- Further investigate the use of the OF-RAN for efficient processing of other offloaded machine-learning or signal processing tasks.
- Investigate the use of Open RAN (O-RAN) for flexible deployment of OF-RAN along with other RANs based on network operators resource requirement.
- It will be also interesting to investigate how the use of cognitive radio in OF-RAN can expand its notion of opportunistic access to device resources to include opportunistic access to spectrum resources.

REFERENCES

1. J. Tang, W. P. Tay, T. Q. Quek, and B. Liang, "System Cost Minimization in Cloud RAN With Limited Fronthaul Capacity," IEEE Transactions on Wireless Communications, vol. 16, Issue. 5, pp. 3371-3384, March 2017.
2. M. Peng, Y. Sun, X. Li, Z. Mao, and C. Wang, "Recent advances in cloud radio access network system architecture, key techniques, and issues", IEEE Communication Surveys and Tutorials, vol. 18, Issue 3, pp 2282-2308, March 2016.
3. A. Checko, H. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. Berger, and L. Dittmann, "Cloud RAN for mobile networks - a technology overview," IEEE Communication Surveys and Tutorials, vol. 17, no. 1, pp. 405–426, September 2014.
4. M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog computing-based radio access networks: Issues and challenges", in-press, IEEE Networks Magazine, 2016.
5. Y. Shih et al., "Enabling Low- Latency Applications in Fog Radio Access Networks," IEEE Network, vol. 31, no. 1, pp. 52–58, January 2017.
6. S. Hung et al., "Architecture Harmonization between Cloud Radio Access Networks and Fog Networks," IEEE Access, vol. 3, pp. 3019–34, December 2015
7. A. V. Dastjerdi, H. Gupta, R. N. Calheiros, S. K. Ghosh, and R. Buyya, "Fog computing principles architecture and applications", Chapter 7 in Internet of Things: Principles and Paradigms (Eds. Buyya & Dastjerdi), Elsevier, Massachusetts, USA, 2016.
8. L. Lilien, A. Gupta, Z. Kamal, and Z. Yang, "Opportunistic resource utilization networks—a new paradigm for specialized ad hoc networks", Computers and Electrical Engineering, vol. 36, no. 2, pp. 328–340, 2010.
9. R. Bassoli, F. Granelli, S.T. Arzo, M. Di Renzo, "Toward 5G cloud radio access network: An energy and latency perspective", Transaction Emergent Telecommunications Technology, vol. 27, no. 1, pp. 433–446, May 2019
10. R. T. Rodoshi, T. Kim, & W. Choi, "Resource management in cloud radio access network: Conventional and new approaches", Sensors, vol. 20, no. 9, pp. 2708, May 2020.

11. S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, & R. Langar "5G RAN: Functional Split Orchestration Optimization", IEEE Journal on Selected Areas in Communications, vol. 38, no. 7, pp. 1448–1463, 2020.
12. L. Wang, S. Zhou, "Flexible functional split and power control for energy harvesting cloud radio access networks", IEEE Transactions on Wireless Communications, vol.19, Issue.3, pp.1535-1548, November 2019.
13. L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," Communication Surveys and Tutorials, vol. 21, no. 1, pp. 146–172, 1st Quart., 2019.
14. X. Wang, L. Wang, S. E. Elayoubi, A. Conte, B. Mukherjee, and C. Cavdar, "Centralize or distribute? A techno-economic study to design a low-cost cloud radio access network," in Proc. IEEE International Conference Communication (ICC), Paris, France, pp. 1–7, May 2017.
15. U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Sege, and P. Sehier, "Quantitative analysis of split base station processing and determination of advantageous architectures for LTE," Bell Labs Technical Journal, vol. 18, no. 1, pp. 105–128, June 2013.
16. A. Marotta, D. Cassioli, K. Kondepu, C. Antonelli, and L. Valcarenghi, "Efficient management of flexible functional split through software defined 5G converged access," in Proc. IEEE International Conference Communications, pp. 1–6. May 2018.
17. Z. Guizani, N. Hamdi, "CRAN, H-CRAN, and F-RAN for 5G systems: Key capabilities and recent advances. International Journal of Network Management", 27(5), e1973, 2017.
18. G. S. Rahman, M. Peng, K, Zhang, & S. Chen, "Radio resource allocation for achieving ultra-low latency in fog radio access networks", IEEE Access, vol. 6, pp. 17442-17454, 2018.
19. M. A. Habibi, M. Nasimi, B. Han, & H. D. Schotten, "A comprehensive survey of RAN architectures toward 5G mobile communication system", IEEE Access, vol.7, pp. 70371-70421, 2019.
20. Z. Li, M. L. Sichitiu, & X. Qiu, "Fog radio access network: A new wireless backhaul architecture for small cell networks", IEEE Access, vol. 7, pp 14150-14161, 2018.

21. R. T. Marler, J. S. Arora, “Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization”, vol. 26, pp. 369-395, 2004.
22. H. Li, Q. “Zhang, Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II”, IEEE transactions on evolutionary computation, vol. 13, Issue. 2, pp 284-302, September 2008.
23. Q. Zhang, & H. Li, “MOEA/D: A multi-objective evolutionary algorithm based on decomposition.”, IEEE Transactions on evolutionary computation, vol. 11, Issue 6, pp. 712-731, November 2007.
24. X. Ling, J. Wang, T. Bouchoucha, B. C. Levy, and Z. Ding, ‘‘Blockchain radio access network (B-RAN): Towards decentralized secure radio access paradigm,’’ IEEE Access, vol. 7, pp. 9714–9723, January 2019.
25. W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Liang, Y. C. Yang, Q. Yang & C. Miao, “Federated learning in mobile edge networks: A comprehensive survey”, IEEE Communications Surveys and Tutorials, vol. 22, Issue.3, pp. 2031-2063, April 2020.
26. J. Jijin and B.-C. Seet, “Opportunistic fog computing for 5G radio access networks: A position paper”, In Proc. 3rd EAI International Conference on Smart Grid and Innovative Frontiers in Telecommunications (SmartGIFT), Auckland, New Zealand, April 2018
27. I. Chih-Lin, J. Huang, R. Duan, C. Cui, J. X. Jiang, and L. Li, “Recent progress on C-RAN centralization and cloudification,” IEEE Access, vol. 2, pp. 1030–1039, 2014.
28. P. Rost, C. J. Bernardos, A. De Domenico, M. Di Girolamo, M. Lalam, A. Maeder, et al., "Cloud technologies for flexible 5G radio access networks," IEEE Communications Magazine, vol. 52, pp. 68-76, 2014.
29. D. Wubben, P. Rost, J. Barlett, M. Lalam, V. Savin, M. Gorgogolione, A. Dekorsy and G. Fettweis, “Benefits and impact of cloud computing on 5G signal processing”, IEEE Signal Processing Magazine, vol. 31, no. 6, pp. 35-44, 2014.
30. C.-Y. Chang, R. Schiavi, N. Nikaein, T. Spyropoulos, and C. Bonnet, "Impact of packetization and functional split on C-RAN fronthaul performance," in Communications (ICC), 2016 IEEE International Conference on, 2016, pp. 1-7.

31. D. Astely, E. Dahlman, G. Fodor, S. Parkvall, and J. Sachs, “LTE release 12 and beyond”, IEEE Communications Magazine, vol. 51, no. 7, pp. 154–160, 2013.
32. E. Hossain, M. Rasti, H. Tabassum, and A. Abdelnasser, “Evolution towards 5G multi-tier cellular wireless networks: An interference management perspective,” IEEE Wireless Communications, vol. 21, no. 3, pp. 118-127, June 2014.
33. M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, “Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies,” IEEE Wireless Communications, vol. 21, no. 6, pp. 126–135, Dec. 2014.
34. T. X. Vu, H. D. Nguyen, and T. Q. Quek, "Adaptive compression and joint detection for fronthaul uplinks in cloud radio access networks," IEEE Transactions on Communications, vol. 63, pp. 4565-4575, 2015
35. Liu, S. Bi, and R. Zhang, "Joint power control and fronthaul rate allocation for throughput maximization in OFDMA-based cloud radio access network," IEEE Transactions on Communications, vol. 63, pp. 4097-4110, 2015.
36. S.-H. Park, O. Simeone, and S. S. Shitz, "Joint optimization of cloud and edge processing for fog radio access networks," IEEE Transactions on Wireless Communications, vol. 15, pp. 7621-7632, 2016.
37. R. G. Stephen and R. Zhang, "Joint millimeter-wave fronthaul and OFDMA resource allocation in ultra-dense CRAN," IEEE Transactions on Communications, vol. 65, pp. 1411-1423, 201.
38. A. Radwan, K. M. S. Huq, S. Mumtaz, K.-F. Tsang, and J. Rodriguez, "Low-Cost On-Demand C-RAN Based Mobile Small-Cells," IEEE Access, vol. 4, pp. 2331-2339, 2016.
39. S. Yan, M. Peng, M. A. Abana, and W. Wang, "An Evolutionary Game for User Access Mode Selection in Fog Radio Access Networks," IEEE Access, vol. 5, pp. 2200-2210, 2017.
40. M. Peng, Y. Li, T. Q. Quek, and C. Wang, "Device-to-device underlaid cellular networks under Rician fading channels," IEEE Transactions on Wireless Communications, vol. 13, pp. 4247-4259, 2014.

41. T.-C. Chiu, W.-H. Chung, A.-C. Pang, Y.-J. Yu, and P.-H. Yen, "Ultra-low latency service provision in 5G Fog-Radio Access Networks," in Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on, 2016, pp. 1-6.
42. S. S. Tanzil, O. N. Gharehshiran, and V. Krishnamurthy, "Femto-cloud formation: A coalitional game-theoretic approach," in Global Communications Conference (GLOBECOM), pp. 1-6, 2015.
43. M. Vondra and Z. Becvar, "QoS-ensuring distribution of computation load among cloud-enabled small cells," IEEE 3rd International Conference on in Cloud Networking (CloudNet), pp. 197-203, 2014.
44. S. Wang, M. Zafer, and K. K. Leung, "Online Placement of Multi-Component Applications in Edge Computing Environments," IEEE Access, vol. 5, pp. 2514-2533, 2017.
45. A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," Computer Networks, vol. 54, no. 6, pp. 960–976, 2010.
46. L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing", IEEE Internet Things Journal, vol. 5, no. 1, pp. 283–294, 2018.
47. L. Cui, C. Xu, S. Yang, J. Z. Huang, J. Li, X. Wang, Z. Ming, and N. Lu, "Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things", IEEE Internet Things Journal, vol. 6, no. 3, pp. 4791-4803, 2018.
48. T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," IEEE Transactions on Communications, vol. 65, no. 8, pp. 3571–3584, 2017.
49. X. Li, D. Zhou, Q. Pan, Y. Tang, and J. Huang, "Weapon-target assignment problem by multi-objective evolutionary algorithm based on decomposition", Complexity, vol. 2018, Article ID 8623051, 19 pages, 2018.
50. W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem", In Multi-Objective Memetic Algorithms (Eds. Goh, Ong

& Tan). Studies in Computational Intelligence, vol. 171, pp. 309-324. Springer, Berlin, Heidelberg, 2009.

51. C. Guerrero, I. Lera and C. Juiz "Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures", Future Generation Computer Systems, vol. 97, pp. 131-144, 2019.
52. H. M. Abdel-Atty, R. S. Alhumaima, S. M. Abuelenin, & E. A. Anowr, "Performance analysis of fog-based radio access networks", IEEE Access, vol.7, pp. 106195-106203, 2019.
53. T. C. Chiu, A. C. Pang, W. H. Chung, & J. Zhang, "Latency-driven fog cooperation approach in fog radio access networks," IEEE Transactions on Services Computing, vol. 12, no. 5, pp. 698-711, 2018.
54. Q. Li, J. Lei, J. Lin, & X. Wu, "Latency minimization for multiuser computation offloading in fog-radio access networks," arXiv preprint arXiv:1907.08759, 2019.
55. M. Peng, & K. Zhang, "Recent advances in fog radio access networks: Performance analysis and radio resource allocation," IEEE Access, vol. 4, pp. 5003-5009, 2016.
56. X. Zhang, & M. Peng, "Testbed design and performance emulation in fog radio access networks," IEEE Network, vol. 33, no. 3, pp. 49-57, 2019.
57. M. Xu, Z. Zhao, M. Peng, Z. Ding, T. Q. Quek, & W. Bai, "Performance analysis of computation offloading in fog-radio access networks," In Proceedings IEEE International Conference on Communications (ICC), Shanghai, China, May 2019.
58. L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji, & Y. Zhang, "Reinforcement learning-based mobile offloading for edge computing against jamming and interference", IEEE Transactions on Communications, vol. 68, no. 10, pp. 6114-6126, 2020.
59. H. Xiang, S. Yan, & M. Peng, "A realization of fog-RAN slicing via deep reinforcement learning", IEEE Transactions on Wireless Communications, vol. 19, no. 4, pp. 2515-2527, 2020.
60. Z. Zhao, C. Feng, H. H. Yang, & X. Luo, "Federated-learning-enabled intelligent fog radio access networks: fundamental theory, key techniques, and future trends", IEEE Wireless Communications, vol. 27, no. 2, 2020.

61. Y. Jiang, W. Huang, M. Bennis, & F. C. Zheng, "Decentralized asynchronous coded caching design and performance analysis in fog radio access networks," IEEE Transactions on Mobile Computing, vol.19, Issue. 3, pp 540-551, 2019.
62. A. Peng, Y. Jiang, M. Bennis, F. C. Zheng, & X. You, "Performance analysis and caching design in fog radio access networks," IEEE Globecom Workshops pp. 1-6, December 2018.
63. C. Wan, Y. Jiang, F. C. Zheng, P. Zhu, X.Gao, & X. You, "Analysis of Delay and Energy Efficiency in Fog Radio Access Networks with Hybrid Caching," IEEE Globecom Workshops pp. 1-6, December, December,2019.
64. X. Lyu, H. Tian, Ni. Wei, Y. Zhang, P. Zhang & R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing", IEEE Transactions on Communications, vol. 66, Issue. 6, pp 2603-2616, 2018.12
65. J. Du, L. Zhao, J. Feng & X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," IEEE Transactions on Communications, vol. 66, Issue. 4, pp 1594-1608, 2018.13
66. X. Tao, K. Ota, M. Dong, H. Qi, & K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," IEEE Wireless Communications Letters, vol.6, Issue.6, pp 774-777, 2017.16
67. J. Yao, & N. Ansari, "Fog resource provisioning in reliability-aware IoT networks," IEEE Internet of Things Journal, vol. 6, Issue. 5, pp 8262-8269, 2019.17
68. H. Zhu, C. Huang, and J. Zhou., "Edgechain: Blockchain-based multi-vendor mobile edge application placement", 4th IEEE Conference on Network Softwarization and Workshops (NetSoft) pp. 222-226, June 2018.
69. Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, "Edge computing resource management and pricing for mobile blockchain", arXiv preprint arXiv:1710.01567.
70. C. Xue, N. Xu, and Y. Bo, "Research on Key Technologies of Software-Defined Network Based on Blockchain", IEEE International Conference on Service-Oriented System Engineering (SOSE) pp. 239-2394, April 2019.

71. P. K. Sharma, M. Y. Chen and J. H. Park, “A software defined fog node based distributed blockchain cloud architecture for IoT”, IEEE Access vol 6, pp 115-124,2017.
72. O. Novo, “Blockchain meets IoT: An architecture for scalable access management in IoT”, IEEE Internet Things Journal, vol. 5, no. 2, pp. 1184–1195, April 2018.
73. S. Rathore, Y. Pan, & J. H. Park, “BlockDeepNet: a Blockchain-based secure deep learning for IoT network,” Sustainability, vol.11(14), 3974.
74. J. Weng, J. Zhang, M. Li, Y. Zhang, & W. Luo, “Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive”, IEEE Transactions on Dependable and Secure Computing, 2019.
75. Y. Lu, X. Huang, K. Zhang, S. Maharjan, & Y. Zhang, “Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles,” IEEE Transactions on Vehicular Technology, vol.69, Issue.4, pp 4298-4311, 2020
76. L. Cui, X. Su, Z. Ming, Z. Chen, S. Yang, Y. Zhou, & W. Xiao, “CREAT: Blockchain-assisted Compression Algorithm of Federated Learning for Content Caching in Edge Computing,” IEEE Internet of Things Journal, pp 1-1, 2020 (Early Access).
77. Y. Lu, X. Huang, K. Zhang, S. Maharjan, & Y. Zhang, “Low-latency Federated Learning and Blockchain for Edge Association in Digital Twin empowered 6G Networks,” IEEE Transactions on Industrial Informatics, pp 1-1, 2020 (Early Access).
78. S. Rathore, B. W. Kwon, & J. H. Park, “BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network”, Journal of Network and Computer Applications, 143, pp 167-177, 2019.
79. C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, ‘‘Fog computing for the Internet of Things: A survey,’’ ACM Transactions Internet Technology, vol. 19, no. 2, p. 18, 2019.
80. S. D’Oro, M. A. Marotta, C. B. Both, L. Da Silva, and S. Palazzo, “Power efficient resource allocation in C-RANs with SINR constraints and deadlines,” IEEE Transactions on Vehicular Technology, vol. 68, no. 6, pp. 6099–6113, Jun. 2019.

81. J. Tang, T. Q. S. Quek, T.-H. Chang and B. Shim, “Systematic resource allocation in cloud RAN with caching as a service under two timescales”, IEEE Transactions on Communications, vol. 67, no. 1, pp. 7755-7770, 2019.

82. J. Li, X. Shen, L. Chen, D. P. Van, J. Ou, L. Wosinska, and J. Chen, “Service Migration in Fog Computing Enabled Cellular Networks to Support Real-time Vehicular Communications,” IEEE Access, 2019.

83. Q. Zhang, L. Gui, F. Hou, J. Chen and S. Zhu, “Dynamic task offloading and resource allocation in mobile edge computing in dense Cloud RAN”, IEEE Internet of Things Journa1, pp. 1-1, January 2020 (Early Access).

84. J. Jijin, B.-C. Seet, P. H. J. Chong, and H. Jarrah, “Service load balancing in fog-based 5G radio access networks”, In Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Montreal, Canada, October 2017.

85. T. Cover, and J. Thomas, “Elements of information theory”, 2nd edition, John Wiley and Sons, Hoboken, NJ, USA, 2006.

86. Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, et al., “A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems”, IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 550-586, 2016.

87. X.-S. Yang, “Nature-inspired metaheuristic algorithms”. Luniver Press, Beckington, Somerset, UK, 2010.

88. X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, et al., “An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks,” Journal of Network and Computer Applications, vol.133, pp 75-85, 2019.

89. J. Du, L. Zhao, J. Feng, & X. Chu, “Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee,” IEEE Transactions on Communications, vol. 66, issue.4, pp. 1594-1608, 2018.

90. S. Luo, X. Chen, Z. Zhou, X. Chen, & W. Wu, "Incentive-Aware Micro Computing Cluster Formation for Cooperative Fog Computing", IEEE Transactions on Wireless Communications, vol. 19, no. 4, pp. 2643-2657, 2020.
91. J. Jijin, B. C. Seet, & P. H. J. Chong, "Multi-objective optimization of task-to-node assignment in opportunistic fog RAN," Electronics, vol. 9, no. 3, 14 pages, 2020.
92. J. Jijin, B.-C. Seet, & P. H. J. Chong, "Blockchain enabled opportunistic fog-based radio access network: a position paper", In Proc. 29th International Telecommunication Networks and Applications Conference (ITNAC), Auckland, New Zealand, November 2019.
93. Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, & H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks", IEEE Network Magazine, vol. 31, no. 1, pp. 52-58, 2017.
94. T. S. Rappaport, G. R. MacCartney, M. K. Samimi, & S. Sun, "Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design," IEEE Transactions on Communications, vol. 63, no. 9, pp. 3029-3056, 2015.
95. S. Sun, T. S. Rappaport, T. A. Thomas, A. Ghosh, H. C. Nguyen, et. al, "Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications," IEEE Transactions on Vehicular Technology, vol. 65, no. 5, pp. 2843-2860, 2016.
96. A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf & S. Capkun, "On the security and performance of proof of work blockchains." In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security pp. 3-16, October 2016.
97. Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, & K. Shudo, "Simblock: A blockchain network simulator," IEEE Conference on Computer Communications Workshops (INFOCOM) pp. 325-329, April 2019.
98. J. Jijin, B. C. Seet, & P. H. J. Chong, "Performance Analysis of Opportunistic Fog Based Radio Access Networks.," IEEE Access, 8, 225191-225200, 2020.