Optimized Federated Learning Framework for Open Radio Access Networks (O-RAN)

by

Amardip Kumar SINGH

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN
PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, 19-DEC-2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Prof. Kim Khoa Nguyen, Thesis supervisor
Dept. of Electrical Engineering, École de technologie supérieure (ÉTS)

Prof. Chamseddine Talhi, Chair, Board of Examiners
Department of Software and IT Engineering, École de technologie supérieure (ÉTS)

Prof. Rami Langar, Member of the Jury
Department of Software and IT Engineering, École de technologie supérieure (ÉTS)

Prof. Ben Liang, External Independent Examiner
Department of Electrical and Computer Engineering, University of Toronto

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON 18-DEC-2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This thesis is submitted for the degree of Doctor of Philosophy at the École de technologie supérieure (ÉTS) under University of Québec. The research described herein are conducted under the supervision of Prof. Kim Khoa Nguyen in the Department of Electrical Engineering, between Winter 2020 and Fall 2025.

The thesis is structured as a compilation of articles published or submitted in top-teer journals in the fields of Communication Networks and Machine Learning. The included articles are integrated with high fidelity to ensure its compliance with the proposed and published papers' structure and shape. However, peripheral modifications (such as figure box, page margin, positioning, rescaling) are reflected to adhere with the thesis guidelines of ÉTS.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support, guidance, and encouragement of many individuals who have accompanied me on this journey.

First and foremost, I extend my deepest gratitude to my supervisor, Prof. Kim Khoa Nguyen, whose expertise, patience, and unwavering support have been invaluable throughout this research. Your insightful questions challenged me to think more deeply. I am profoundly grateful for the intellectual freedom you gave me to pursue my ideas while providing the guidance I needed to see them through.

I would like to thank my thesis research committee members for their thoughtful feedback and constructive criticism right from the DGA1033 course that significantly strengthened this work. Your diverse perspectives enriched my research in ways I could not have achieved alone. Also to my colleagues at Synchromedia Lab and A-2406, our brainstorming sessions really helped shape the problems.

To my family, I owe more than words can express. Hemant Bhaiya, thank you for believing in me even when the path seemed uncertain. My parents who could not even attend a school and yet believed in education. My first teacher, Sumant Bhaiya, who nurtured my interest in mathematics quite early. Your sacrifices and endless encouragement have been my inspiration.

To Ravneet, thank you for believing in me. I could fight my share of self-doubts and anxiety only because of your patience, understanding, and unwavering support. Your love guided me through the emotional roller coaster of this journey.

Finally, to my friends, who have helped me brave the difficult times of pandemic, with whom I could find a sense of belonging in a foreign city, and persevere towards my goal. To those who reminded me of life beyond academia, celebrated small victories with me, and provided much-needed perspective and laughter, thank you for keeping me grounded and human.

Any errors or shortcomings in this work remain entirely my own.

Cadre d'apprentissage fédéré optimisé pour les réseaux d'accès radio ouverts (O-RAN)

Amardip Kumar SINGH

RÉSUMÉ

L'architecture Open Radio Access Network (O-RAN) représente un paradigme transformateur pour la réalisation de la vision ambitieuse des réseaux mobiles au-delà de la cinquième génération (B5G) et de la sixième génération (6G). Grâce à ses caractéristiques fondamentales de désagrégation, d'interfaces ouvertes standardisées permettant une véritable interopérabilité multi-fournisseurs, et d'intelligence intégrée via les contrôleurs intelligents du RAN (RICs), O-RAN permet des capacités réseau sans précédent. La nature programmable d'O-RAN, en particulier via le RIC non temps réel (Non-RT-RIC) pour l'optimisation stratégique et le RIC quasi temps réel (Near-RT-RIC) pour la gestion tactique des ressources, crée une plateforme où l'intelligence artificielle et l'apprentissage automatique (ML) peuvent révolutionner les opérations réseau. À travers ces avancées, O-RAN permet la vision de réseaux mobiles supportant simultanément des exigences de services hétérogènes telles que l'amélioration de la bande large mobile (eMBB), les communications ultra-fiables et de très faible latence (uRLLC), et les communications massives de type machine (mMTC).

La réalisation des cas d'usage basés sur l'intelligence intégrée d'O-RAN dépend de manière critique du cadre de formation des modèles ML lui-même. L'apprentissage fédéré (FL), un paradigme d'apprentissage automatique distribué permettant la formation collaborative de modèles à travers des nœuds réseau désagrégés sans centraliser les données brutes, émerge comme une solution prometteuse. Contrairement à l'apprentissage centralisé, FL est particulièrement adapté à l'environnement multi-fournisseurs d'O-RAN sensible à la confidentialité où : les données opérationnelles ne peuvent pas être partagées au-delà des frontières des fournisseurs en raison de préoccupations concernant la confidentialité et la concurrence ; les volumes massifs de données provenant de millions d'équipements utilisateur et de milliers de stations de base créent des goulets d'étranglement de communication prohibitifs s'ils sont transmis à des serveurs centraux ; et les cadres réglementaires comme le RGPD mandatent la localisation des données. FL permet des cas d'usage critiques d'O-RAN incluant la détection d'anomalies distribuée across les équipements multi-fournisseurs pour la sécurité, l'optimisation collaborative du spectre sans exposer les algorithmes propriétaires, la prédiction de qualité d'expérience préservant la vie privée en utilisant des données d'abonné sensibles, et la gestion en temps réel des ressources radio exploitant les informations d'état de canal localisées qui seraient obsolètes si traitées de manière centralisée.

Cependant, le déploiement de FL dans les environnements O-RAN présente des défis techniques formidables fondamentalement distincts des scénarios d'apprentissage fédéré traditionnels. Les exigences strictes de synchronisation des boucles de contrôle d'O-RAN rendent le temps d'apprentissage tout aussi critique que la précision du modèle. Des contraintes de ressources sévères existent à plusieurs niveaux : l'unité distribuée (O-DU) et l'unité radio (O-RU) disposent d'une capacité de calcul limitée en tant que nœuds périphériques ; les Near-RT-RICs fonctionnent

sur des serveurs aux ressources limitées ; et les liaisons de backhaul reliant ces composants présentent une connectivité variable et limitée en bande passante. De plus, les appareils mobiles participant à la formation FL sont fréquemment remis entre les O-DU, causant la perte de calculs, des changements imprévisibles de l'ensemble des participants, et des délais d'agrégation causés par les nœuds traînards. Les déploiements O-RAN en production exigent également l'exécution simultanée de multiples tâches FL où chaque tâche servant potentiellement différentes tranches réseau (eMBB, uRLLC, mMTC) avec des objectifs conflictuels et des exigences de qualité de service tout en concurrençant pour les ressources d'infrastructure partagée pour la formation du modèle. Ensemble, cela pose trois défis principaux. Le modèle FL entraîné doit être (i) efficace pour gérer des capacités système diverses incluant une puissance de calcul hétérogène et des conditions de canal variables ; (ii) robuste pour assurer la performance du modèle global selon des algorithmes d'agrégation optimisés ; et (iii) fiable pour converger dans les seuils définis.

Cette thèse aborde systématiquement ces défis à travers trois contributions progressives. Premièrement, nous développons un cadre FL efficace en ressources et en communication unifié qui résout conjointement la sélection de formateurs et l'allocation de ressources tout en attaquant le goulet d'étranglement de communication par l'intégration synergique de techniques d'accélération basée sur le momentum et de techniques de compression agressive. Deuxièmement, nous proposons MHORANFed, un cadre d'apprentissage fédéré hiérarchique tenant compte de la mobilité qui traite explicitement les perturbations de passation inter-O-DU par le biais de l'agrégation hiérarchique mappée à l'architecture d'O-RAN, la réallocation adaptative des ressources pour les participants mobiles, et l'ajustement dynamique du poids d'agrégation. Troisièmement, nous présentons l'application O-FL rApp, un cadre d'orchestration au niveau système fonctionnant dans le Non-RT-RIC qui gère le cycle de vie complet de multiples tâches d'apprentissage par renforcement multi-agents fédérés concurrentes par l'assignation stratégique de tâches et de tranches, la réallocation tactique des ressources parmi les tâches concurrentes.

Contrairement aux travaux antérieurs se concentrant sur des aspects individuels isolément ou dans des contextes idéalisés, notre solution intégrée démontre la faisabilité du déploiement de qualité production à travers une analyse théorique rigoureuse, une validation expérimentale extensive utilisant des ensembles de données FL de référence, des testbeds O-RAN réalistes avec des modèles de canal conformes à la 3GPP et des traces de trafic 5G, alignés avec les spécifications de l'O-RAN Alliance. Cette recherche établit les principes fondamentaux et les cadres opérationnels permettant un cadre d'apprentissage fédéré autonome, adaptatif, préservant la vie privée à l'échelle qui sont des éléments constitutifs essentiels pour réaliser le plein potentiel d'O-RAN intelligent dans l'évolution B5G et les réseaux 6G.

Mots-clés: apprentissage fédéré, open RAN, O-RAN, contrôleur intelligent RAN, optimisation des ressources, efficacité de communication, compression de gradient, apprentissage hiérarchique, qualité de service, B5G, 6G, découpage de réseau, IA préservant la confidentialité

Optimized Federated Learning Framework for Open Radio Access Networks (O-RAN)

Amardip Kumar SINGH

ABSTRACT

The Open Radio Access Network (O-RAN) architecture represents a transformative paradigm for realizing the grand vision of beyond fifth-generation (B5G) and sixth-generation (6G) mobile networks. Through its foundational characteristics of disaggregation, standardized open interfaces enabling genuine multi-vendor interoperability, and embedded intelligence via RAN Intelligent Controllers (RICs), O-RAN enables unprecedented network capabilities. The programmable nature of O-RAN, particularly through the Non-Real-Time RIC (Non-RT-RIC) for strategic optimization and Near-Real-Time RIC (Near-RT-RIC) for tactical resource management, creates a platform where artificial intelligence and machine learning (ML) can revolutionize network operations. Through these advancements, O-RAN enables the vision of mobile networks simultaneously supporting heterogeneous service requirements such as enhanced Mobile Broadband (eMBB), ultra-Reliable Low-Latency Communications (uRLLC), and massive Machine-Type Communications (mMTC).

Realizing O-RAN's embedded intelligence based use-cases critically depends on the ML model training framework itself. Federated Learning (FL), a distributed machine learning paradigm enabling collaborative model training across disaggregated network nodes without centralizing raw data, emerges as a promising solution. Unlike centralized learning, FL is uniquely suited for O-RAN's multi-vendor, privacy-sensitive environment where: operational data cannot be shared across vendor boundaries due to confidentiality and competitive concerns; massive data volumes from millions of user equipment and thousands of base stations create prohibitive communication bottlenecks if transmitted to central servers; and regulatory frameworks like GDPR mandate data localization. FL enables critical O-RAN use cases including distributed anomaly detection across multi-vendor equipment for security, collaborative spectrum optimization without exposing proprietary algorithms, privacy-preserving quality-of-experience prediction using sensitive subscriber data, and real-time radio resource management leveraging localized channel state information that would be stale if centrally processed.

However, deploying FL in O-RAN environments presents formidable technical challenges fundamentally distinct from traditional federated learning scenarios. O-RAN's strict control loop timing requirements makes learning time equally critical as model accuracy. Severe resource constraints exist at multiple layers: Distributed Unit (O-DU) and Radio Unit (O-RU) have limited computational capacity as edge nodes; Near-RT-RICs operate on resource-constrained servers; and backhaul links connecting these components have time-varying, bandwidth-limited connectivity. Moreover, the mobile devices participating in FL training are frequently handed over between O-DUs, causing lost computations, unpredictable changes in the participant set, and aggregation delays from straggler nodes. Production O-RAN deployments also demand concurrent execution of multiple FL tasks where each task potentially serving different network slices (eMBB, uRLLC, mMTC) with conflicting objectives and quality-of-service requirements

while competing for shared infrastructure resources for the model training. Together, it poses three main challenges. The trained FL model must be (i) efficient to handle diverse system capabilities including heterogeneous compute power and varying channel conditions; (ii) robust to ensure global model performance under optimized aggregation algorithms; and (iii) reliable to converge within the defined thresholds.

This thesis systematically addresses these challenges through three progressive contributions. First, we develop a unified resource-efficient and communication-efficient FL framework that jointly solves trainer selection and resource allocation while attacking the communication bottleneck through synergistic integration of momentum-based acceleration and aggressive compression techniques. Second, we propose MHORANFed, a mobility-aware Hierarchical Federated Learning framework that explicitly handles inter-O-DU handover disruptions through hierarchical aggregation mapping to O-RAN's architecture, adaptive resource reallocation for mobile participants, and dynamic aggregation weight adjustment. Third, we present the O-FL rApp, a system-level orchestration framework operating in the Non-RT-RIC that manages the complete lifecycle of multiple concurrent Federated Multi-Agent Reinforcement Learning tasks through strategic task and slice assignment, tactical resource reallocation among competing tasks.

Unlike prior works focusing on individual aspects in isolation or idealized settings, our integrated solution demonstrates practical, production-grade deployment feasibility through rigorous theoretical analysis, extensive experimental validation using benchmark FL datasets, realistic O-RAN testbeds with 3GPP-compliant channel models and 5G traffic traces, aligned with O-RAN Alliance specifications. This research establishes foundational principles and operational frameworks enabling autonomous, adaptive, privacy-preserving federated learning framework at scale which are essential building blocks for realizing the full potential of intelligent O-RAN in B5G evolution and 6G networks.

Keywords: federated learning, open RAN, O-RAN, RAN intelligent controller, resource optimization, communication efficiency, gradient compression, hierarchical learning, quality of service, B5G, 6G, network slicing, privacy-preserving AI

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Page

LIST OF ABBREVIATIONS

ADMM	Alternating Direction Method of Multipliers
AR	Augmented Reality
B5G	Beyond Fifth Generation
6G	Sixth Generation
C-RAN	Cloud Radio Access Network
CN	Core Network
CPU	Central Processing Unit
CVX	Convex optimization library
EC	Edge Cloud
eMBB	enhanced Mobile Broadband
FL	Federated Learning
gNB	gNodeB (5G base station)
gNB-CU	gNodeB Centralized Unit
gNB-DU	gNodeB Distributed Unit
GPU	Graphics Processing Unit
HFL	Hierarchical Federated Learning
HO	Handover
IID	Independent and Identically Distributed
KPI	Key Performance Indicator

MILP	Mixed-Integer Linear Program
MINLP	Mixed-Integer Non-Linear Program
ML	Machine Learning
mMTC	massive Machine-Type Communications
MOSEK	Specialized solver for conic optimization
NAG	Nesterov Accelerated Gradient
Near-RT-RIC	Near-Real-Time RAN Intelligent Controller
Non-RT-RIC	Non-Real-Time RAN Intelligent Controller
O-CU	Open Centralized Unit
O-DU	Open Distributed Unit
O-RAN	Open Radio Access Network
O-RU	Open Radio Unit
OPEX	Operational Expenditure
PDCP	Packet Data Convergence Protocol
PPO	Proximal Policy Optimization
PWL	Piecewise Linear Approximation
QCQP	Quadratically Constrained Quadratic Program
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network

RAR	Resource Allocation and Routing
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
SCA	Successive Convex Approximation
SDP	Semidefinite Programming
SGD	Stochastic Gradient Descent
SINR	Signal-to-Interference-plus-Noise Ratio
SLA	Service Level Agreement
TFF	TensorFlow Federated
TSA	Task and Slice Assignment
TTT	Time-to-Trigger
UE	User Equipment
UPF	User Plane Function
uRLLC	ultra-Reliable Low-Latency Communications
V-RAN	Virtual Radio Access Network
VCG	Vickrey-Clarke-Groves
xApps	intelligent applications (in Near-RT-RIC)

rApps	Non-RT RIC applications
5GC	5G Core

INTRODUCTION

0.1 Context, Background, and Motivations

Mobile networks constitute the backbone of modern communication in today's digitally interconnected world, where user equipments (UEs) continuously generate and exchange data. According to industry projections, global mobile data traffic is anticipated to increase 4.2-fold by the end of 2028, reaching approximately 354 exabytes per month (Ericsson AB, 2023). The Radio Access Network (RAN) serves as a fundamental component of mobile networks, facilitating the connection between UEs and the core network (CN). Over time, RAN architectures have evolved to accommodate the ever-increasing demands and service requirements of UEs. The most recent advancement in this evolution is the Open Radio Access Network (O-RAN) architecture, specifically designed to address the ambitious use cases envisioned for beyond Fifth Generation (B5G) and Sixth Generation (6G) communications (O-RAN Alliance, 2019). More than 100 leading telecommunications industry stakeholders have formed the O-RAN Alliance to standardize this architecture. The global O-RAN market is projected to reach USD 15.6 billion by 2027, with major network operators such as Vodafone, Orange, and Rakuten committing to O-RAN deployments expected to cover 35% of their networks by 2026. These developments underscore the strategic significance of O-RAN as a transformative paradigm in the evolution of mobile network architectures (O-RAN White Paper, 2023).

While earlier RAN architectures such as Cloud-RAN (C-RAN) centralized baseband processing and Virtual RAN (V-RAN) enabled the softwarization of network functions, Open RAN (O-RAN) represents a genuine paradigm shift towards a fully disaggregated, intelligent, and open multi-vendor ecosystem (Figure 0.1). In O-RAN, the base station is decomposed into logically and physically distinct functional components: the Radio Unit (O-RU), responsible for radio frequency processing; the Distributed Unit (O-DU), which manages real-time Layer 2 protocols with control loops under 10 milliseconds; and the Centralized Unit (O-CU), which oversees

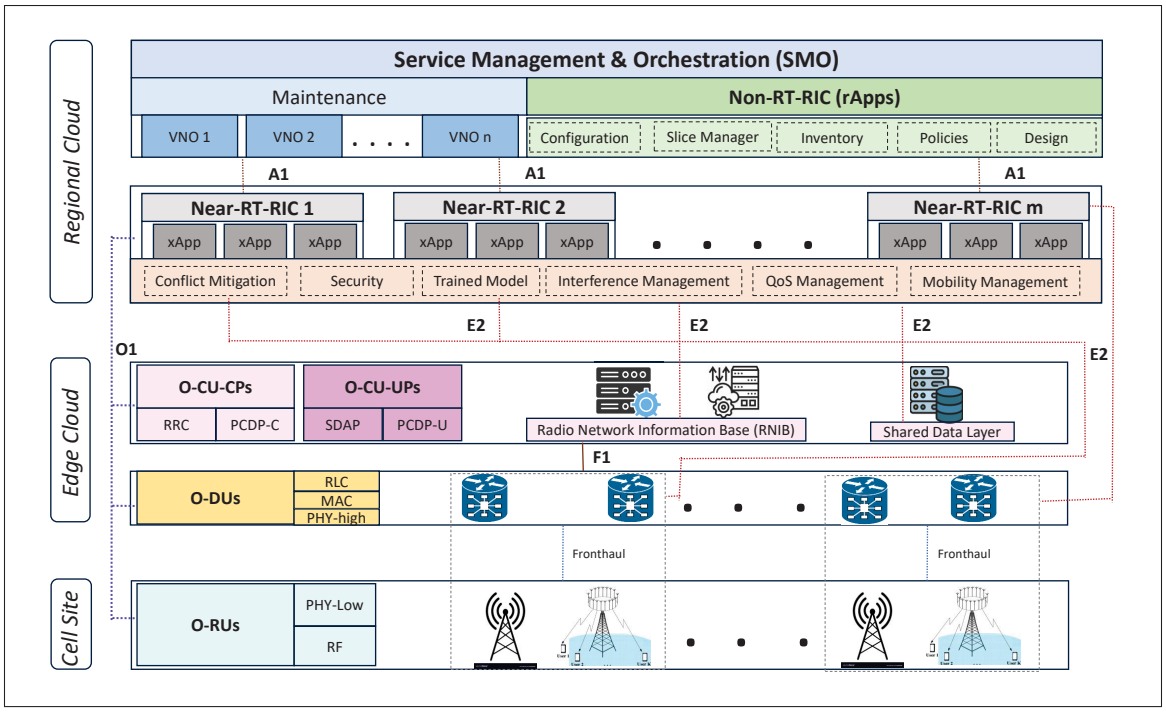


Figure 0.1 Disaggregated and Hierarchical O-RAN Architecture

higher-layer protocols, including the Radio Resource Control (RRC) and Packet Data Convergence Protocol (PDCP) (O-RAN Alliance, 2024c). Furthermore, O-RAN introduces vendor-neutral interfaces between network components, such as the Open Fronthaul interface between the O-DU and O-RU, the F1 interface connecting the O-DU and O-CU, and the E2 interface linking RAN components to intelligent controllers (Polese, Bonati, D'Oro, Basagni & Melodia, 2023a). The most transformative feature of O-RAN lies in its self-optimizing capability, enabled through the RAN Intelligent Controllers (RICs): the Non-Real-Time RIC (Non-RT-RIC), responsible for policy management and orchestration decisions with control loops exceeding 500 milliseconds, and the Near-Real-Time RIC (Near-RT-RIC), dedicated to resource optimization with control loops ranging between 10 and 500 milliseconds (Figure 0.2).

Realizing O-RAN's intelligent network vision critically depends on machine learning (ML) training frameworks capable of processing vast data collected and stored locally in distributed data

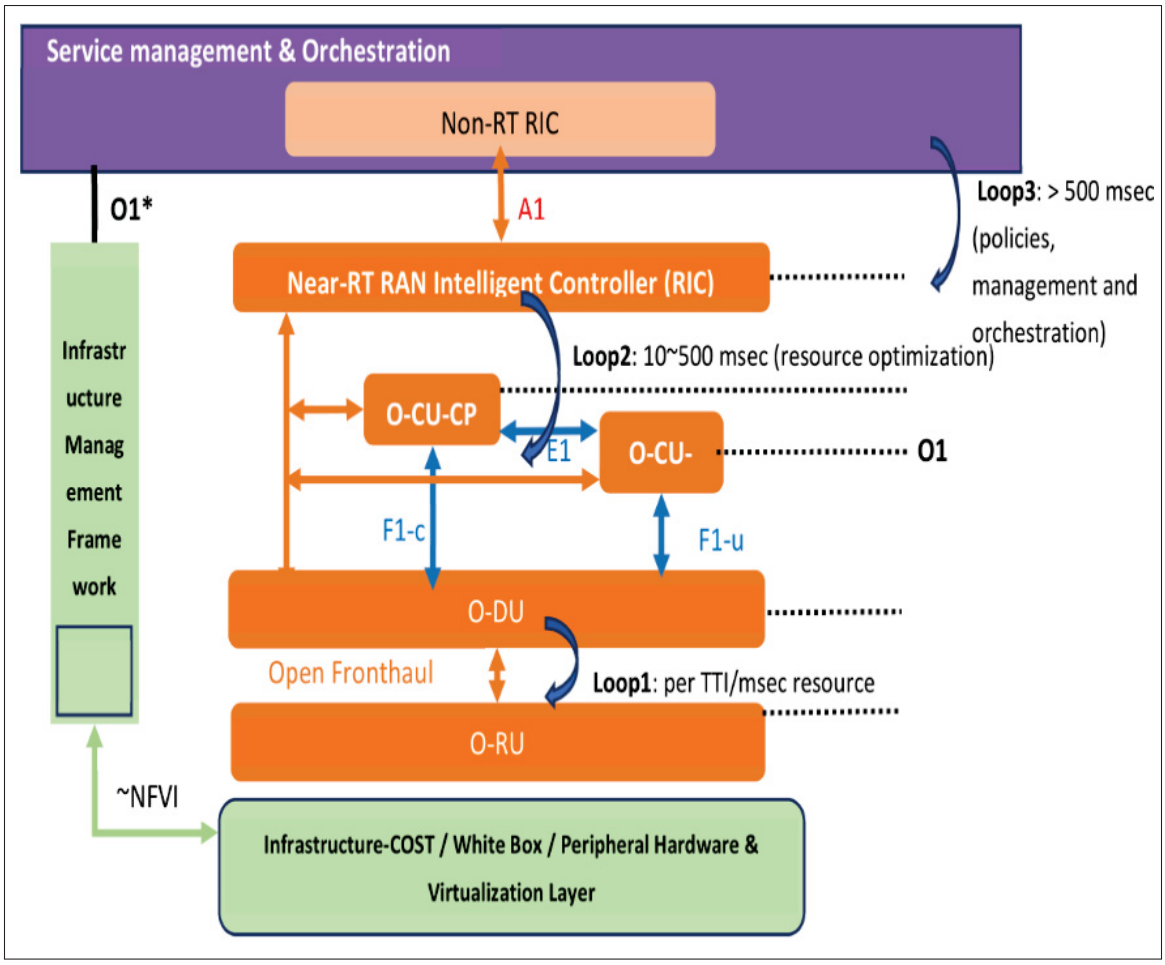


Figure 0.2 Standard interfaces and control loops of O-RAN (O-RAN Alliance, 2024b)

bases owned by multiple operators. Such capability is essential to provide ultra-Reliable Low-Latency Communications (uRLLC) for mission-critical services including industrial automation, remote telesurgery; enhanced Mobile Broadband (eMBB) to support ultra-high-definition video streaming, immersive augmented reality (AR); and massive Machine-Type Communications (mMTC) for smart cities and Industry 4.0 applications (ITU-R, 2015). However, classical centralized ML paradigm has two major drawbacks: (i) local data need to be transferred at a central server, and (ii) raises privacy concerns to the data collection nodes. While the role of predictive intelligence plays an important part in O-RAN, it also amounts to communication

overhead, data privacy concerns, scalability limitations, and latency constraints (Li, Sahu, Talwalkar & Smith, 2020a). Federated Learning (FL) emerges as a theoretically and practically viable alternative, enabling collaborative model training across geographically distributed network nodes while preserving data locality (McMahan, Moore, Ramage, Hampson & y Arcas, 2017). FL inherently addresses privacy requirements through distributed computation, reduces communication costs by transmitting model parameters rather than raw data (Zhuansun *et al.*, 2024).

The integration of FL within O-RAN architectures directly addresses 6G performance requirements through distributed cooperative learning across edge computing units, thereby reducing service latency while enhancing system reliability and scalability (Qin, Li & Ye, 2021). This capability is particularly critical for Quality of Service (QoS)-differentiated network slicing, where heterogeneous user groups (e.g., uRLLC and eMBB) generate distinct traffic patterns captured in distributed O-RAN data lakes. The time-series traffic data characterizing each slice varies in both volume and statistical distribution, creating non-identically distributed datasets across local nodes. FL-trained traffic prediction models can leverage this heterogeneity to enable proactive, slice-specific radio resource allocation aligned with differentiated QoS requirements. The canonical FL workflow in O-RAN RICs, illustrated in Figure 0.3, operates through iterative coordination between hierarchical aggregation layers: The global aggregator (Non-RT-RIC) initializes model distribution to participating local trainers (Near-RT-RICs), which either conduct direct training using local computational resources and data, or further delegate model training to subordinate RAN nodes within respective slice groups. Edge Cloud (EC) nodes execute local model training on raw data using dedicated computational infrastructure, subsequently transmitting trained model parameters through lower-level communication links to their respective slice managers. Near-RT-RICs perform intermediate hierarchical aggregation, optionally uploading consolidated parameters to regional cloud infrastructure while concurrent local training iterations proceed. The Non-RT-RIC executes global parameter aggregation and

broadcasts unified updates for subsequent federated rounds, periodically evaluating convergence criteria until acceptable model performance is achieved.

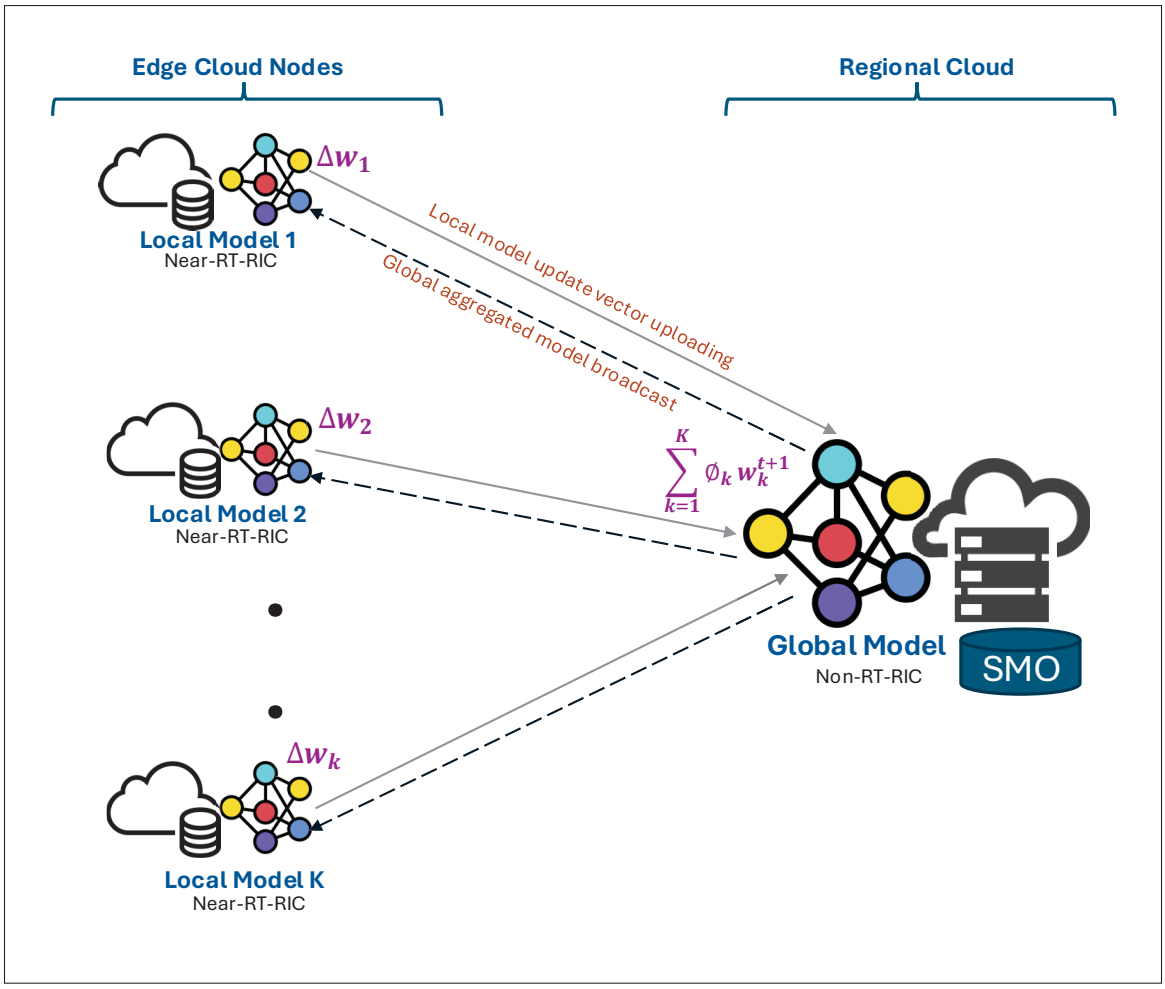


Figure 0.3 Federated Learning in O-RAN Intelligent Controllers

Despite its theoretical advantages, FL remains nascent in practical deployment, with numerous fundamental research challenges requiring systematic investigation (Kairouz *et al.*, 2021). The performance characteristics of FL systems exhibit strong dependency on deployment context and configuration parameters, precluding the existence of universally standard frameworks. This is particularly pronounced in O-RAN environments where heterogeneity pervades network topology, computational resources, and data distributions. An FL model training framework can be optimized by addressing the following interconnected research problems:

– **Algorithmic Robustness:** The aggregation algorithm must maintain stability under dynamic trainer availability and unreliable network conditions. Network-induced message loss, asynchronous trainer dropout during execution cycles, and intermittent connectivity introduce stochastic perturbations to the aggregation process, potentially compromising convergence guarantees. Robust aggregation mechanisms must therefore accommodate partial participation patterns while ensuring statistical consistency of parameter updates.

– **Resource Heterogeneity:** Computational heterogeneity across local trainers induces asynchronous model update cycles, where resource-constrained nodes exhibit significantly prolonged training latency compared to high-capacity counterparts. Communication bandwidth limitations restrict the fraction of trainers capable of simultaneous aggregator interaction, while physical layer constraints (including antenna transmit power budgets, UE battery capacity, base station sleep scheduling, and inter-cell interference) impose additional training bottlenecks. Effective FL frameworks must adaptively allocate resources and orchestrate training schedules to mitigate performance degradation under these multidimensional constraints.

– **Convergence Optimization:** While first-order gradient descent methods and stochastic gradient descent (SGD) variants constitute standard optimization approaches in FL, convergence rates exhibit strong sensitivity to data distribution characteristics and model architecture specifications. Hyperparameter optimization encompassing mini-batch size, learning rate schedules, and regularization parameters must account for the convex or non-convex geometry of distributed loss landscapes, particularly under conditions of non-independent and identically distributed (non-IID) data across trainers.

– **Communication Efficiency:** The iterative exchange of model parameters between distributed trainers and centralized aggregators imposes substantial communication overhead. Although individual parameter vectors occupy minimal bandwidth relative to raw datasets, modern deep neural network architectures comprise thousands to millions of parameters. Across hundreds of

global aggregation rounds, cumulative transmission costs and training latency become dominant performance bottlenecks, necessitating communication-efficient protocols incorporating gradient compression, sparsification, or quantization techniques.

– **Multi-Task Learning Coordination:** Scenarios where heterogeneous local nodes train distinct but related models introduce dual complexities: convergence analysis becomes mathematically intractable under arbitrary task distributions, and resource allocation must balance competing objectives across tasks. This challenge is particularly relevant for network slicing applications, where differentiated QoS requirements across slice-specific user groups demand coordinated multi-objective optimization frameworks.

– **Performance Goals:** FL system optimization inherently involves multi-objective decision-making among competing performance dimensions: model accuracy, generalization capability to unseen data distributions, end-to-end training latency, energy consumption, cumulative resource expenditure (computational and communicational), and system scalability. These metrics exhibit fundamental trade-offs. For instance, enhanced model accuracy typically demands increased communication rounds and computational resources, while aggressive communication compression improves efficiency at the cost of convergence rate and final accuracy. The absence of universally dominant solutions necessitates context-specific prioritization strategies that balance conflicting objectives according to deployment constraints and application requirements. Establishing appropriate performance metrics and their relative priorities constitutes a critical design decision that determines the operational feasibility and practical utility of FL frameworks in resource-constrained O-RAN environments.

Optimization of FL training in distributed O-RAN components is crucial as it paves way for an efficient and scalable intelligence at the edge of the network. Consider the following two use-cases:

– Due to the various QoS requirements of users, user behavior prediction is crucial for the optimization of wireless network performance. The QoS of users can be predicted using FL, where each near-RT-RIC (at edge cloud nodes) uses the FL algorithm based on stored information such as requested data, device type, and so forth, and all near-RT-RIC instances transmit the FL model results to a server (at regional cloud) to obtain a unified FL model. However, the processing capacity of these edge nodes are typically constrained by limited availability and heterogeneous compute power. Moreover, the backhaul link is shared by all the edge nodes for not just model training but for passing RAN's control messages too. We cannot wait for idle duration to train such a predictive model as it'll further jeopardize the QoS of the users. An optimal training algorithm will secure an efficient and reliable model that obeys the deadline for this QoS policy control-loop.

– Distributed anomaly detection can be trained leveraging multi-vendor equipment data to identify sophisticated attacks while preserving vendor confidentiality. Such a model also helps quality-of-experience prediction using sensitive subscriber behavior data that remains localized while enabling network-wide service improvements. In addition to tackling the resource constraints, here the learning time becomes critically important to minimize the failures. A framework that posses guaranteed convergence and requires minimal number of communication rounds can train the anomaly detection algorithm with acceptable model accuracy.

0.2 Problem Statement and Challenges

While an optimal FL training framework brings in great opportunities, the path to such implementation in O-RAN is fraught with distinct and deeply interrelated challenges spanning multiple layers, from low-level resource allocation to high-level multi-task orchestration. So, the main problem addressed in this thesis is:

"How to optimize the resources required for training O-RAN FL models and its performance while guaranteeing the convergence? "

This thesis structures its systematic investigation around identifying, formulating, and solving these challenges progressively, building from foundational algorithmic components to comprehensive system-level orchestration solutions. Due to the complexity of FL deployment in O-RAN, this main problem has been divided into the following subproblems that can be solved more efficiently:

0.2.1 SP 1: Subproblem 1

The first subproblem addresses the fundamental challenges of efficient FL deployment within resource-constrained O-RAN environments, encompassing both optimal resource management and communication optimization. We can articulate the first subproblem as:

"Can we efficiently allocate computational and communication resources while optimizing FL training time and model convergence in network slice aware heterogeneous O-RAN environments?"

The efficiency, convergence speed, and feasibility of FL in O-RAN depend critically on joint optimization of resource allocation and communication protocols. Naive transplantation of standard FL algorithms designed for idealized settings fails catastrophically in production O-RAN environments due to unique operational constraints and resource limitations.

C1: The main challenges related to SP1 are:

- *Strategic Local Trainer Selection:* The system must intelligently determine which subset of distributed nodes should participate in each training round, considering data quality and representativeness, computational capacity, backhaul channel quality, and strategic

importance for network optimization goals. Poor selection can slow convergence, introduce gradient noise, or destabilize learning through Byzantine-like failures.

- *Joint Resource Allocation Optimization:* The system must allocate scarce computational resources (CPU, GPU, memory) and communication bandwidth across selected trainers. These allocation decisions create complex trade-offs—additional computational power accelerates training but increases costs and may starve other network functions, while greater bandwidth reduces latency but consumes expensive backhaul capacity needed for user traffic.
- *Communication Bottleneck:* Contemporary deep learning models contain millions of parameters, resulting in model updates ranging from tens to hundreds of megabytes per participant per round. The cumulative communication time often exceeds local computation time by one to two orders of magnitude, becoming the dominant factor in total training time and violating O-RAN's rapid adaptation requirements.
- *Heterogeneous Data Constraints:* O-RAN environments exhibit diverse computational capabilities, varying network conditions, and different data distributions across nodes. Standard FL approaches fail to account for this heterogeneity, leading to straggler problems and suboptimal resource utilization.
- *Gradient Compression Trade-offs:* While compression techniques can reduce communication overhead, they introduce quantization errors that may degrade convergence quality. Balancing compression ratios with model accuracy while maintaining theoretical convergence guarantees poses significant challenges.

0.2.2 SP 2: Subproblem 2

The foundational optimization framework operates under the limiting assumption of static network topology. However, real-world cellular networks are characterized by constant dynamism driven by user mobility. So, we articulate the second subproblem as:

"Can we optimize a mobility-aware hierarchical FL framework to handle varying set of local trainers caused by user handovers?"

As mobile users move through coverage areas, frequent handovers between base stations introduce profound challenges to FL execution. O-RAN's hierarchical architecture naturally supports Hierarchical Federated Learning (HFL), but inter-DU handovers fundamentally disrupt the aggregation topology and require sophisticated mobility-aware coordination.

C2: The main challenges related to SP2 are:

- *Handover-Induced Computational State Loss:* In-progress local training computation may be abruptly interrupted during handovers, wasting computational resources and forcing training epoch restarts.
- *Varying Participants:* The set of active learners within cells changes unpredictably as UEs enter and exit coverage areas, creating time-varying participant populations that violate classical FL convergence assumptions. This requires adaptive aggregation strategies.
- *Global Aggregation Delays:* Central aggregation servers must handle "straggler" nodes undergoing handovers, introducing delays in global model updates that accumulate across rounds and significantly extend training time. Efficient waiting strategies are needed.
- *Hierarchical Topology Disruption:* Inter-DU handovers require reconfiguration of HFL aggregation topology, potential transfer of training state across DUs, and careful coordination to maintain learning progress without compromising the hierarchical structure benefits.
- *Convergence Guarantees Under Non-Stationarity:* Providing formal convergence guarantees for FL algorithms operating under time-varying, non-stationary participant sets poses significant theoretical and practical challenges requiring novel analytical frameworks.

0.2.3 SP 3: Subproblem 3

Given efficient single FL task execution and mobility-aware coordination, the system must address orchestration of multiple concurrent intelligent applications competing for shared infrastructure resources. We articulate the third subproblem as:

"Can we orchestrate, coordinate, and manage the lifecycle of multiple concurrent FL tasks with potentially conflicting objectives while maximizing overall system utility?"

The operational vision for O-RAN involves a dynamic, multi-tenant platform hosting an ecosystem of concurrent, independently developed intelligent applications. This introduces profound coordination challenges that transcend optimizing individual FL tasks, requiring sophisticated orchestration at the Non-RT-RIC level.

C3: The main challenges related to SP3 are:

- *Resource Contention and Allocation Conflicts:* Multiple xApps compete for finite computational resources and communication bandwidth. Aggregate resource demand typically exceeds available supply, necessitating principled resource arbitration that considers task priorities, deadlines, and resource requirements.
- *Objective Misalignment and Policy Conflicts:* Different xApps may pursue conflicting optimization objectives. For example, eMBB optimization favoring high throughput may conflict with uRLLC requirements for strict latency guarantees, requiring sophisticated conflict detection and resolution mechanisms.
- *Task Lifecycle Management:* New xApps require runtime control decisions, existing tasks may complete or enter dormant phases, and some tasks may need preemption to free resources for higher-priority applications. This changing environment requires adaptive orchestration strategies.

- *Global Utility Maximization:* The orchestrator must formulate and solve system-level optimization problems balancing conflicting objectives across all tasks to maximize aggregate network utility, which is a weighted combination of service quality, resource efficiency, and fairness metrics.

0.3 Research Objectives

In direct response to the multifaceted problem statement and the distinct technical challenges identified above, this thesis is guided by four primary, sequential research objectives. Each objective is carefully designed to address one of the sub-problems articulated in Section 0.2, with each building progressively upon the foundations established by its predecessors. Collectively, these objectives constitute a comprehensive research agenda for realizing intelligent, federated, and autonomous systems in next-generation Open RAN networks.

RO 1: Foundational Framework for Single-Task Communication Efficient FL in O-RAN: To rigorously formulate, mathematically model, and algorithmically solve the fundamental joint optimization problem of local trainer selection and multi-dimensional resource allocation (computation and communication) for a single Federated Learning task executing within the resource-constrained O-RAN architecture. The framework must reduce the communication overhead (total bits transmitted across all participants and rounds) and the time-to-convergence (wall-clock time to reach target accuracy), thereby making FL a more scalable, practical, and economically viable tool for deployment in time-sensitive, bandwidth-constrained O-RAN control loops where rapid model adaptation is critical.

RO 2: Mobility-Aware Hierarchical FL Framework: To develop, theoretically analyze, and empirically validate a novel, mobility-aware Hierarchical Federated Learning (HFL) framework explicitly designed for the multi-tiered O-RAN architecture. This

framework must explicitly model and account for the disruptive effects of UE mobility and inter-cell handovers on the distributed learning process.

RO 3: System-Level Orchestration Framework for Multiple Federated MARL Tasks: To optimize, implement, and experimentally evaluate a holistic, rApp-based orchestration framework for the lifecycle management, resource arbitration, and coordination of multiple, concurrent, potentially conflicting Federated Multi-Agent Reinforcement Learning (MARL) xApps executing within the O-RAN Near-RT-RIC.

0.4 List of Publications

The author has been actively presenting and publishing the ongoing works conducted during the course of his PhD. All the published and submitted articles are listed below:

0.4.1 Peer Reviewed Journals:

1. Amardip Kumar Singh and Kim Khoa Nguyen, "Communication Efficient Compressed and Accelerated Federated Learning in Open RAN Intelligent Controllers," in IEEE/ACM Transactions on Networking, vol. 32, no. 4, pp. 3361-3375, Aug. 2024.
2. Amardip Kumar Singh and Kim Khoa Nguyen, "User Handover Aware Hierarchical Federated Learning for Open RAN-Based Next-Generation Mobile Networks," in IEEE Transactions on Machine Learning in Communications and Networking, vol. 3, pp. 848-863, 2025.
3. Amardip Kumar Singh and Kim Khoa Nguyen, "Orchestrating xApps Resources for Multiple Federated MARL Tasks in O-RAN", in IEEE Transactions on Networking, 2025 (submitted).

0.4.2 International Conferences:

1. Amardip Kumar Singh and Kim Khoa Nguyen, "Joint Selection of Local Trainers and Resource Allocation for Federated Learning in Open RAN Intelligent Controllers," 2022

IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 2022, pp. 1874-1879.

2. Amardip Kumar Singh and Kim Khoa Nguyen, "MCORANFed: Communication Efficient Federated Learning in Open RAN," 2022 14th IFIP Wireless and Mobile Networking Conference (WMNC), Sousse, Tunisia, 2022, pp. 15-22.
3. Amardip Kumar Singh and Kim Khoa Nguyen, "MHORANFed: Hierarchical Federated Learning with User Hand- Over Awareness for Open RAN," 2025 IEEE International Conference on Communications (ICC), Montreal, 2025.

0.5 Thesis Outline and Organization

The remainder of this dissertation is organized to progressively address each research objective, building from theoretical foundations to complete system implementation.

– Chapter 1: provides a comprehensive, critical review of the state-of-the-art literature spanning multiple domains: O-RAN architecture specifications and ongoing standardization efforts; Federated Learning principles, algorithms, and convergence theory; resource optimization techniques in wireless networks; Hierarchical FL and mobility-aware learning; communication-efficient ML and model compression; and orchestration of multiple learning Tasks in networked systems. This chapter positions our contributions within the broader research landscape and identifies gaps that motivate our work.

– Chapter 2: details the overarching research methodology employed throughout this thesis. The chapter presents the core optimization techniques and algorithmic frameworks (non-convex optimization, decomposition methods) leveraged in our solutions. It also outlines the performance evaluation strategies, experimental testbed configurations, datasets employed, and metrics used to assess the proposed frameworks. A summary of our research is also presented with the thesis structure diagram.

– Chapter 3: presents our published work on communication efficient FL in O-RAN. It describes our foundational framework for joint trainer selection and resource allocation with integrated compression and acceleration based FL algorithm addressing the research objective 1.

– Chapter 4: presents our published work on Mobility-Aware Hierarchical Federated Learning framework addressing research objective 2. This chapter extends the foundational model to explicitly account for the hierarchical structure of O-RAN and the effects of user mobility due to hand-overs.

– Chapter 5: presents our submitted work, O-FL rApp: System-Level Orchestration of Multiple Federated MARL Tasks, addressing research objective 3.

– Finally, we provide our concluding remarks by synthesizing the key findings across all contributions, discussing their theoretical and practical implications for O-RAN deployments, acknowledging limitations of the current work, and outlining promising directions for future research that build upon the foundations established in this thesis.

CHAPTER 1

LITERATURE REVIEW

In the previous chapter, we introduced the problem of training FL models in O-RAN within the context of broader research impact. Then, we narrowed it down into the subproblems of resource and communication efficient FL, mobility-aware FL, and orchestration of multiple FL tasks in O-RAN. This chapter provides a critical review of the relevant literature.

This literature review is structured into five complementary sections that collectively address the multidimensional optimization challenges inherent to deploying federated learning in Open RAN systems. Section 1.1 establishes the wireless systems foundation by examining resource allocation, client selection, and communication efficiency techniques developed for FL in mobile access networks, thereby identifying optimization dimensions (bandwidth, computational capacity, local iterations, gradient compression) that are fundamental to the problem. Section 1.2 then synthesizes convergence acceleration theory to analyze how algorithmic choices, specifically momentum-based methods and compression operators, interact with the resource constraints identified in the previous section, revealing critical inter-dependencies that existing works treat independently. Section 1.3 introduces the temporal and topological dynamics of cellular networks by examining handover procedures and their impact on distributed learning, a dimension largely absent from the wireless-generic FL literature and essential to O-RAN control loop operation. Section 1.4 integrates previous sections into a system-level orchestration problem, where multiple heterogeneous FL services compete for shared edge infrastructure, requiring joint optimization of task assignment, resource allocation, and routing decisions. This progression from problem decomposition to holistic system integration reveals that each identified research gap stems not from deficiencies in individual optimization techniques, but rather from their isolated treatment and lack of formal integration under O-RAN-specific constraints (deadlines, slice differentiation, infrastructure-as-client paradigm, SLA guarantees). Consequently, the thesis contributions are structured to progressively integrate these dimensions: communication-efficient resource allocation under deadlines (Section 1.1 synthesis), heterogeneous convergence acceleration

(Section 1.2 synthesis), mobility-aware hierarchical learning (Section 1.3 synthesis), and finally, a closed-loop orchestration framework with convergence guarantees (Section 1.4 synthesis).

1.1 Federated Learning in RAN

1.1.1 Resource Allocation for FL in Wireless Networks

Early work on resource allocation for FL focused on minimizing the total energy consumption of mobile devices serving as FL clients. Yang, Chen, Saad, Hong & Shikh-Bahaei (2021) formulated a joint optimization problem to minimize computation and communication energy subject to latency constraints, developing approximated closed-form solutions for CPU frequency and transmit power allocation. Their analysis revealed that optimal resource allocation achieves a balance between local computation (which consumes CPU energy) and wireless transmission (which consumes transmit power energy), with the optimal operating point depending on channel conditions and deadlines.

Chen *et al.* (2021a) extended this framework to consider the joint optimization of learning and communication, incorporating the impact of bandwidth allocation on model accuracy. They demonstrated that allocating more bandwidth to clients with better channel conditions accelerates convergence, achieving 30% faster training compared to uniform allocation.

Dinh *et al.* (2021) addressed FL over wireless networks under synchronous communication, where the global aggregation round cannot begin until all selected clients have uploaded their updates. This introduces a straggler effect, where clients with poor channel conditions or limited compute resources delay the entire system. Their solution involves adaptive client selection and bandwidth allocation to minimize the maximum per-round latency, formulated as a mixed-integer non-linear program (MINLP).

Wang *et al.* (2019a) proposed adaptive FL for resource-constrained edge systems, introducing a control algorithm that jointly optimizes the number of local iterations E and global communication rounds T to achieve a target accuracy within a specified time budget. Their key insight is that

increasing local iterations reduces communication frequency but may lead to model divergence in non-IID settings, necessitating careful tuning.

Mo & Xu (2021) investigated energy-efficient FL with joint communication and computation design, revealing that the optimal strategy depends on the energy-accuracy tradeoff. For accuracy-critical applications, allocating more energy to computation (higher CPU frequency) is optimal, while for energy-critical applications, reducing communication frequency (more local iterations) dominates.

Research Gap: Commonly, these works are primarily focused on minimizing total energy consumption under the constraints of CPU frequency allocation, transmit power, bandwidth allocation, client selection, and number of local and global iterations. However, these works are designed for generic mobile edge computing scenarios and do not address the specific requirements and constraints of O-RAN systems. They assume user devices (e.g., smartphones) serve as FL clients, optimizing for battery life and user experience. In O-RAN, FL clients are network infrastructure elements (O-DUs hosting near-RT-RIC xApps), where the optimization objective should be minimizing operational expenditure (OPEX) and meeting SLA guarantees rather than device energy efficiency.

1.1.2 Client Selection Strategies

The FedAvg baseline (McMahan *et al.*, 2017) randomly samples a fixed fraction of clients in each round. This approach is unbiased and simple but ignores heterogeneity in client data distributions, channel conditions, and computational capabilities.

Cho, Wang & Joshi (2020) proposed importance-based sampling that selects clients with higher loss values (indicating greater potential for improvement) with higher probability. This accelerates convergence in heterogeneous settings but requires clients to compute and communicate their local losses, adding overhead.

Nishio & Yonetani (2019) formulated client selection as an optimization problem maximizing total utility (combination of data quality, channel conditions, and resource availability). Their greedy algorithm selects clients with the highest marginal utility, improving convergence by 20% over random selection.

Xu & Wang (2021) incorporated wireless channel state information into client selection, preferentially selecting clients with better channel conditions to minimize communication latency. However, this introduces bias toward well-connected clients, potentially neglecting valuable data from poorly connected regions.

Yang, Liu, Quek & Poor (2020) investigated scheduling policies for FL in wireless networks, comparing Round-Robin (fair but inefficient), Proportional Fair (balances fairness and efficiency), and Opportunistic (selects best channels but unfair). Their analysis shows that Proportional Fair scheduling achieves the best convergence-fairness tradeoff in dynamic wireless environments.

In the context of hierarchical FL, where edge aggregators perform intermediate aggregation before sending to the central server, user association (assigning clients to edge aggregators) becomes important. Liu, Yu, Chen & Bennis (2022) jointly optimized user association and resource allocation for hierarchical FL in multi-cell networks, demonstrating that co-locating clients with similar data distributions at the same edge aggregator accelerates convergence.

Research Gap: In O-RAN, the FL clients are infrastructure elements (near-RT-RIC instances) with relatively stable availability, but the data they process (slice-specific KPIs) changes dynamically. This requires rethinking the selection problem: rather than selecting clients, the system must select which slices each near-RT-RIC should process. Moreover, O-RAN control loops impose strict deadlines. So, client selection must ensure the selected subset can complete local training and upload updates within the deadline, accounting for heterogeneous processing capabilities and time-varying bandwidth. Furthermore, to maintain service quality, each network slice (eMBB, uRLLC, mMTC) must be adequately represented in each FL round. Existing selection methods do not incorporate slice-based stratification or quota constraints.

1.1.3 Communication Efficiency Techniques

Sparse gradients (many near-zero elements) can be compressed before transmission. Konečnỳ, McMahan, Ramage & Richtárik (2016) proposed structured and sketched updates, achieving 10–100× compression ratios with minimal accuracy loss. However, compression introduces additional computational overhead for encoding/decoding.

Representing gradient elements with fewer bits reduces transmission size. Reisizadeh, Mokhtari, Hassani, Jadbabaie & Pedarsani (2020) proposed FedPAQ (Federated Learning with Periodic Averaging and Quantization), demonstrating that 8-bit quantization achieves comparable accuracy to full-precision (32-bit) communication while reducing bandwidth by 75%.

Transmitting only the top-k largest gradient elements or applying random sparsification. Khaled & Richtárik (2019) analyzed gradient descent with compressed iterates, proving convergence for randomized compression operators satisfying unbiasedness and bounded variance conditions.

Splitting the model between edge and cloud, with only activations transmitted. This is effective for deep neural networks where intermediate activations are smaller than full gradients (Lim *et al.*, 2020).

Exploiting the superposition property of wireless channels to perform analog gradient aggregation, where multiple clients transmit simultaneously and the received signal is a weighted sum of transmitted gradients (Zhu *et al.*, 2019). This eliminates multiple access overhead but requires precise synchronization and power control.

Convergence analysis with compression has been studied extensively. Li, Kovalev, Qian & Richtárik (2020c) proved that for ω-compression operators (satisfying $\mathbb{E}[\|C(x) - x\|^2] \leq \omega\|x\|^2$), compressed gradient descent converges at rate $O((1+\omega)\frac{L}{\epsilon})$, where L is the smoothness constant. This indicates a graceful degradation: compression slows convergence proportionally to the compression error ω, but convergence is still guaranteed.

Research Gap: Existing work treats compression, client selection, and resource allocation as independent problems. In O-RAN, these decisions are tightly coupled: allocating more bandwidth to a client may obviate the need for aggressive compression, while selecting fewer clients allows higher compression ratios per client. No existing framework jointly optimizes all three dimensions.

1.2 Convergence Acceleration in Federated Learning

The foundational FedAvg algorithm (McMahan *et al.*, 2017) employs Stochastic Gradient Descent (SGD) for local training. Variants of SGD have been proposed to improve convergence in FL settings. Algorithms such as Adam (Kingma & Ba, 2014) and AdaGrad (Duchi, Hazan & Singer, 2011) adaptively adjust learning rates based on gradient history, accelerating convergence in the presence of sparse gradients or varying gradient magnitudes. However, adaptive methods can suffer from premature convergence in non-IID FL settings due to biased gradient estimation (Reddi *et al.*, 2020).

Exponential decay, step decay, and cosine annealing schedules reduce the learning rate over time to ensure convergence to sharper minima. However, optimal scheduling depends on problem characteristics (smoothness, strong convexity) and is challenging to tune in heterogeneous FL environments (Li *et al.*, 2020b).

Li *et al.* (2020b) augmented the local objective with a proximal term $\frac{\mu}{2}\|\mathbf{w} - \mathbf{w}^{(t)}\|^2$ to limit client drift in non-IID settings, where $\mathbf{w}^{(t)}$ is the global model. This regularization improves stability but does not fundamentally accelerate convergence.

Convergence analysis of first-order methods in FL typically assumes:

1. ρ-Lipschitz Continuity: $\|f(\mathbf{w}_1) - f(\mathbf{w}_2)\| \leq \rho\|\mathbf{w}_1 - \mathbf{w}_2\|$
2. β-Smoothness: $\|\nabla f(\mathbf{w}_1) - \nabla f(\mathbf{w}_2)\| \leq \beta\|\mathbf{w}_1 - \mathbf{w}_2\|$
3. Convexity or μ-Strong Convexity: The loss function is convex or satisfies $f(\mathbf{w}_2) \geq f(\mathbf{w}_1) + \nabla f(\mathbf{w}_1)^T(\mathbf{w}_2 - \mathbf{w}_1) + \frac{\mu}{2}\|\mathbf{w}_2 - \mathbf{w}_1\|^2$

Under these assumptions, Konečnỳ *et al.* (2016) proved that FedAvg with local accuracy θ requires $O(\frac{\log(1/\epsilon)}{(1-\theta)})$ global rounds to achieve ϵ-accuracy, indicating that convergence rate is inversely proportional to local training quality.

Second-order optimization methods utilize curvature information (Hessian matrix) to accelerate convergence. While exact second-order methods (Newton's method) are computationally prohibitive for high-dimensional models, quasi-Newton methods and momentum-based acceleration offer practical alternatives (Nocedal & Wright, 2006). Momentum Gradient Descent (MGD) augments gradient descent with a momentum term that accumulates past gradients:

$$\mathbf{v}^{(t)} = \gamma \mathbf{v}^{(t-1)} + \nabla f(\mathbf{w}^{(t-1)}) \tag{1.1}$$

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta \mathbf{v}^{(t)} \tag{1.2}$$

where $\gamma \in [0, 1)$ is the momentum coefficient and $\mathbf{v}^{(t)}$ is the velocity vector. Momentum acceleration dampens oscillations in gradient directions with high curvature and accelerates progress in low-curvature directions, achieving faster convergence than vanilla SGD (Sutskever, Martens, Dahl & Hinton, 2013).

Nesterov Accelerated Gradient (NAG) improves upon classical momentum by computing gradients at the "look-ahead" position:

$$\mathbf{v}^{(t)} = \gamma \mathbf{v}^{(t-1)} + \nabla f(\mathbf{w}^{(t-1)} - \gamma \mathbf{v}^{(t-1)}) \tag{1.3}$$

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta \mathbf{v}^{(t)} \tag{1.4}$$

NAG achieves optimal convergence rate $O(1/t^2)$ for smooth convex functions, compared to $O(1/t)$ for gradient descent (Nesterov, 1983).

Application of momentum methods to FL introduces challenges. Momentum can be applied at the server (aggregating velocity across clients) or at clients (maintaining local velocity). Server-side momentum is more stable but requires transmitting velocity in addition to gradients (Liu, Chen, Chen & Zhang, 2020). Client drift in non-IID settings can cause momentum

to accumulate in suboptimal directions. Liu *et al.* (2020) proposed Momentum Federated Learning (MFL) with adaptive momentum coefficients based on gradient similarity, achieving 25% faster convergence than FedAvg on heterogeneous data. Synchronous FL waits for all clients, and momentum accumulated from straggler gradients may be stale. Asynchronous momentum aggregation partially addresses this but introduces consistency challenges (Chen, Ning, Slawski & Rangwala, 2021b).

Convergence analysis of momentum-based FL has been established under strong convexity assumptions. For μ-strongly convex and L-smooth loss functions, momentum FL with appropriate parameter tuning converges linearly at rate:

$$\mathbb{E}[F(\mathbf{w}^{(T)})] - F^* \leq (1 - \sqrt{\mu/L})^T [F(\mathbf{w}^{(0)}) - F^*] \tag{1.5}$$

This represents a significant improvement over the sublinear rate of first-order methods, particularly when the condition number $\kappa = L/\mu$ is large (Liu *et al.*, 2020).

1.2.1 Compression with Convergence Guarantees

Combining compression with acceleration introduces additional complexity in convergence analysis. Khaled & Richtárik (2019) studied gradient descent with compressed iterates, proving that randomized ω-compression operators satisfying:

$$\mathbb{E}[C(\mathbf{x})|\mathbf{x}] = \mathbf{x} \quad \text{(unbiasedness)} \tag{1.6}$$

$$\mathbb{E}[\|C(\mathbf{x}) - \mathbf{x}\|^2] \leq \omega\|\mathbf{x}\|^2 \quad \text{(bounded variance)} \tag{1.7}$$

ensure convergence at rate $O((1+\omega)\frac{L}{\epsilon})$ for L-smooth convex functions.

Random Sparsification: A specific compression operator that retains k randomly selected elements and zeros out the rest:

$$C(\mathbf{x}) = \frac{d}{k}(\zeta_k \odot \mathbf{x}) \tag{1.8}$$

where $\zeta_k \in \{0,1\}^d$ is a random binary mask with k non-zero entries and d is the dimension. This operator satisfies the compression conditions with $\omega = \frac{d}{k} - 1$, meaning higher sparsification (smaller k) increases ω and slows convergence (Li *et al.*, 2020c).

Li *et al.* (2020c) extended this analysis to acceleration with compression, proving that Momentum Gradient Descent with Compressed Iterates (MGDCI) converges provided:

$$\omega \leq \frac{\mu}{4} \cdot \frac{1 - 2\gamma L}{2\gamma L^2 + \frac{2}{\gamma} + L - \mu} \tag{1.9}$$

where γ is the momentum coefficient. This establishes a fundamental tradeoff: aggressive compression (large ω) limits the maximum achievable momentum, constraining acceleration benefits.

In the FL context, compression interacts with aggregation in non-trivial ways. Horváth *et al.* (2022) analyzed natural compression in distributed gradient descent, showing that compressing the aggregated gradient (post-aggregation) yields better convergence than compressing individual gradients (pre-aggregation) due to variance reduction through averaging.

Research Gap: Existing theory analyzes compression and acceleration independently. In practice, these must be jointly optimized with resource allocation: allocating more compute to a client enables more local iterations (acceleration), while allocating more bandwidth allows less aggressive compression. No existing framework provides joint convergence guarantees under this coupled optimization. Different near-RT-RICs may apply different compression ratios and momentum coefficients based on their resource constraints and channel conditions. Convergence analysis typically assumes homogeneous parameters across all clients, which may be overly restrictive for heterogeneous O-RAN deployments.

1.3 Mobility Management in Federated Learning

1.3.1 Handover in 5G and O-RAN Networks

In 5G New Radio (NR) and O-RAN systems, handover procedures are more complex than in previous generations due to disaggregation of network functions and the introduction of network slicing (Kim *et al.*, 2021).

Handover Types:

- Intra-gNB Handover: UE moves between cells controlled by the same gNodeB (gNB). The O-DU and O-CU remain unchanged, minimizing service disruption.
- Inter-gNB-DU Handover: UE moves between cells controlled by different gNB-DUs while remaining under the same O-CU control. This is the most common handover type in O-RAN (Kim *et al.*, 2021).
- Inter-gNB Handover: UE moves between cells controlled by different gNBs, requiring coordination with the 5G Core (5GC) to update user plane paths.

The inter-gNB-DU handover procedure specified by 3GPP TS 38.300 (3GPP, 2018) and O-RAN specifications (O-RAN Alliance, 2019) involves:

1. Measurement Phase: Source gNB-DU configures UE to measure neighboring cells. UE reports measurements (RSRP, RSRQ, SINR) when conditions specified by Time-to-Trigger (TTT) and hysteresis parameters are met.
2. Decision Phase: O-CU-CP evaluates measurement reports and decides whether to initiate handover based on optimization algorithms (ML-based decision-making can be employed here via xApps (Prananto, Iskandar & Kurniawan, 2023)).
3. Preparation Phase: Source gNB-DU sends handover request to target gNB-DU via X2/Xn interface. Target prepares resources and admits the UE.
4. Execution Phase: Source gNB-DU commands UE to perform handover. UE synchronizes with target cell and sends handover complete message.
5. Path Switch Phase: For inter-gNB handovers, user plane path is updated in the 5GC (UPF).

Handover delay, defined as the time interval from handover decision to handover completion, typically ranges from 30–100ms in LTE and 15–50ms in 5G NR (Karmakar, Kaddoum & Chattopadhyay, 2023). This delay comprises:

$$T_{HO} = T_{meas} + T_{proc} + T_{exec} + T_{path} \tag{1.10}$$

where T_{meas} is measurement reporting delay, T_{proc} is processing delay, T_{exec} is execution delay (includes UE synchronization and RACH procedure), and T_{path} is path switch delay (for inter-gNB handover).

In O-RAN, handover decisions can be optimized through ML-driven xApps in the Near-RT-RIC (Prananto *et al.*, 2023). Adaptive TTT and hysteresis parameters reduce unnecessary handovers (ping-pong effect) while ensuring timely handovers before link quality degrades excessively. However, these optimizations must account for the impact of mobility on ongoing FL training.

1.3.2 Impact of Mobility on Distributed Learning

In FL scenarios where end-user devices serve as clients, device availability changes due to mobility (devices moving out of coverage), battery depletion, and application lifecycle (user closing the app). This dynamic participation affects model convergence, as clients with unique data distributions may drop out before contributing sufficiently (Bonawitz *et al.*, 2019). In location-aware applications (e.g., traffic prediction), a device's data distribution changes as it moves between geographic regions. This spatial non-stationarity can cause catastrophic forgetting if models are not updated to reflect new environments (Smith, Chiang, Sanjabi & Talwalkar, 2017).

Handover execution causes temporary service interruption (typically 10–50ms), during which communication between client and server is suspended. If handover occurs during model update transmission, the update may be lost or delayed, causing synchronization issues in synchronous FL (Bonawitz *et al.*, 2019). After handover, a UE may experience different channel conditions

(path loss, interference) and different load at the target cell, affecting available bandwidth and expected communication latency (Noor-A-Rahim *et al.*, 2022).

In the O-RAN FL context, where near-RT-RIC instances serve as FL clients, mobility has a different manifestation. When a UE hands over from source gNB-DU to target gNB-DU, the RL agent (xApp) managing that UE may need to be reassigned. If each near-RT-RIC instance trains a local model based on UEs it currently serves, handover changes the data distribution available to each near-RT-RIC (Cao, Lien, Liang, Chen & Shen, 2022). Network slicing adds complexity: a UE belonging to a uRLLC slice must maintain QoS guarantees during and after handover. The FL model training for that slice should account for temporary unavailability of data from the handed-over UE (Karmakar *et al.*, 2023). In hierarchical FL (HFL), where edge aggregators perform local aggregation before sending to the central server, UE handover may cause edge aggregator reassignment. This disrupts the hierarchical aggregation structure and may require re-initializing local aggregation state (Liu *et al.*, 2022).

Unlike synchronous FL, which waits for all selected clients to complete local training, asynchronous FL allows clients to upload updates at any time without synchronization (Xie, Koyejo & Gupta, 2019). This naturally handles client dropouts due to mobility, as the server aggregates whenever updates arrive. However, asynchronous aggregation introduces staleness: updates from slow clients may be based on outdated global models, slowing convergence (Chen *et al.*, 2021b).

Prioritizing clients with stable connections over mobile clients reduces the probability of communication failures. However, this introduces bias, as mobile clients may have geographically diverse data that is underrepresented (Nishio & Yonetani, 2019).

Feng *et al.* (2022) proposed Mobility-Aware Cluster Federated Learning (MACFL) for hierarchical wireless networks. Clients are clustered based on mobility patterns, and each cluster has a dedicated edge aggregator. When a client hands over, it switches to the edge aggregator of its new cluster. This approach reduces handover impact by localizing aggregation, but clustering is static and may not adapt to dynamic mobility patterns.

Zhao, Li, Li, Yang & Xu (2025) proposed FedPPO, using Proximal Policy Optimization to learn client selection policies that account for mobility, channel conditions, and data quality. The RL agent learns to avoid selecting clients likely to experience handover during the training round, reducing communication failures. However, FedPPO requires extensive training data (mobility traces) and may not generalize to unseen mobility patterns.

Proactive resource allocation based on predicted handovers can reserve resources at the target cell in advance, reducing handover delay (Karmakar *et al.*, 2023). In the FL context, predicting which clients will hand over allows the aggregator to adjust client selection or reallocate bandwidth preemptively.

Research Gap: Existing methods treat mobility as a binary availability problem (client present or absent) without incorporating the continuous handover delay T_{HO} into the learning time optimization. For O-RAN applications with strict control loop deadlines (10ms–1s), handover delays of 30–100ms constitute a significant fraction of the time budget. There is no formal optimization framework for deciding whether UEs that have undergone handover should: (i) continue participating in the current FL round (introducing handover delay but preserving data diversity), (ii) be dropped from the current round (sacrificing data but avoiding delay), or (iii) be reassigned to different local trainers (balancing load across near-RT-RICs). When a UE hands over between gNB-DUs, its associated near-RT-RIC changes. Existing HFL frameworks do not address: (i) how to handle partial local aggregations at the source near-RT-RIC, (ii) whether to restart local training at the target near-RT-RIC, or (iii) how to maintain global model consistency across handovers. Handover decisions interact with resource allocation: allocating more bandwidth to a UE reduces transmission delay, potentially enabling update completion before handover. Conversely, predicting imminent handover may justify reduced resource allocation to that UE. No existing work jointly optimizes these coupled decisions.

1.4 Orchestration of Multiple Federated Learning Services

Modern edge computing platforms host multiple concurrent services with heterogeneous requirements, including latency-sensitive applications (augmented reality, autonomous driving), throughput-intensive applications (video streaming, bulk data transfer), and compute-intensive applications (ML inference, data analytics) (Mao, You, Zhang, Huang & Letaief, 2017). When multiple FL services coexist on shared infrastructure, resource contention arises across multiple dimensions. Edge servers have limited CPU/GPU capacity that must be shared among concurrent FL tasks for local model training and aggregation (Xu *et al.*, 2023b). Wireless bandwidth is shared among FL services transmitting model updates, introducing queuing delays and potential packet loss (Xu, Wang & Chen, 2022a). Different FL services support applications with conflicting QoS requirements. For example, a uRLLC FL service requires sub-10ms latency, while an eMBB FL service prioritizes throughput over latency. Satisfying one service's requirements may violate another's (Qureshi, Manalastas, Zaidi, Imran & Al Kalaa, 2021). FL service providers and infrastructure providers establish SLAs specifying performance guarantees (model accuracy, training time, resource cost). Violating SLAs incurs penalties, necessitating careful resource allocation (Nguyen, Tran, Tun, Han & Hong, 2023b).

Cheng *et al.* (2022b) proposed an auction-based framework for multiple FL services in UAV-aided networks, where FL services bid for UAV resources (bandwidth, transmit power, hovering time) based on their utility functions. A Vickrey-Clarke-Groves (VCG) auction ensures truthfulness (services bid their true valuations) and achieves social welfare maximization.

Fixed priorities are assigned to services based on their criticality. For example, uRLLC services receive higher priority than eMBB services (Hamdan *et al.*, 2023a). Resources are allocated greedily: highest-priority services are served first, and remaining resources are allocated to lower-priority services. While simple and widely used in 5G RAN scheduling, priority-based approaches have drawbacks.

Resources are divided into fixed partitions, with each FL service assigned to a partition. This ensures isolation (no resource contention) but is inflexible: resources allocated to idle services cannot be reallocated to active services (Xu *et al.*, 2023b).

Nguyen *et al.* (2023b) applied proportional fairness to multiple FL services in mobile edge networks, formulating a two-stage optimization: (i) FL service association (assigning services to edge servers), and (ii) resource allocation within each edge server. However, proportional fairness does not explicitly enforce QoS constraints and may still lead to violations for latency-sensitive services.

Research Gap: Existing approaches optimize individual aspects (e.g., auction for bandwidth allocation, separate algorithm for compute allocation) without jointly considering the tight coupling between task assignment, resource allocation, and routing decisions. In O-RAN, these decisions are interdependent: assigning a task to a particular near-RT-RIC determines the required communication path (routing), which in turn affects available bandwidth and communication delay, which impacts the required compute allocation to meet deadlines. For iterative orchestration schemes, where resource allocations are refined based on performance feedback from previous rounds, it is critical to prove: (i) the algorithm converges to a feasible solution, (ii) the solution quality (optimality gap) is bounded, and (iii) the convergence rate is acceptable for practical deployment. Existing work provides empirical validation but not rigorous mathematical guarantees. Orchestrators should leverage *closed-loop feedback* that monitors actual achieved performance (model accuracy, QoS, cost) and adapting future allocations accordingly. While conceptually straightforward, integrating feedback into optimization frameworks while maintaining convergence guarantees is non-trivial and remains largely unexplored.

1.5 Summary of Research Gaps

This literature review has identified critical gaps across multiple dimensions of FL deployment in O-RAN systems. Table 1.1 synthesizes these gaps and maps them to the research directions pursued in this thesis.

Together, these contributions establish a systematic foundation for deploying Federated Learning in Open RAN systems, enabling intelligent, privacy-preserving, and resource-efficient optimization of next-generation mobile networks.

Table 1.1 Summary of Research Gaps and Thesis Contributions

Research Area & Key Works	Identified Gaps	Thesis Contribution
O-RAN FL Integration with Communication Efficiency (Dinh *et al.*, 2021; Chen *et al.*, 2021a; Wang *et al.*, 2020; Mo & Xu, 2021; Yang *et al.*, 2020; Li *et al.*, 2020c; Khaled & Richtárik, 2019; Wang, Chen, Wang & Yang, 2024b; Zhuansun *et al.*, 2024; Koursioumpas *et al.*, 2024; Xu *et al.*, 2023a; Luo *et al.*, 2023)	No systematic framework for FL deployment within RIC hierarchy	Slice-aware FL framework integrated with Near-RT-RIC and Non-RT-RIC
	Lack of deadline-aware resource allocation for control loops	Joint optimization respecting control loop deadlines
	Missing slice-specific differentiation in FL training	Slice-based stratified local trainer selection with priority scheduling
	Compression and acceleration not jointly optimized	Momentum-based FL with randomized compression (MCORANFed)
	No convergence analysis for compressed momentum FL	Convergence rate $O((1+\omega)\frac{L}{\epsilon})$ under O-RAN constraints
	Heterogeneous compression across clients not addressed	Client-adaptive compression based on channel conditions
Mobility Management (Karmakar *et al.*, 2023; Kim *et al.*, 2021; Cao *et al.*, 2022; Feng *et al.*, 2022; Zhao *et al.*, 2025)	Handover delays not explicitly modeled in FL learning time	HO delay T_{HO} integrated into hierarchical FL formulation
	No strategic framework for post-handover trainer retention	Joint optimization of trainer participation and resource allocation
	Missing hierarchical aggregation under UE mobility	Edge aggregator reassignment with state transfer mechanisms
	Slice-aware handover priorities not considered	Slice-differentiated handover management (uRLLC priority)
Multi-Service Orchestration (Yu, Wang & Luan, 2023a; Qureshi *et al.*, 2021; Nguyen *et al.*, 2023b; Cheng *et al.*, 2022b; Hamdan *et al.*, 2023a)	Lack of holistic optimization for task assignment, resource allocation, routing	Two-stage decomposition (TSA + RAR) with joint optimization
	No dynamic lifecycle management (instantiation, preemption, termination)	O-FL rApp framework with task preemption based on QoS violations
	Missing convergence guarantees for iterative orchestration	Geometric convergence proof with bounded steady-state error
	Performance feedback integration not formalized	Closed-loop orchestration with measurement noise handling

CHAPTER 2

METHODOLOGIES

This chapter presents the comprehensive methodological framework employed throughout this thesis to address the identified research challenges in deploying Federated Learning within Open Radio Access Networks. The methodology follows a systematic approach across all chapters while employing specific optimization techniques tailored to each research problem.

2.1 General Research Methodology (M0)

This thesis follows a systematic six-stage research methodology applied consistently across all chapters to ensure rigor and reproducibility. The general methodology (M0) provides the foundational framework for addressing each subproblem:

M0.1: Literature Analysis and Gap Identification: Carrying out a thorough analysis of the existing literature work and methods related to each research question. This stage involves comprehensive review of state-of-the-art approaches, identification of their limitations, assessment of performance gaps, and determination of possible improvements that motivate the proposed contributions.

M0.2: Mathematical Problem Formulation: Based on the findings from literature analysis, particularly the identified limitations and improvement opportunities, we formulate each problem mathematically and define the proposed models. Problem formulation is crucial in this work since the same problems can be formulated and solved using different learning models, optimization frameworks, and constraint structures.

M0.3: Solution Design and Theoretical Analysis: Based on the mathematical model, we propose suitable solution approaches and conduct rigorous convergence analysis, optimality analysis, and complexity evaluation. This stage includes algorithm design, theoretical guarantees establishment, and performance bound derivation.

M0.4: Experimental Design and Baseline Definition: We define comprehensive evaluation metrics, establish appropriate baseline methods for comparison, design experimental scenarios, and configure simulation environments that capture the essential characteristics of O-RAN deployments.

M0.5: Results Analysis and Interpretation: We analyze numerical and visual outcomes through statistical evaluation, interpret observations in the context of research objectives, discuss the significance of findings, validate theoretical predictions, and draw rational inferences about the practical applicability of proposed solutions.

M0.6: Publication and Dissemination: Finally, we present the work organized into manuscript form and share findings at international conferences and journals to contribute to the broader research community and obtain peer feedback for further improvements.

2.2 Chapter-Specific Methodologies

2.2.1 Methodology (M1): Communication Efficient FL for RO1

Chapter 3 addresses the research sub-objective RO1 for an efficient resource management and communication optimization for a single FL task in O-RAN. We design a basic framework named ORANFed by formulating a joint local trainers' selection and resource optimization problem. We then design an improved communication efficient problem through a comprehensive methodology incorporating multiple optimization techniques as follows:

M1.1: Mixed-Integer Programming Formulation: The joint trainer selection and resource allocation problem is formulated as a Mixed-Integer Non Linear Problem (MINLP) where binary variables represent trainer selection decisions and continuous variables represent compute and bandwidth allocations. The non-convex objective combining training time minimization and resource cost is addressed through problem decomposition.

M1.2: Successive Convex Approximation (SCA): Non-convex constraints arising from wireless channel models and FL convergence relationships are linearized using first-order Taylor approximations. The SCA framework ensures monotonic improvement while maintaining computational tractability.

M1.3: Two-Stage Decomposition: The optimization problem is decomposed into strategic trainer selection (Stage 1) and tactical resource allocation (Stage 2). This decomposition exploits the hierarchical structure of O-RAN control loops where selection decisions are made at longer timescales than resource allocation decisions.

M1.4: Momentum-Based Optimization with Compression: A novel momentum gradient descent algorithm is developed that integrates randomized compression techniques. The algorithm maintains momentum vectors while applying compression operators that satisfy unbiasedness and bounded variance conditions for convergence guarantees.

M1.5: Convergence Analysis: Theoretical convergence rates are derived for the compressed momentum algorithm under non-IID data distributions. The analysis establishes $O((1+\omega)\frac{L}{\epsilon})$ convergence rate where ω is the compression parameter and L is the smoothness constant.

M1.6: Slice-Aware Trainer Selection: A stratified sampling approach ensures representation of all network slices (eMBB, uRLLC, mMTC) in each training round while respecting computational and communication constraints. Priority weights are assigned based on slice-specific QoS requirements.

2.2.2 Methodology (M2): Mobility-Aware Hierarchical FL for RO2

Chapter 4 extends the ORANFed framework from M1 to a mobility-aware hierarchical FL that maintain stability under varying participant sets caused by handovers. Here we employ the following methods to address the research sub-objective RO2:

M2.1: Semidefinite Programming for Handover Optimization: The joint optimization of trainer participation and resource allocation under handover constraints is formulated as a

quadratically constrained quadratic program (QCQP) and relaxed to a Semi Definite Program (SDP). This approach handles the combinatorial complexity of handover-induced topology changes while providing tractable solutions.

M2.2: Lagrangian Decomposition for Hierarchical Aggregation: The hierarchical FL problem is decomposed using Lagrangian duality where edge aggregators optimize local training decisions while coordinating through dual variables. This enables distributed optimization while maintaining global convergence guarantees.

M2.3: Fixed-Point Theory for Convergence Analysis: The iterative aggregation process under mobility is analyzed using fixed-point theory. Contraction mapping properties are established for the aggregation operator, ensuring geometric convergence despite participant changes.

M2.5: Hierarchical Aggregation Protocols: Novel aggregation protocols are designed for the two-tier O-RAN architecture where Near-RT-RIC instances perform local aggregation before communicating with Non-RT-RIC. State transfer mechanisms handle participant migration between aggregators.

M2.6: Strategic Trainer Retention: Decision criteria are developed for determining whether UEs that have undergone handover should continue participating in the current FL round, be dropped to avoid delays, or be reassigned to different trainers based on QoS requirements and learning progress.

2.2.3 Methodology (M3): Multi-Task Orchestration for RO3

Chapter 5 extends the single FL task scenario in previous chapters to a system-level resource orchestration for multiple concurrent Federated tasks. On top of utilizing M1 and M2 for FL modeling in O-RAN system, we address the research sub-objective RO3 by employing the following methods:

M3.1: Multi-Objective Optimization: The orchestration problem is formulated as a multi-objective optimization balancing conflicting goals including resource cost minimization, training

time reduction, and QoS violation prevention. Pareto-optimal solutions are computed using weighted sum and ϵ-constraint methods.

M3.4: Geometric Convergence Analysis: The iterative orchestration scheme is analyzed to establish geometric convergence rates. Contraction properties of the resource allocation operator ensure rapid convergence to steady-state solutions with bounded approximation errors.

M3.5: Varying Resource Scheduling: Adaptive scheduling algorithms are developed that adjust resource allocations based on real-time performance feedback. The algorithms handle task lifecycle events including instantiation, preemption, suspension, and termination while maintaining system stability.

M3.6: rApp Framework Implementation: The orchestration solution is implemented as an O-RAN rApp hosted in the Non-RT-RIC, leveraging standardized interfaces (A1, E2, O1) for communication with Near-RT-RIC instances and service management functions.

2.3 Thesis Roadmap

This chapter has established the comprehensive methodological foundation for addressing the research challenges identified in this thesis. The systematic approach combines rigorous mathematical optimization techniques with practical implementation strategies and thorough experimental validation. The general methodology (M0) provides a consistent six-stage framework applied across all chapters, while the chapter-specific methodologies (M1, M2, M3) detail the targeted optimization approaches employed for each research problem. The common system modeling framework ensures consistency across contributions while allowing for problem-specific extensions and refinements.

The subsequent technical chapters build upon these methodological foundations, applying the described optimization techniques and evaluation frameworks to develop novel solutions for communication-efficient FL, mobility-aware hierarchical learning, and multi-task orchestration in O-RAN systems. A thesis diagram is presented in Figure 2.1.

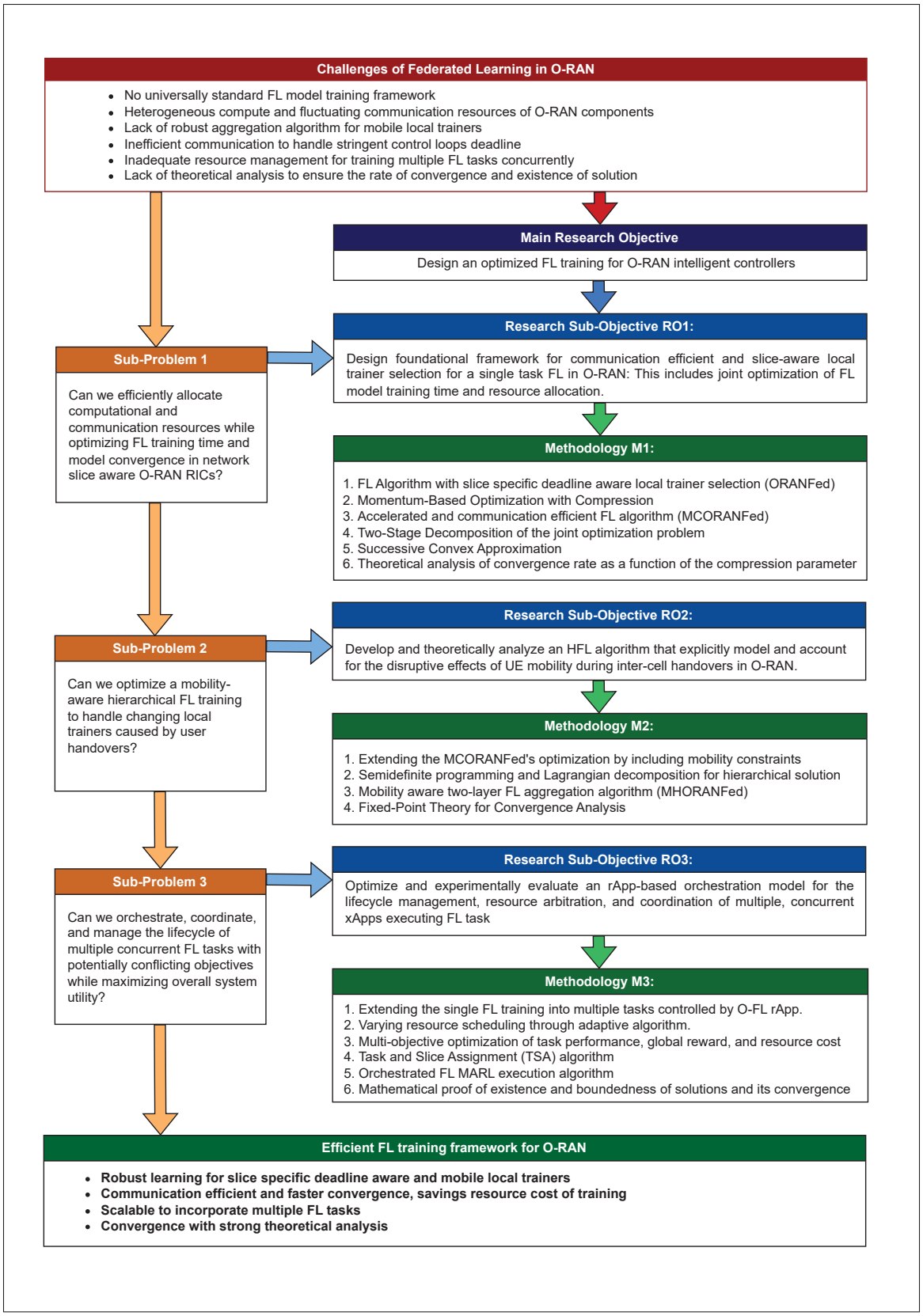


Figure 2.1 Thesis roadmap and research element mapping

CHAPTER 3

COMMUNICATION EFFICIENT COMPRESSED AND ACCELERATED FEDERATED LEARNING IN OPEN RAN INTELLIGENT CONTROLLERS

Amardip Kumar Singh[1] , Kim Khoa Nguyen[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in *IEEE/ACM Transactions on Networking*, April 2024

Abstract

The disaggregated and hierarchical architecture of Open Radio Access Network (ORAN) with openness paradigm promises to deliver the ever demanding 5G services. Meanwhile, it also faces new challenges for the efficient deployment of Machine Learning (ML) models. Although ORAN has been designed with built-in Radio Intelligent Controllers (RIC) providing the capability of training ML models, traditional centralized learning methods may be no longer appropriate for the RICs due to privacy issues, computational burden, and communication overhead. Recently, Federated Learning (FL), a powerful distributed ML training, has emerged as a new solution for training models in ORAN systems. 5G use cases such as meeting the network slice Service Level Agreement (SLA) and Key Performance Indicator (KPI) monitoring for the smart radio resource management can greatly benefit from the FL models. However, training FL models efficiently in ORAN system is a challenging issue due to the stringent deadline of ORAN control loops, expensive compute resources, and limited communication bandwidth. Moreover, to deliver Grade of Service (GoS), the trained ML models must converge with acceptable accuracy. In this paper, we propose a second order gradient descent based FL training method named MCORANFed that utilizes compression techniques to minimize the communication cost and yet converges at a faster rate than state-of-the-art FL variants. We formulate a joint optimization problem to minimize the overall resource cost and learning time, and then solve it by the decomposition method. Our experimental results prove that MCORANFed is communication efficient with respect to ORAN system, and outperforms FL methods like MFL, FedAvg, and ORANFed in terms of costs and convergence rate.

keywords

federated learning, ORAN, 5G, resource allocation, RAN intelligent controller, network slicing, RIC

3.1 Introduction

Due to an exponential growth in high performing user devices and a massive increase in connectivity, the demand of a reliable and scalable data network is ever increasing. To meet this demand and realise the envision of 5G and beyond communications, leading telecommunication infrastructure providers have taken many initiatives and instituted focused research studies to redesign their network management systems (Ericsson, 2021). A major part of this ongoing study concentrates on improving the capability of Radio Access Network (RAN). In this direction, the O-RAN Alliance has already published the specifications of Open RAN (ORAN) which is particularly based on two key concepts: (i) openness and (ii) in-built intelligence (O-RAN Alliance, 2019). ORAN frees the network from vendor lock-in and disaggregates the processing layers from proprietary hardware to shared open clouds. On the other hand, it also calls for a smarter Service and Management Orchestration (SMO) by installing intelligent controllers which utilizes the operational and performance data to optimize the RAN resources (Sun *et al.*, 2021).

With its unique features, ORAN can support many use cases such as Quality of Experience (QoE) optimization, ensuring Service Level Agreements (SLA) agreements, providing slice based ultra Reliable Low Latency Communications (uRLLC) and enhanced Mobile Broadband (eMBB) services, and monitoring Key Performance Indicators (KPIs) which are intrinsic part of 5G and beyond network (Costa-Perez, Swetina, Guo, Mahindra & Rangarajan, 2013). All of these use cases require training of fast and secure Machine Learning (ML) models. However, a traditional centralized ML model suffers from two important limitations: (i) transferring locally collected data to a central server and (ii) bandwidth usage cost incurred in transferring these data. Further, the privacy issues of shared processing open clouds and security of raw data

pose additional challenges in the paradigm of ORAN architecture (Maternia *et al.*, 2016). Due to this reason, we turn to Federated Learning (FL) which is becoming an advanced ML tool in the recent times (Lim *et al.*, 2020). The basic idea of FL is to train an ML model without sharing the locally collected raw data and communicating only its model updates to a distant server (McMahan *et al.*, 2017) to iteratively improve the final model so trained. In this training, an iterative gradient descent method is used to minimize the loss function, which is the main objective of any ML model. FL, being in its nascent stage, is still being studied and improved progressively by researchers across the domains (Zhao *et al.*, 2020). However, there is hardly any standard version of it which can adhere to ORAN architectural requirements and therefore stands as an open problem (Lim *et al.*, 2020). The main questions raised within this set up are: (i) How to allocate the ORAN resources required to train such a model?; (ii) How to meet the stringent deadlines of ORAN control loops that automates the performance of the network while training an FL model?; and (iii) How to guarantee the convergence of this model with accuracy?

Recent works on FL focus on adaptive resource allocation policies according to their mobile edge-cloud system architecture (Lim *et al.*, 2020). In general, their FL models try to improve (i) model accuracy, and (iii) energy efficiency of user end devices serving as FL clients. The network slicing (NS) based hierarchical architecture of ORAN imposes additional challenges such as constrained learning time and communication costs for FL training within the RICs. Moreover, the control loop deadline varies with respect to the performance of each network slice. ORAN also deals with multiple carriers' networks, resulting in bandwidth fluctuation in various links. Hence, an FL deployment model designed specifically for ORAN is required to fill this research gap. In this paper, we first propose an accelerated iteration method that utilizes a random sparsification compression operator to optimize the communication resources. Then, we derive an updated FL algorithm by incorporating slice specific and deadline aware selection of local trainers (Singh & Nguyen, 2022b). Our main contributions can be summarized as follows:

1. A second order momentum gradient descent based novel FL training method that utilizes a compression model updates in each global round.
2. A mathematical proof of the rate of convergence of this method.

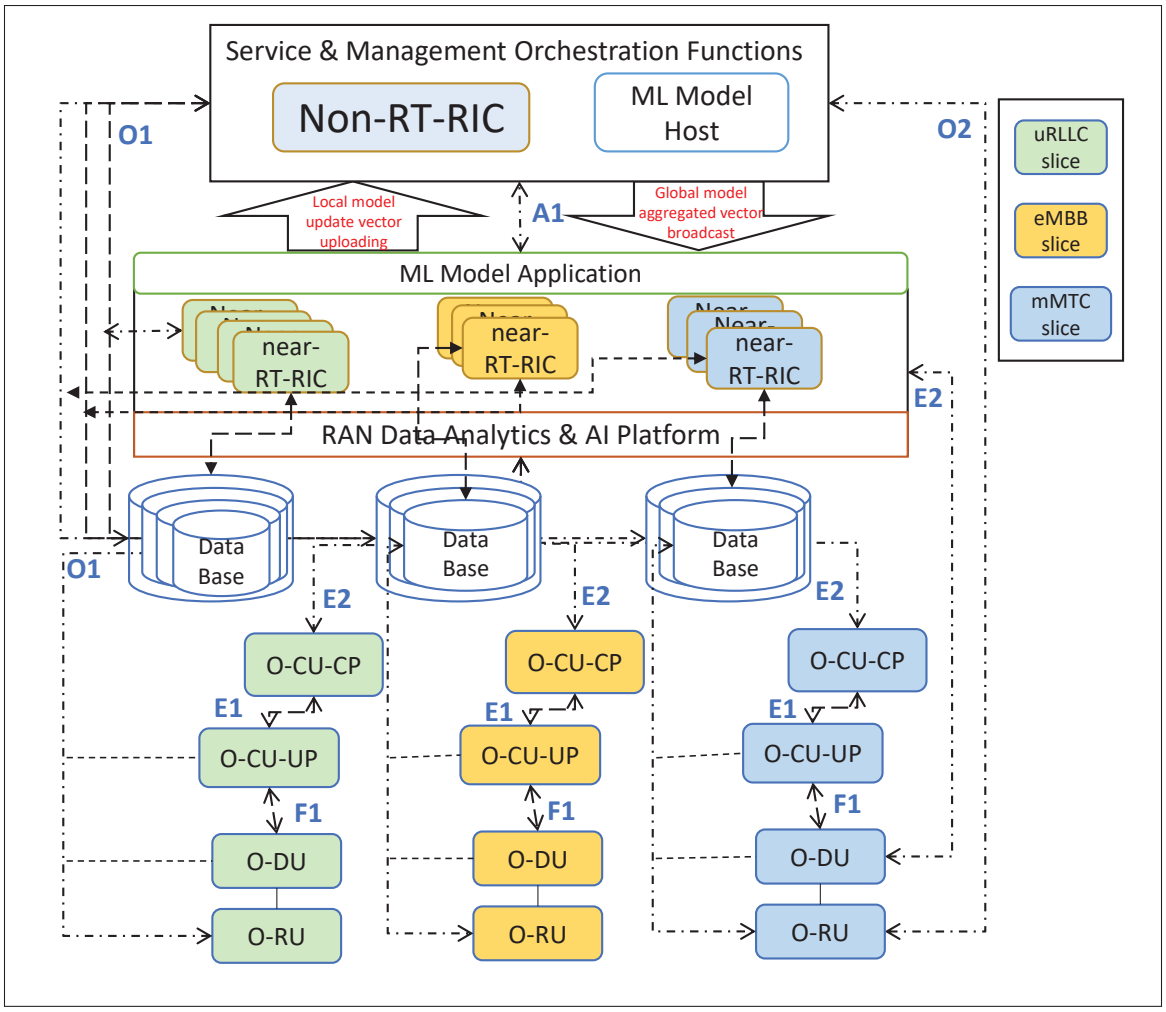


Figure 3.1 O-RAN Architecture for Radio Intelligent Controllers based FL

3. A mathematical formulation of the joint optimization problem to minimize communication resource usage cost and total learning time under the constraints of limited bandwidth, selected local trainers, and compression parameter.
4. A solution of this problem by decomposition method. We first solve for a near optimal set of local trainers and then find the optimal allocation of compute and bandwidth resources to this set of selected trainers.
5. An updated FL algorithm, so called MCORANFed (Momentum Compressed ORAN FL) to train the model through several iterations.

6. An analysis of results obtained from extensive experiments that validates our proposed method compared to the state-of-the-art FL variants.

To the best of our knowledge, this is the first work that takes into account acceleration and compression techniques to optimize ML training through federated settings in ORAN. The remainder of this paper is organized as follows. In Section 3.2, a summary of the related works is presented. Then, we briefly describe the concerned ORAN architecture for the FL setup in section 3.3. In Section 3.4, the considered system model and the problem formulation are presented. In Section 3.5, we describe MCORANFed, and in Section 3.6, we present our proposed solution approach. In Section 3.7, we present the numerical results to evaluate the performance of our proposed solution. Finally, we conclude the paper and discuss our future work in section 3.8.

3.2 Related Works and Challenges

McMahan *et al.* (2017) introduced the initial concept of federated averaging to train deep learning model in a distributed setting. It also proposed a method to select a fraction of clients instead of all the local trainers. While it solves the challenges of privacy by not sharing the raw data with the aggregator, the communication efficiency is still questionable. Based on this work, several variants of FL such as (Chen *et al.*, 2021a), (Yang *et al.*, 2021), (Dinh *et al.*, 2021), (Wang *et al.*, 2019a), and (Mo & Xu, 2021) have been proposed recently to meet the requirements of mobile edge-cloud architecture. (Chen *et al.*, 2021a) and (Mo & Xu, 2021) dealt with the energy efficiency optimization problems where a joint problem is solved under limited compute resources. Latency constraints are considered in (Yang *et al.*, 2021) to minimize the computation and transmission energy of user devices. Through approximated and closed form solutions to the resource allocation problem, it assigns optimal processing power for the local training. The authors in (Dinh *et al.*, 2021) have considered a wireless communication scenario to train an FL model under synchronous mode of communication between the model update aggregator and local devices. The control algorithm in (Wang *et al.*, 2019a) regulates the number of global and local iterations required to attain a desired level of model accuracy. In all

of these FL variants, the local model is processed at the user end devices. Their main objective was to minimize the energy cost incurred while training the local model. However, in ORAN system, the main objective should be minimizing the learning time to boost the performance of FL training itself. Nonetheless, it laid the foundation of mathematically formulating a resource allocation problem for FL in mobile edge-cloud architecture.

Table 3.1 Comparison with existing literature on FL for wireless networks

Work	**System Participants**	**Objective(s)**	**Client Selection**	**Compute Resource Alloc.**	**Communication Compression**	**Conv. Analysis**
McMahan *et al.* (2017)	UEs, BS	Model Loss	✗	✗	✗	✗
Wang *et al.* (2019a)	Edge nodes, Remote cloud	Model Loss	✗	✗	✗	✓
Yang *et al.* (2021)	UEs, BS	UEs' Energy	✗	✓	✗	✓
Dinh *et al.* (2021)	UEs, BS	UEs' Energy, Learning Time	✗	✓	✗	✓
Yang *et al.* (2020)	UEs, Access Point	Client Scheduling Policies	✓	✗	✗	✗
Xu & Wang (2021)	Edge Devices, Central Server	Model Convergence	✓	✗	✗	✓
Our Work	**O-RAN RICs**	**Learning Time, Bandwidth Cost**	✓	✓	✓	✓

In prior works: Yang *et al.* (2020), Xu & Wang (2021), Amiri, Gündüz, Kulkarni & Poor (2021), Shi, Zhou, Niu, Jiang & Geng (2021), and Luo, Chen, Wu, Zhou & Yu (2020), the authors have tackled another aspect of FL performance. They focus on optimizing the number of local trainers in each global round of FL. There are multiple approaches to deal with this aspect such as Round Robin Scheduling, Proportional Fair Scheduling, and Random Scheduling. Yang *et al.* (2020) investigated the role of client scheduling and user association problems of FL training.

Luo *et al.* (2020) proposed a hierarchical FL algorithm in single edge cloud system. Xu & Wang (2021) takes into account the channel states as well to optimize the client selection. All of the these works considered that the set of end users that trains the local models is fixed. However, in ORAN system, the ML model is not trained by end users. Rather, the edge node dataset keeps on changing and only the set of local training point remains fixed (O-RAN Alliance, 2020).

Table 3.1 summarizes the key differences of our work when compared with related works in terms of system model participants, defined objectives, consideration of local trainers' selection, resource allocation, and convergence. Since the simple first order iterative gradient descent method is commonly used in all prior works, their training methods are not significantly efficient in terms of the learning time. Unlike these prior works, we propose a novel FL training method that suits the architectural requirements of ORAN.

3.3 FL Training in O-RAN RICs

FL training within the O-RAN RICs is a distributed and privacy-preserving approach to machine learning tailored for the complexities of wireless networks. In this context, FL allows for the collaborative training of ML models through O-RAN's standard interfaces and protocols. These interfaces facilitate the transfer of local model updates between the near Real-Time Radio Intelligent Controllers (near-RT-RICs) and Non Real-Time Radio Intelligent Controller (Non-RT-RIC) in each global round. As illustrated in Figure 3.1, dedicated databases that collect the slice PM from xApps of the respective near-RT-RICs are connected to the rApp that trains the FL model in the Non-RT-RIC. These xApps can be hosted by rented Virtual Machines (VMs) on pay-per-usage model (Alliance, 2021b). A dedicated management plane (m-Plane) fiber link can be used for FL update transfer between the set of near-RT-RICs and the Non-RT-RIC. This m-Plane is a unique feature of ORAN that enables ORAN's SMO layer to design intelligent policies (Lee, Cha, Kwon, Jeong & Park, 2020). Training an FL model using the standard interfaces of O-RAN and RICs involves the following steps:

- **Data Collection and Distribution:** E2, which is a logical interface connecting near-RT-RIC with O-DU, O-CU-UP, and eNB RAN, is used to collect measurement and operational data

within the RAN (Polese *et al.*, 2023a). This data is stored in Radio Network Information Base (RNIB) and shared by multi-vendor edge clouds.

- **Model Initialization:** The initial untrained model parameters are set on the near-RT-RIC hosts. This model is trained by iteratively improving its performance metric (e.g. accuracy) through local and global rounds of FL.
- **Model Training:** Each near-RT-RIC instance trains its local model using the data available to it via the O1 interface, which is responsible for collecting differentiated traffic performance measurements (PM) for each RAN slice. Each O-RAN slice is composed of an ordered chain of virtual network functions spanning over one or more O-RUs, O-DUs, and O-CU-UPs (Singh & Nguyen, 2022a).
- **Model Aggregation:** The A1 interface can be used for the transfer of updates from the near-RT-RIC (xApps) to the Non-RT-RIC (rApp) at a regional cloud server. This interface is essential for updating the global model with the aggregated model updates from each local trainers.

By leveraging FL within the O-RAN RICs, the network operator can dynamically predict the QoE or QoS performance of ongoing applications and adapts RAN behaviors to maintain the user experience (Alliance, 2021a). This predictive modelling is important to guarantee the SLA of the slicing services.

3.4 System Model and Problem Formulation

We consider an ORAN system with a single regional cloud and a set $\mathcal{M}$ of M distributed edge clouds cooperatively performing an FL training model. The slice specific segregated PM data is collected into data lakes associated with at least one instance of near-RT-RIC at an edge cloud as illustrated in Figure 1. We denote this dataset by D_i as the PM data of i^{th} near-RT-RIC instance. In order to collect these data, we exploit the following hierarchical relation among O-RU, O-DU,

and O-CU as (0.a) and (0.b) respectively:

$$\sum_{v \in \{DU\}} \delta_{nv} \leq 1 \quad \forall n \in \{RU\}$$
$$\sum_{u \in \mathcal{M}} \zeta_{mu} \leq 1 \quad \forall m \in \{CU\}.$$

Here, (0.a) imposes the condition a single O-RU can associate with only one O-DU and (0.b) enforces that one O-DU can associate with a single O-CU.

As illustrated in Figure 3.2, in this FL setup each edge cloud node hosts a near-RT-RIC entity that processes its locally collected data to train a local FL model. The Non-RT-RIC placed at the regional cloud integrates the local FL models from participating edge clouds and generates an aggregated FL model. This aggregated FL model is further used to improve local FL models of each near-RT-RIC enabling the local models to collaboratively perform a learning algorithm without transferring its raw training data. We call this aggregated FL model so generated by using the local FL models as the global FL model. The uplink from near-RT-RICs to the Non-RT-RIC is used to send the local FL model update parameters and the downlink is used to broadcast the global FL model in global rounds of the training.

Table 3.2 describes a summary of key notations used in this paper, and the complete system model is described through the following subsections, addressing the complexities of each component.

3.4.1 The Learning Model

In this model, each near-RT-RIC processes a dataset $D_i = [\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,S_i}]$ of input data where S_i is the number of the input samples processed by the near-RT-RIC i and each element $\mathbf{x}_{is}$ is the FL model's input vector. Let y_{is} be the output of $\mathbf{x}_{is}$. For simplicity, we consider an FL model with single output, which can be readily generalized to a case with multiple outputs. The output data vector for training the FL model of near-RT-RIC i is $y_i = [y_{i,1}, ..., y_{i,S_i}]$. We assume that the data collected by each near-RT-RIC is different from the other near-RT-RICs i.e.

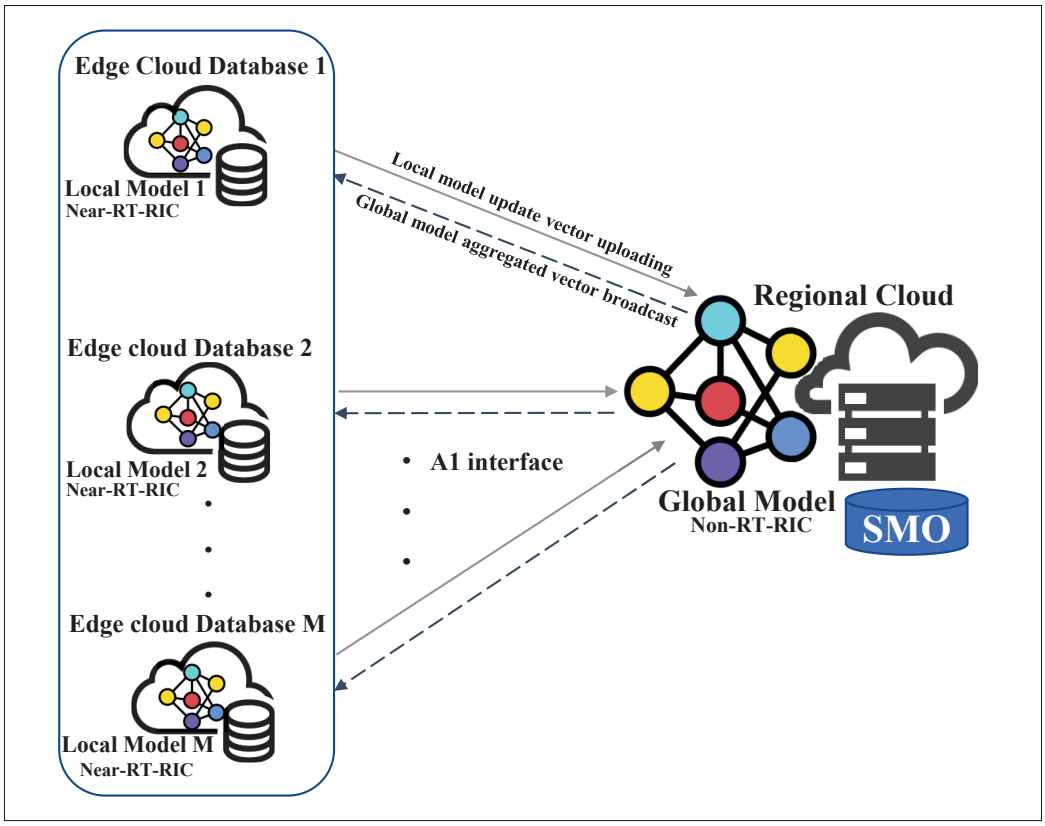


Figure 3.2 System Model for FL update interaction

($\mathbf{x}_i \neq \mathbf{x}_j \;\; ; i \neq j, i, j \in \mathcal{M}$). So, each local trainer will train the model using a different dataset. This is in line with the real scenario as each local near-RT-RIC collects the operational data from the corresponding slice specific users. We define a vector $\mathbf{g}_i$ to capture the parameters related to the local FL model that is trained by D_i and y_i. The vector $\mathbf{g}_i$ actually determines the local FL model of each near-RT-RIC i. For example, in a linear regression prediction algorithm, $\mathbf{x}_i^T . y_i$ represents the output, and $\mathbf{g}_i$ is the associated weight vector that is used to calculate the prediction accuracy. So, in each local model, the target is to get the optimal $\mathbf{g}_i$ so that the the model accuracy can be maximized. Hence, for a system of M local trainers i.e. near-RT-RICs,

the objective of the global FL training process is to solve the following optimization problem:

$$\min_{\mathbf{g}_1,\ldots\ldots,\mathbf{g}_M} \frac{1}{S}\sum_{i=1}^{M}\sum_{s=1}^{S_i} f_i(\mathbf{g}_i, \mathbf{x}_{is}, y_{is}) \tag{3.1}$$

$$s.t. \quad \mathbf{g}_1 = \mathbf{g}_2 = \ldots\ldots = \mathbf{g}_M = \mathbf{g} \quad \forall i \in \mathcal{M} \tag{3.1a}$$

where $S = \sum_{i=1}^{M} S_i$ is the total size of training data of all near-RT-RICs. $\mathbf{g}$ is the global FL model generated by the Non-RT-RIC and $f(\mathbf{g}_i, \mathbf{x}_{is}, y_{is})$ is a loss function that indicates the FL model's training accuracy. The exact expression for the loss function varies depending upon the ML model that is being trained. However, the objective of these functions remains the same. Constraint (3.1a) ensures that, once the FL model converges, all of the near-RT-RICs and the Non-RT-RIC will transmit the parameters $\mathbf{g}$ of the global FL model to its connected near-RT-RICs so that they train their local FL models with the updated weights. Then the near-RT-RICs will transmit their local FL models to the Non-RT-RIC to update the global FL model. The update of each near-RT-RIC i's local FL model $\mathbf{g}_i$ depends on all near-RT-RICs' local FL models.

Generally, the iterative Gradient Descent (GD) method is used to approximate the local model's loss function and its corresponding weights. In this method, the iterative process updates the weights as below:

$$\mathbf{g}_i(t) = \mathbf{g}_i(t-1) - \eta \nabla f_i(\mathbf{g}_i(t-1)), \tag{3.2}$$

where t denotes the iteration count and η is the learning rate.

Once the optimal solution of (3.1) is gained using this updated weight vector ($\mathbf{g}_i$) for all the local trainers i.e. ($i \in 1, 2, ..., M$), the global weight vector ($\mathbf{g}$) can be updated by any FL aggregation policy such as FedAvg (McMahan *et al.*, 2017). This results in the updated global loss function, $f(\mathbf{g})$:

$$f(\mathbf{g}) = \frac{\sum_{i=1}^{M} |S_i| f_i(\mathbf{g}_i)|}{|S|}. \tag{3.3}$$

To assess the performance of model training, the model accuracy is calculated in each global round following the global aggregation. We present the defining notion of accuracy in the next subsection.

Table 3.2 Summary of key notations.

Notation	System Model Parameters
$\mathcal{M}$	Set of distributed edges
D_i	Dataset at i^{th} local trainer
S	Total size of all training data
R^{co}	Total communication resource cost
R_m^{co}	Communication cost of m^{th} near-RT-RIC
R^{cp}	Total compute resource cost
c_m	CPU cycles required per bit of data
f_m	Processing power of m^{th} host
p_c	Per-unit computation cost
R_m^{cp}	Compute cost of m^{th} near-RT-RIC
p_{tr}	Per-unit transmission cost
F	Loss function at the Non-RT-RIC
T_m^{co}	Transmission time of m^{th} local updates
T_m^{cp}	Computation time of m^{th} local model
T_m	Learning time per global FL round at m^{th} model
T^{total}	Total FL training time
B_u	Uplink bandwidth allocation
B_d	Downlink bandwidth allocation
B	Total bandwidth for FL tasks
Notation	**Input Parameters**
g_i	Local model vector at i^{th} near-RT-RIC
θ	Local accuracy
ϵ	Target global accuracy
ρ	Pareto trade-off parameter
$\mathcal{N}$	Set of selected local trainers
K_ϵ	No. of global rounds to attain ϵ accuracy
η	Learning rate
γ	Momentum attenuation factor
S_m^{ω}	Size of compressed update vectors
Notation	**Decision Variables**
a^t	Trainers' selection decision vector
b^t	Bandwidth allocation vector
ω	Compression ratio

3.4.2 FL Model Accuracy

The target for each of the local model trainers is to attain a $\theta \in (0, 1)$ level of accuracy, defined as:

$$||\nabla f_i^t(\mathbf{g}_i^t)|| \leq \theta ||\nabla f_i^t(\mathbf{g}_i^{t-1})||, \forall\ i\ \epsilon\ \{1, 2, 3, .., M\} \tag{3.4}$$

To attain this accuracy, a near-RT-RIC takes several iterations so called local iterations. Correspondingly, in the global model placed at the non-RT-RIC, the target is to attain the optimal model weights to reach ϵ level of global model accuracy, defined as below:

$$|f(\mathbf{g}^t) - f(\mathbf{g})| \ \leq\ \epsilon \quad \forall\ t \geq\ X \tag{3.5}$$

(3.5) simply states that $\mathbf{g}$ is the optimal model parameter i.e. for every global round beyond X, the difference between the loss function values falls within the defined accuracy level no matter how long we keep on iterating.

Now, the convergence of this iterative method is ensured under a set of conditional bounds on the loss function:

$f : \mathbb{R}^n \rightarrow \mathbb{R}$ s. t.

(i) $f(\mathbf{g})$ is convex.

(ii) $f(\mathbf{g})$ is ρ-Lipschitz, i.e. $||f(\mathbf{g}) - f(\mathbf{g}^{'})|| \leq \rho||\mathbf{g} - \mathbf{g}^{'}||$, for any $\mathbf{g}, \mathbf{g}^{'} \in \mathbb{R}^n$.

(iii) $f(\mathbf{g})$ is β-smooth, i.e, $||\nabla f(\mathbf{g}) - \nabla f(\mathbf{g}^{'})|| \ \leq\ \beta||\mathbf{g} - \mathbf{g}^{'}||$, for any $\mathbf{g}, \mathbf{g}^{'}$.

Under the above stated conditions, it is proven by (Konečnỳ *et al.*, 2016) that the number of global iterations required to attain a level of global accuracy ϵ and local accuracy θ can be upper bounded by:

$$K(\epsilon, \theta) = \frac{O(log(1/\epsilon))}{(1 - \theta)} \tag{3.6}$$

We use this relationship among the local accuracy level, global model accuracy, and the upper limit on the number of required global rounds to model resource cost and the FL model training time.

3.4.3 FL Resource Model

In order to transfer model updates from the participating local trainers to the global aggregator and vice versa, the available communication resources are to be assigned. On the other hand, the local trainers require compute resources in the form of processing capacities to locally train the individual models. While the compute resource at the Non-RT-RIC (hosted by a regional cloud) is not overwhelmed, the compute resources at the local node are scarce and provided by the shared edge clouds of ORAN. So, it needs to be judiciously allocated and utilised. In effect, allocation of these resources determine the learning time, communications rounds, and therefore impacts the performance of FL model so trained. Hence, the two aspects of this model training must be jointly considered.

For the communication part, a portion of ORAN's m-Plane fiber link capacity is budgeted for local model uploading with a total bandwidth B. Let $b_m^t \in [0, 1]$ be the bandwidth fraction allocated to trainer m in round t. So, its allocated bandwidth is $b_m^t B$. Let $\mathbf{b^t} = (b_1^t, ..., b_M^t)$ be the bandwidth allocation vector. Bandwidth allocation must satisfy $\sum_{m \in \mathcal{M}} b_m^t = 1 \ \forall t$.

In each global round, the concerned rApp decides which local training points i.e. which near-RT-RICs to participate. This is because at each time interval only a limited number of clients can participate due to delay constraints originating from the control loops of ORAN (Alliance, 2021b). We define a binary variable $a_m^t \in \{1, 0\}$ to decide whether or not the trainer m is selected in round t, and vector $\mathbf{a^t} = (a_1^t, ..., a_M^t)$ collects the overall trainers' selection decisions. A selected near-RT-RIC in round t i.e. $(a_m^t = 1)$, consumes compute resources to train locally with the collected data. Clearly if $a_m^t = 0$, namely trainer m is not selected in round t, then no bandwidth is allocated to it i.e. $b_m^t = 0$. On the other hand, if $a_m^t = 1$, then we require at least a minimum bandwidth b_{min} is to be allocated to the trainer m i.e. $b_m^t \geq b_{min}$. To make the problem feasible, we assume $b_{min} \leq \frac{1}{M}$. Therefore, total resource cost for using communication bandwidth is:

$$R^{co} = \sum_{m=1}^{M} R_m^{co} = K_\epsilon \sum_{m=1}^{M} a_m^t b_m^t B p_{tr} \tag{3.7}$$

for K_ϵ global rounds where p_{tr} is the unit cost of bandwidth usage. For each near-RT-RIC m, let R_m^{cp} denote its local training compute resource cost in every round which depends on its computing host and the dataset. To process the local dataset each near-RT-RIC uses the CPU cycle frequency of the edge host. Let the CPU power of m^{th} host be f_m cycles/s and the per unit time usage cost be p_c. Then the total compute resource cost is:

$$R^{cp} = \sum_{m=1}^{M} R_m^{cp} = \sum_{t=1}^{T} a_m^t \frac{D_m c_m}{f_m} p_c \tag{3.8}$$

where c_m is the CPU cycles required for processing one bit of data.

3.4.4 Latency Model

Since, the number of distributed local edge nodes is expected to be in large numbers, we consider a synchronous mode of communication, in other words the t^{th} round of global aggregation starts only when all the near-RT-RICs have finished sending their local update vectors to the Non-RT-RIC. Therefore, before entering this communication round, all the near-RT-RICs must finish its local ML processing. In each of the global round, the FL tasks are spanned over three operations: (i) computation, (ii) communication of local updates to the Non-RT-RIC using uplink, and (iii) broadcast communication to all the involved near-RT-RICs using downlink. Let the computation time required for one local round for m^{th} near-RT-RIC be T_m^{cp}, and there be K_l local iterations in each interval of the global communication. Then, the computation time in one global iteration round is $K_l T_m^{cp}$. Let the communication time required in transferring the local update vectors from m^{th} near-RT-RIC to the Non-RT-RIC be T_m^{co} in the uplink phase. Let S_m be the datasize of the update vector of m^{th} local trainer. Then, the learning time in one global round for the m^{th} local FL model trainer is:

$$T_m = K_l . T_m^{cp} + T_m^{co} \quad ; m \in \mathcal{M} \tag{3.9}$$

Where T_m^{co} is calculated as:

$$T_m^{co} = \frac{S_m}{b_m^t.B} \quad ; m \in \mathcal{M} \tag{3.10}$$

In the downlink phase, we do not consider the delay because it is negligible as compared to the uplink delay as a result of high speed downlink communication. Since, K_ϵ is the total number of global rounds to attain the global accuracy ϵ as established in (3.6), the total learning time can be modeled as:

$$T^{total} = K_\epsilon.T_{max} = K_\epsilon.\max\{T_m; \ \ \forall m \in \mathcal{M}\} \tag{3.11}$$

Minimizing the resource cost naturally leads to higher learning time and vice-versa. Since the two conflicting goals of FL training must be jointly optimized, for the sake of modeling, we treat learning time as another component of the total cost along with resource usage costs. As such, we use a dimension coherent parameter ρ to frame these two costs into one expression as below:

$$\begin{aligned} cost(t) &= \left\{(R^{co} + R^{cp}) + \rho.(T_m^{co} + K_l.\max\{T_m^{cp}\})\right\} \\ &= \left\{(K_\epsilon.\sum_{t=1}^{T}\sum_{m=1}^{M}.a_m^t(b_m^t.B.p_{tr} + \frac{D_m.c_m}{f_m}.p_c))+ \right. \\ &\left. \rho.(\frac{S_m}{b_m^t.B} + K_l.\max\{T_m^{cp}\})\right\} \end{aligned} \tag{3.12}$$

3.4.5 Problem Formulation

Our goal is to jointly minimize the resource cost and the learning time under the constraints of bandwidth resources. This can be done by optimizing the number of local trainers i.e. near-RT-RICs and bandwidth allocation as formulated in the optimization model (3.13).

$$\mathbf{P:} \min_{\mathbf{a^t}, \mathbf{b^t}} \quad cost(t) \tag{3.13}$$

subject to:

$$\sum_{m=1}^{M} a_m^t . b_m^t . B \leq B, \tag{3.13a}$$

$$\sum_{m=1}^{M} b_m^t = 1, \tag{3.13b}$$

$$b_{min} \leq b_m^t \leq 1 \quad ; \forall m \in \mathcal{M}, \tag{3.13c}$$

$$a_m^t \in \{1, 0\}. \tag{3.13d}$$

$$f_m \geq f_{min} \tag{3.13e}$$

$$\sum_{v \in \{DU\}} \delta_{nv} \leq 1 \quad \forall n \in \{RU\} \tag{3.13f}$$

$$\sum_{u \in \mathcal{M}} \zeta_{mu} \leq 1 \quad \forall m \in \{CU\} \tag{3.13g}$$

The objective function (3.13) has two components balanced by a trade-off parameter ρ because the two goals are conflicting. The total resource cost is represented as $R^{total} = R^{cp} + R^{co}$, and the FL training time T^{total} is given by (3.11). Constraint (3.13a) bounds the total bandwidth allocated for the FL tasks. Constraint (3.13b) presents the definition of b_m^t; that is, the sum of bandwidth fractions must be 1. In (3.13c), we denote the boundary of the bandwidth fractional allocation. (3.13d) represents the defining domain of the decision variable. Additionally, (3.13e) ensures that the assigned computation resource is greater than the minimum CPU frequency, while (3.13f) and (3.13g) state that the selection of near-RT-RIC instances must adhere to the hierarchical architecture of ORAN.

3.5 MCORANFed

In our prior work, we proposed ORANFed (Singh & Nguyen, 2022b) that uses a ORAN system based deadline aware and slice specific local trainers' selection and then trains the FL model.

This method, although provides a novel FL algorithm to suit the requirements of ORAN, could not be extended for a large number of local trainers as it does not mitigate the communication latency. Therefore, we address this communication bottleneck by using compression on the update vectors and further decreasing the total number of global rounds by expediting the convergence through momentum gradient descent. Figure 3.3 illustrates the proposed steps of MCORANFed.

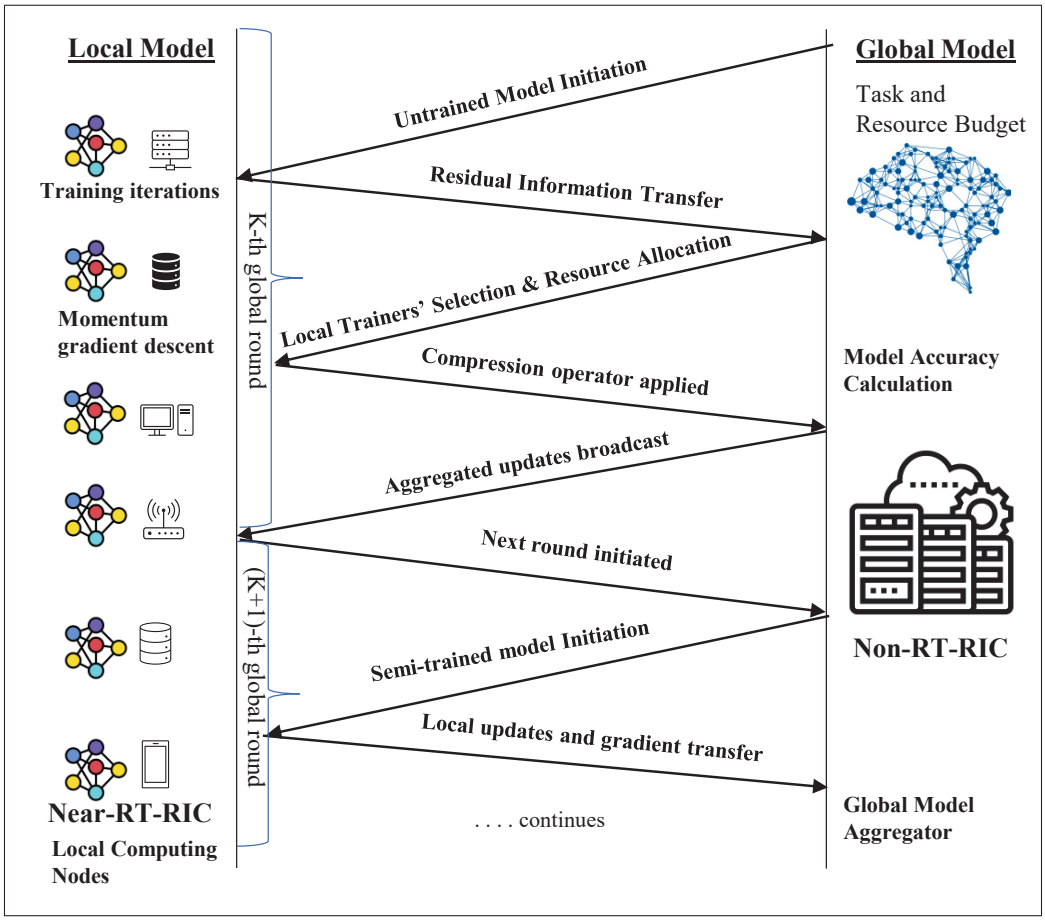


Figure 3.3 MCORANFed model with proposed steps

3.5.1 Randomized Compression Operator

It has been shown in (Li *et al.*, 2020c) that a randomized compression operator achieves a convergence rate of $\mathcal{O}((1+\omega)\frac{L}{\epsilon})$ as opposed to (3.6) in no-compression scenario. Due to the faster convergence, it saves on communication time too.

An ω - compression operator can be defined as a map $\boldsymbol{C : \mathbb{R}^d \rightarrow \mathbb{R}^d}$
s.t. it satisfies the following conditional properties:
(i)

$$] \mathbb{E}[C(x)] = x, \quad \forall x \in \mathbb{R}^d \tag{3.14}$$

i.e. C(.) is unbiased and

(ii) $\exists\ \omega \geq 0$ s.t.

$$\mathbb{E}[||C(x) - x||^2] \leq \omega ||x||^2, \forall x \in \mathbb{R}^d \tag{3.15}$$

i.e. its variance is uniformly bounded.

For the case of no compression $(\omega = 0), C(x) \equiv x$.
From this class of compression operator, a random sparsification operator can be defined as:

$$C(x) := \frac{d}{k}(\zeta_k \cdot x) \quad \forall x \in \mathbb{R}^d$$

where $\zeta_k \in \{0, 1\}^d$ is a uniformly random binary vector with k non-zero elements. From the above definition of compression operator, this specific operator can be derived by substituting $\omega = \frac{d}{k} - 1$ and $k = d$ implies zero compression.

3.5.2 Accelerated Gradient Descent

As can be inferred from (3.2), GD is a first order approximation method. To increase the convergence rate of this iterative approximation, we impose second order term using Momentum Gradient Descent (MGD) (Liu *et al.*, 2020). MGD improves GD by adding a momentum term which leads to the following update rule:

$$d_i(t) = \gamma d_i(t-1) + \nabla f_i(\mathbf{g}_i(t-1)) \tag{3.16}$$

and

$$\mathbf{g}_i(t) = \mathbf{g}_i(t-1) - \eta d_i(t) \tag{3.17}$$

where $d_i(t)$ is the momentum term having the same dimension as $\mathbf{g}_i(t)$, η is the learning step size, γ is the momentum attenuation factor, and t is the iteration index. $f(\mathbf{g})$ with iterations of (3.16) and (3.17) can converge to the minimum loss function value faster than the GD. MGD converges in the range of $-1 < \gamma < 1$ with a bounded η. It further shows accelerated convergence within the range of $0 < \gamma < 1$ with small values of η (Liu *et al.*, 2020).

Local iterations start with initial values for $d_i(0)$, and $\mathbf{g}_i(0)$. Then local updates are performed using (3.16) and (3.17) for each $t \in [k]$. $[k]$ is the periodic global aggregation interval. Without loss of generality, we assume that $t = kT$, where T is the number of global rounds, t is the total number of iterations, and k is the period of global model aggregation. Whenever t is a multiple of T, global aggregation is performed using the following rule:

$$d(t) = \frac{\sum_{i=1}^{M} |S_i| d_i(t)}{|S|} \tag{3.18}$$

and

$$g(t) = \frac{\sum_{i=1}^{M} |S_i| \mathbf{g}_i(t)}{|S|} \tag{3.19}$$

In order to ensure the accelerated convergence rate of MGD based FL over GD based FL training, we require an additional set of conditions on the loss functions f_i. Together with conditions **(i)** to **(iii)**, two additional conditions have to fulfilled by the loss functions.

(iv) For any $\mathbf{g}$ and i, the difference between the global gradient and local gradient can be bounded by
$||\nabla f_i(\mathbf{g}) - \nabla f(\mathbf{g})|| \ \leq \ \delta_i$, and $\delta := \frac{\sum_i S_i . \delta_i}{S}$.

(v) The loss function f is μ-strongly convex: i.e.
$f : \mathbb{R}^d \rightarrow \mathbb{R}$ and
$\exists\ L \geq \mu \geq 0$ s.t. $\mu||\mathbf{g} - \mathbf{g}'|| \leq ||\nabla f(\mathbf{g}) - \nabla f(\mathbf{g}')||$.

These conditions form a class of well behaved functions which aligns with most of the ML methods' loss functions and are in line with the recent works(Yang *et al.*, 2021), (Wang *et al.*, 2019a), (Yang *et al.*, 2020), (Chen *et al.*, 2021a) on convergence analysis of FL. This leads to the aggregated model loss function, $f(\mathbf{g})$, defined over all the local loss functions as:

$$f(\mathbf{g}) := \sum_{i=1}^{M} (|S_i|/S) f_i(\mathbf{g}_i) \tag{3.20}$$

Then the iterative model compression, when applied on the model updates, works as;

$$\mathbf{g}(t) = C(\mathbf{g}(t-1)) - \gamma \nabla f(C(\mathbf{g}(t-1))) \tag{3.21}$$

This is Momentum Gradient Descent with Compressed Iterates (MGDCI) with compression parameter ω from the class of randomized compression operators(Li *et al.*, 2020c). Such an operator is proven to converge linearly to an approximate solution of the size $O(\kappa\omega)$ in the neighbourhood of the optimal solution of (3.1) provided it is bounded by the following threshold (Khaled & Richtárik, 2019):

$$\omega \leq \frac{\mu}{4} . \frac{1 - 2\gamma L}{2\gamma L^2 + \frac{2}{\gamma} + L - \mu} \tag{3.22}$$

3.5.3 Updated Optimization Problem

Let S_m^{ω} be the datasize of the update vector of m^{th} trainer under the ω-compression operator. Then, S_m^{ω} is calculated as:

$$S_m^{\omega} = \frac{k}{1 - \omega}, \tag{3.23}$$

where k is the sparsification factor.

Therefore, the learning time in one global round of FL training for the m^{th} local FL model trainer becomes:

$$T_m^{co} = \frac{S_m^{\omega}}{b_m^t . B} \quad ; m \in \mathcal{M}, \tag{3.24}$$

the cost function updates as:

$$cost^*(t) = \Big\{(K_\epsilon . \sum_{t=1}^{T}\sum_{m=1}^{M} .a_m^t(b_m^t.B.p_{tr} + \frac{D_m.c_m}{f_m}.p_c))+ \\ \rho.(\frac{S_m^\omega}{b_m^t.B} + K_l.\max\{T_m^{cp}\})\Big\} \tag{3.25}$$

Hence, the updated joint resource allocation and FL model learning time optimization problem is:

$$\mathbf{P:} \min_{\mathbf{a^t},\mathbf{b^t},\omega} \quad cost^*(t) \tag{3.26}$$

subject to:

$$\sum_{m=1}^{M} a_m^t.b_m^t.B \leq B, \tag{3.26a}$$

$$\sum_{m=1}^{M} b_m^t = 1, \tag{3.26b}$$

$$b_{min} \leq b_m^t \leq 1 \quad ;\forall m \epsilon \mathcal{M}, \tag{3.26c}$$

$$a_m^t \in \{1, 0\}. \tag{3.26d}$$

$$f_m \geq f_{min} \tag{3.26e}$$

$$\frac{4\omega}{\mu} \leq \frac{1 - 2\gamma L}{2\gamma L^2 + \frac{2}{\gamma} + L - \mu} \tag{3.26f}$$

Constraint (3.26f) bounds the feasible domain of ω to ensure the convergence of MCORANFed, where μ, γ, and L are the parameters as defined in the set of conditions **(i)** to **(v)** on the loss function and as described in (3.22).

3.6 Proposed Solution

The problem formulated in (3.26) is a non-convex optimization problem because the objective function and the constraint (3.26a) are non-convex functions. Moreover, it is hard to transform this problem to get at least a close form solution. Therefore, we take an approach of decomposition method. As shown in Figure 3.4, we layout a scheme of the solution approach. We first divide the main optimization problem into two sub-problems, named respectively Local Trainers' selection (P1) and Resource Allocation (P2). Then we use a solution of the first sub-problem to reshape the second sub-problem. Finally, we solve the updated second sub-problem.

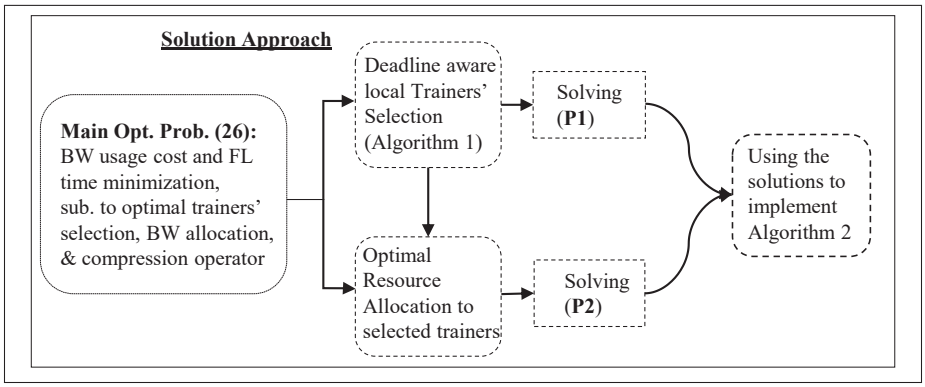


Figure 3.4 Schematic Diagram of the Proposed Solution

3.6.1 Local Trainers' Selection

Among the three decision variables, $\mathbf{a^t}$ has been derived through a deadline aware and slicing based local trainers' selection algorithm as proposed in (Singh & Nguyen, 2022b). In this step, we solve the following sub-problem:

$$\mathbf{P1:} \min_{\mathbf{a^t}} (K_\epsilon . \sum_{t=1}^{T} \sum_{m=1}^{M} .a_m^t (b_m^t . B . p_{tr} + \frac{D_m . c_m}{f_m} . p_c)) \tag{3.27}$$

subject to:

$$a_m^t \in \{1, 0\}. \tag{3.27a}$$

The objective of this trainers' selection algorithm is to maximize the number of near-RT-RICs to participate in each global round, and to allow the non-RT-RIC to aggregate all received local model updates. This is based on the proposition that a larger fraction of trainers in each round saves the total time required for a global FL model to attain the desired model accuracy (McMahan *et al.*, 2017). As specified in O-RAN Alliance whitepaper (O-RAN Alliance, 2024a), the collected RAN operational data can be separated based on their slice-user groups. Each near-RT-RIC is then fed with slice specific network data. The selection of a near-RT-RIC corresponding to a slice must be incorporated in each iteration of gradient descent training of the model. However, not all the local models can be accommodated in each iteration because of the deadline constraint and limited computational and bandwidth resources to be assigned for this learning task. Moreover, due to the variation in traffic patterns for different kinds of slicing services of ORAN, the local FL model might encounter inconsistency problem. This may lead to a degradation in accurate prediction. We take into account this differentiation and propose a trainers' selection algorithm that respects the formation of slices in ORAN while maintaining a deadline awareness. In Algorithm 3.1 (Singh & Nguyen, 2022b), we categorize the set of near-RT-RICs into three classes corresponding to eMBB, uRLLC, and mMTC slicing services.

Let $\mathcal{N}(\subseteq \mathcal{M})$ be the set of selected near-RT-RICs, t_{round} be the deadline for each global round, t_1 be the elapsed time to perform Algorithm 3.1, and t_{agg} be the time taken in aggregating the update parameters at the Non-RT-RIC. Therefore, the mathematical optimization problem for the trainer selection becomes:

$$\max_{\mathcal{N}} \{|\mathcal{N}|\} \tag{3.28}$$

$$s.t. \quad t_1 + t_{agg.} + \frac{1}{2}(t_n^{k-1} + \alpha.t_n^k) \leq t_{round}. \tag{3.28a}$$

(3.28) is a combinatorial optimization problem which makes it non-trivial. So, we employ a greedy heuristic to solve this problem as shown in Algorithm 3.1. We repeat the steps in each

global round until we get the desired accuracy. Here, the constraint (3.28a) restricts the violation of the deadline for every near-RT-RIC in each global round. The deadline is assigned separately for each slice-user groups while the total deadline in each round is varied experimentally to observe its impact on overall learning time of the global FL model.

Algorithm 3.1 Deadline aware and Slicing based Local Trainers' Selection

Input: $\mathcal{M}$: Set of all near-RT-RICs
Output: $\mathcal{N} = \mathcal{N}^u \cup \mathcal{N}^e \cup \mathcal{N}^m$

Initialize $\mathcal{N}^u, \mathcal{N}^e, \mathcal{N}^m = \emptyset$;
for t^i_{round} *defined for* $i \in \{\mathcal{N}^u, \mathcal{N}^e, \mathcal{N}^m\}$ **do**
  while $|\mathcal{N}| > 0$ **do**
    $x \leftarrow \arg\min_{n \in \mathcal{N}} \frac{1}{2}(t_n^{k-1} + \alpha t_n^k$ (estimated));
    $t \leftarrow t_i + t_{agg} + t_n^k$;
    $\mathcal{N} \leftarrow \mathcal{N} \setminus \{x\}$;
    if $t < t^i_{round}$ **then**
      $t \leftarrow t + t_n^k$;
    end if
  end while
end for

3.6.2 Resource Allocation

From the trainers' selection phase, we obtain $\mathbf{a^t}$, i.e. a binary valued vector of selected trainers in k^{th} global round. The next phase is to allocate the compute and bandwidth resources to support the local training, parameters uploading, model aggregation, and broadcast of updated model weights. For this task, we solve the optimization problem (3.26) with the obtained value of $\mathbf{a^t}$ and unknown variable $\mathbf{b^t}$ and ω.

Still, (3.26) is a non-convex optimization problem, therefore, an exact solution of it is intractable using traditional methods. Therefore, we employ an approximation approach with equivalent surrogate functions. With these changes and substituting the defining expressions, the optimization problem (3.26) reduces the following mathematical form:

$$\textbf{P2:} \min_{\mathbf{b^t},\omega} \quad cost^*(t) \tag{3.29}$$

subject to: (26a), (26b), (26c), and (26f)

The number of local iterations (K_l) in each global round is determined experimentally as required in attaining the local accuracy value θ. We solve this problem using an iterative approximation method and implemented the solution using Ipopt solver (Boyd & Vandenberghe, 2004). Here, the constraints are linear with respect to each of the vector variable $\mathbf{b^t}$ and ω.

3.6.3 Federated Training in ORAN RICs (MCORANFed)

Using the solutions of trainer selection and resource allocation in (3.29), we train the FL model as described in Algorithm 3.2. In each global round, a subset of participating local trainers is selected followed by resource allocation in each global round. This loop continues for K_ϵ iterations, which is the maximum number of global rounds required to attain the prefixed accuracy of the model.

3.6.4 Theoretical Analysis

Theorem 1: *Suppose a constant learning step size* $\eta^k = \frac{\theta\sqrt{N}}{\sqrt{H}}, \forall k$ *is chosen where* $\theta > 0$ *is a constant satisfying* $\frac{\theta\sqrt{N}}{\sqrt{H}} \leq \frac{1}{2L}$*, the convergence rate of Algorithm 3.2* is:

$\mathbb{E}[||z_T||^2] \leq$

$4(\mathbb{E}[F(g^0)] - F^*) + \frac{8\zeta\theta L\delta^2}{(\zeta-1)\sqrt{N}H^{3/2}} + (4\omega^2+1)\frac{8N\theta^2L^2\mu^2H}{K_\epsilon^2}$,

Algorithm 3.2 Momentum Compressed ORANFed

Input: The dataset D_i ($\forall i \in \mathcal{M}$); The number of participants: M; The number of iterations: K_ϵ
Output: Final model parameter $\mathbf{g}$

for $k = 1, 2, 3, \ldots, K_\epsilon$ **do**
  Non-RT-RIC uses Alg. 3.1 to get subset $\mathcal{N}$;
  Allocation of compute and bandwidth resources to selected near-RT-RICs ($\mathcal{N}$);
  Each near-RT-RIC trains using local data till it achieves an accuracy θ and obtains $\mathbf{g}_{i,k}$;
  for *all local trainers in parallel* $m \in \mathcal{N}$ **do**
    Compress local gradients using (3.21) and obtain $\mathbf{g}_i$ for $i \in \mathcal{N}$;
    Transmit $\mathbf{g}_i$ to Non-RT-RIC;
  end for
  Decompress the received compressed weights and aggregate $\mathbf{d}(t)$ and $\mathbf{g}(t)$ according to (3.18) and (3.21);
  Calculate global loss function $f(\mathbf{g})$ through (3.20);
  Non-RT-RIC broadcasts the aggregated parameters;
  Non-RT-RIC calculates the global accuracy attained (3.5);
end for
Final trained model is sent to SMO for deployment;

where H is the total number of overall iterations (local and global), z_T is a random variable which samples a weight parameter g^t with probability $\frac{1}{NH}$, and ζ is a constant.

Proof: Please refer to Appendix I

In Theorem 1, we follow a weaker notion of convergence where the mean expected squared gradient norm is taken to express the convergence rate because of the non-convex settings (Li *et al.*, 2021).

Theorem 2: *Suppose $f(x)$ is convex with L-Lipschitz continuous gradient and the compression operator $C(.)$ satisfies (3.14) and (3.15). Let the learning step size $\eta = \frac{1}{(1+\omega)L}$, then the number of iterations performed by MCORANFed to find an ϵ-solution such that $\mathbb{E}[||z_T||^2] \leq \epsilon$ is at most*

$K_\epsilon = \mathbb{O}(\frac{(1+\omega)L}{\epsilon})$.

Where ϵ is the right hand side of the inequality given in Theorem 1.

Proof: Please refer to Appendix I

Momentum Compressed ORANFed consists of trainers' selection in Step 3, and assignment of resources in Step 4. Steps 5 to 13 train the FL model iteratively. So, the complexity of Algorithm 3.2 can be analysed in two parts. In the first part, (3.27) is solved using Algorithm 3.1 having time complexity of $O(L)$ where L is the cardinality of the set $\mathcal{M}$. In the second part, (3.29) is solved using Interior Point Approximation method having complexity $O(J_{Ipopt})$ (Boyd & Vandenberghe, 2004), where J_{Ipopt} is the total number of iterations within the Ipopt solver algorithm.

3.7 Numerical Results

To provide a seamless network performance, FL training as well as both the subproblems of the local trainers' selection and resources' allocation are performed offline in the ORAN SMO layer. An rApp installed in the SMO monitors the model accuracy by running it in an offline mode before deploying it onto the lower network elements (e.g., O-CU, O-DU). To this end, both Algorithms 3.1 and 3.2 are run by the rApp on the Non-RT-RIC platform. In the following subsections, we describe the FL training task, experimental settings, baselines used to compare the results, and analyse those results.

3.7.1 Federated Learning Task:

We train a supervised data traffic prediction model using a time series dataset (Liu, 2021). This labelled data is accumulated over a year, and the goal is to predict incoming traffic for the next

Table 3.3 Experimental Settings

Parameter	Description	Value
N	Max. no. of local trainers	50
B	Bandwidth budget for FL training	1 MHz
c_m	Processing rate	15 cycles/bit
f_m	Max. CPU power	$\sim U(1,\ 1.6)$ GHz
p_{tr}	Per-unit transmission cost	1
p_c	Per-unit computation cost	1
D_m	Dataset size	$\sim U(5,\ 10)$ MB
d	Update vector size	20 bits
b_{min}	Minimum bandwidth allocation	0.1 MHz
k	Sparsification factor	0.35
η	Learning rate	$\sim (0.1,\ 0.4)$
ρ	Pareto trade-off parameter	$(0,\ 1)$

hour. This kind of task is generally required to address a radio resource assignment problem. We assign the labels for 3 different types of network slices (e.g., URLLC, eMBB, and mMTC), and uniformly distribute the entire dataset onto 50 local nodes. To obtain good learning performance of the time-series datasets used in our experiments, we train this forecasting model using Long Short Term Memory (LSTM) based recurrent neural network (RNN) consisting of 4 layers. LSTM has largely been adopted as an efficient ML method for time-series data thanks to its ability to produce better accuracy by training with non-uniform sequences of daily data traffic (Wang, Zhu & Li, 2019b).

In Fig. 3.5 and Fig. 3.6, a comparison of centrally trained ML based forecasting models is drawn. These RNN models include: `Basic_RNN` (4 input and one dense output layer, activation function = $tanh$, optimizer = $Adam$), `GRU` (four stacked GRU layer and single output layer, activation function = $tanh$, optimizer =$stochasticgradientdescent(SGD)$, learning rate = 0.01), and `LSTM` (three input layer and one dense output layer, optimizer = $Adam$, activation function = $sigmoid$) and MSE is the loss function and accuracy is the evaluation metric. The loss function for these models is mean square error (MSE) metric, used for prediction accuracy as explained with equation (3.1) in Section 3.4. Before going into the federated settings, the model

is trained using centralised learning to get the benchmark model accuracy which is around 96.3%. Therefore, the value of ϵ (global accuracy for FL) is set as 0.96.

3.7.2 Network Settings:

We run this training on a machine with Intel(R) Core(TM) i5-8265U CPU having a maximum processing power of 1.60 GHz. For simplicity, all near-RT-RIC nodes have the same data processing rate c_m. A uniform distribution random number generator is used for assigning CPU frequency (f_m) of the host in the range of (1, 1.6). The value of compression parameter (ω) lies in (0, 1) and the sparsification factor (κ) is fixed as 0.35 as obtained from a experimental data for lossless compression. Small values of learning rate (η) in (0.1, 0.4) are taken and the average convergence rate is reflected over 10 simulations. The main settings are summarized in Table 3.3.

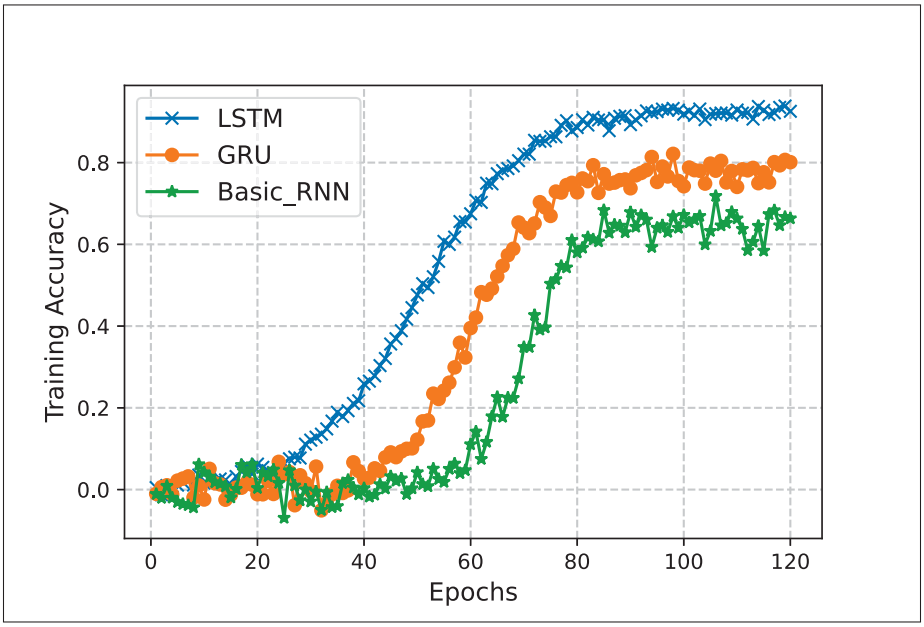


Figure 3.5 Convergence of Centralized ML Models

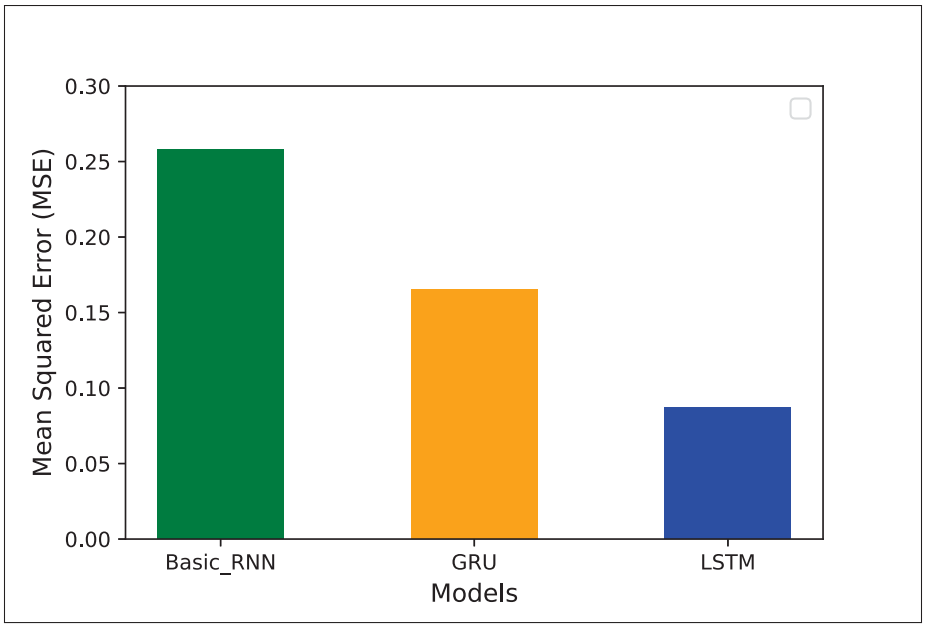


Figure 3.6 Performance of Centralized ML Models

3.7.3 Baselines:

Since every FL variant targets a specific objective corresponding to the mobile edge environment it is trained in, we consider the following FL models as baselines for comparison:

- **Federated Averaging (FedAvg) (McMahan *et al.*, 2017):** FedAvg algorithm with fixed number of clients ($N = 50$) is considered for the benchmark. This serves as the basic FL model without compression and without any selection of local trainers in global rounds. This provides a limiting comparison with other FL variants and our proposed method.
- **Momentum Federated Learning (MFL) (Liu *et al.*, 2020):** MFL is used to compare the effect of momentum attenuation factor on the accelerated convergence model. This method is an improvement of the traditional gradient descent approach. Serving as the model without compression operator, this FL variant does not use any local trainers' selection algorithm. We performed several variants of MFL by setting the momentum attenuation factor (γ) between 0 to 1. Out of these, two variants of MFL are considered: one with $\gamma = 0.7$ and another with $\gamma = 0.9$. These variations are used to show the differential behaviours of momentum attenuation factor when applied on the iterative convergence of FL.

- **ORAN Federated Learning (ORANFed) (Singh & Nguyen, 2022b):** In our prior work, we have proposed ORANFed algorithm that uses a local trainers' selection to assign the resources for FL training. This baseline serves as an FL variant with selection but without compression or acceleration operator.

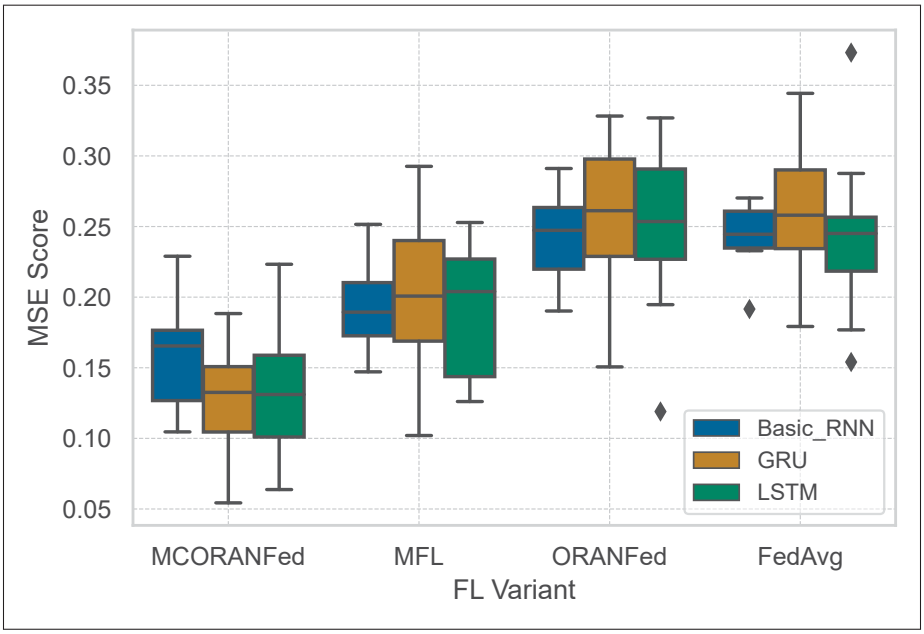


Figure 3.7 Distribution of the MSE score on the local trainers

3.7.4 Performance Evaluation

We train the FL models with three different ML based forecasting methods trained in 50 local trainers, and observe the mean square error (MSE) values attained by the local trainers. Using a box and whiskers plot, Figure 3.7 compares the distribution of final model's MSE scores around the mean and quartiles taken over all the locally trained models. As shown, MCORANFed exhibits its faster convergence property, achieving the best model fit in particular for LSTM. The other baselines experience higher MSE scores and show a mixed performance with respect to the RNN models.

In general, the performance of a FL model can be assessed on the basis of how its accuracy improves according to the global rounds. Figure I-1 in Appendix II clearly shows the advantages

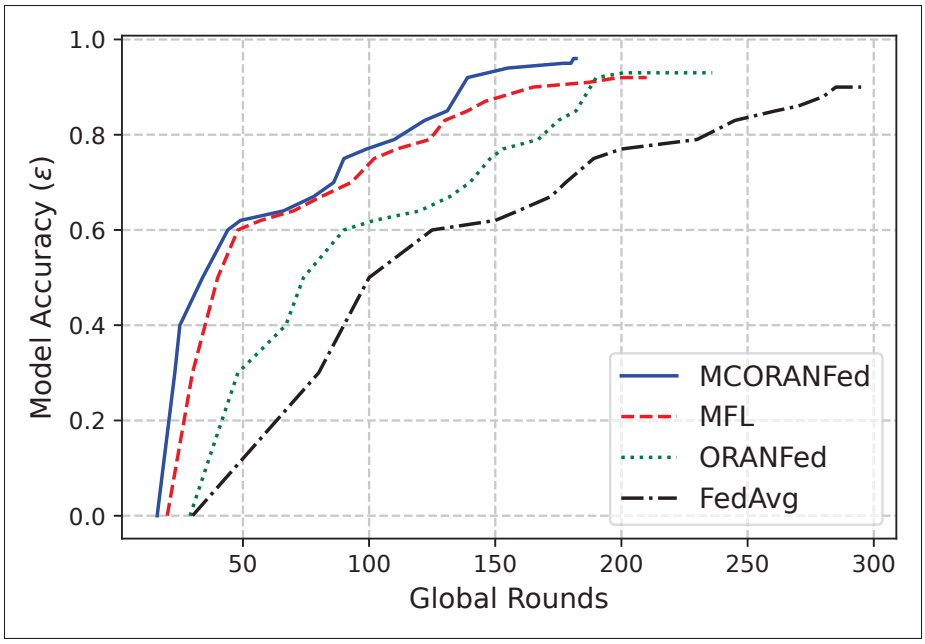


Figure 3.8 Impact of Accelerated Convergence

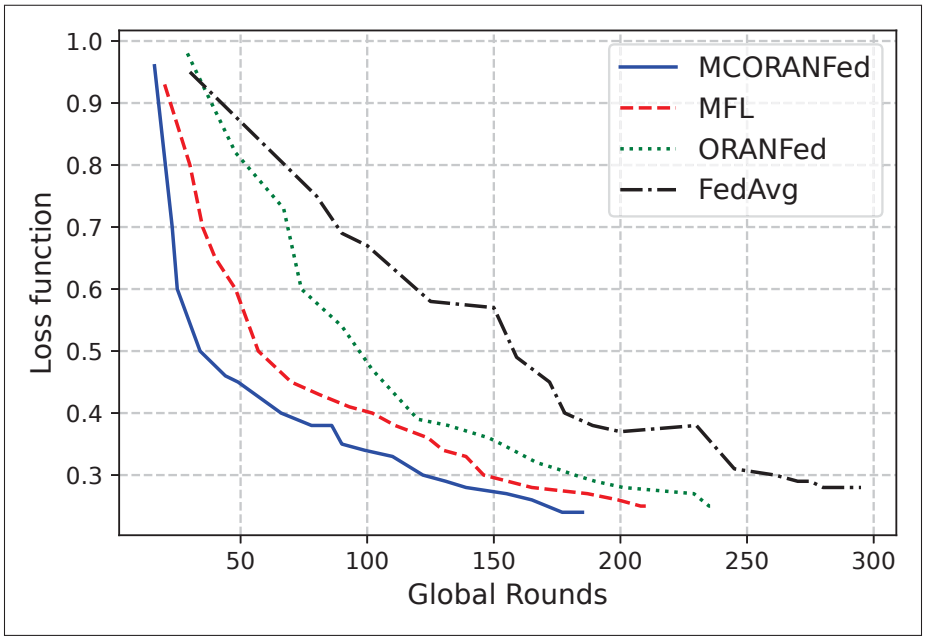


Figure 3.9 Accelerated Convergence w.r.t Loss function

of our proposed model in terms of accuracy improvement of the local model trained at near-RT-RICs (denoted as θ in the system modelling). This FL setting includes 50 local trainers (e.g.,

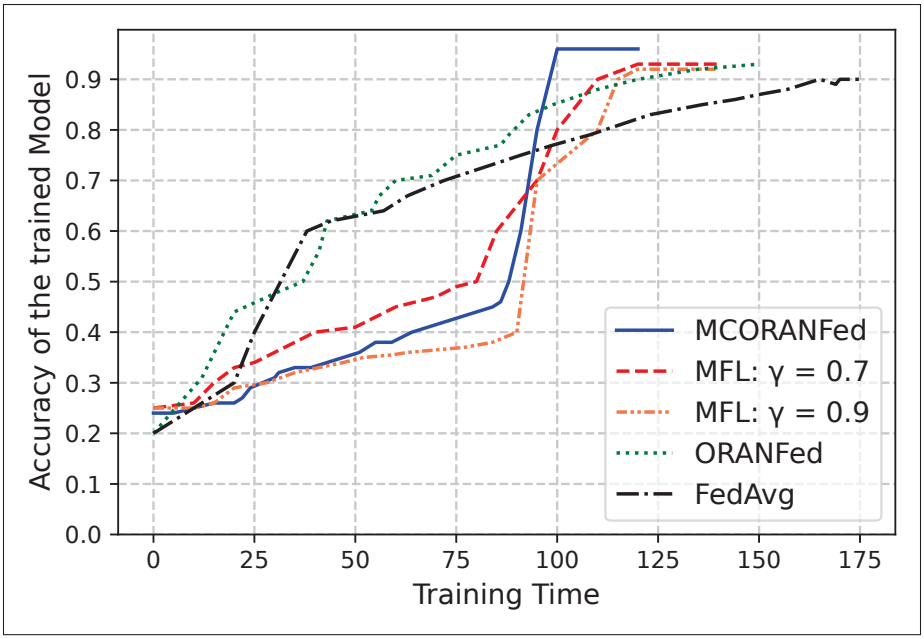


Figure 3.10 Time elapsed to converge

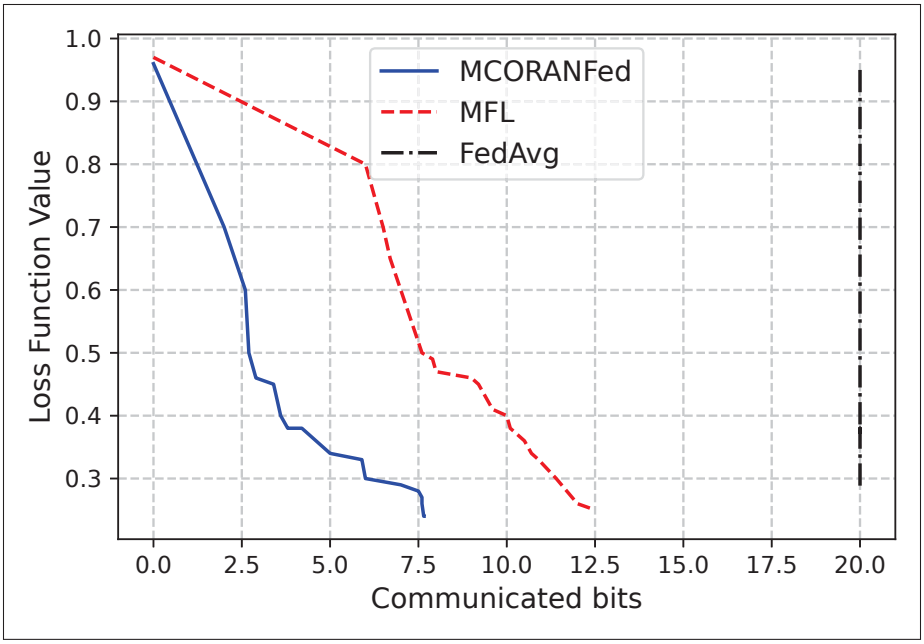


Figure 3.11 Impact of Compression Operator

near-RT-RIC1 to near-RT-RIC50) that train their models from 0 to 300 global rounds, using different FL methods. Figure I-1 captures the accuracy of each local trainer at the 50th round,

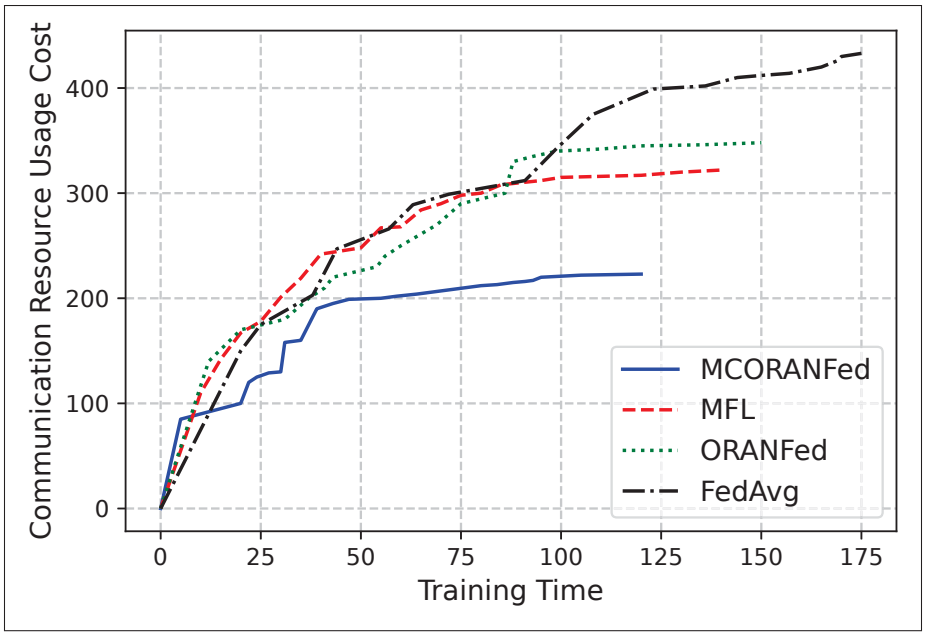


Figure 3.12 Objective Cost Comparison

100th round, 150th round, 200th round, 250th round, and 300th round, respectively. After 300 rounds, MCORANFed constantly outperforms the other baselines in terms of the accuracy of all local trainers. We present the performance evaluation of our proposed model under the following metrics:

3.7.4.1 Convergence Rate

Figure 3.8 shows the joint impact of acceleration and compression in converging the model. While FedAvg requires a significantly high number of global communication rounds to converge as compared to other three methods, it also performs poorly with respect to the model accuracy so attained. Whereas, ORANFed, due to its deadline aware local trainers' selection, requires a smaller number of global rounds and also attains higher model accuracy. MCORANFed not only attains the highest model accuracy but in the same time it also requires a significantly low number of global communication rounds to converge. Although MFL performs better than FedAvg and ORANFed, it converges slower than our proposed method. MCORANFed takes advantages of both momentum gradient descent and trainers' selection algorithm. A similar

impact can be seen in Figure 3.9 where loss function value is plotted against the number of global communication rounds for each FL variants.

3.7.4.2 Training Time

FL methods can also be evaluated according to the time it takes to train the final model. The total time includes data processing time at the local trainer, model update transmit delay, and global aggregation time in each global round. Figure 3.10 shows that MCORANFed takes about 120 time units whereas both variants of MFL takes about 135 units of time. ORANFed and FedAvg require longer learning time: 150 and 175 units respectively. A smaller number of communication rounds results in a lower transmit delay. Therefore, MCORANFed saves the total training time too.

3.7.4.3 Impact of Compression

In Figure 3.11, the loss function value is plotted against the size of the compressed bits that are communicated in each global round. FedAvg which uses no compression operator, transfers 20 bits in every round. Whereas MFL and MCORANFed share variable compressed bit sizes. This Figure shows that MCORANFed reaches a lower loss function value within a smaller number of global rounds and sends lower bit sizes as compared to MFL. This result demonstrates the efficiency of the compression operator that we have applied on this model.

3.7.4.4 Resource Usage Costs

The objective of the main optimization problem defined in this paper is to minimize in the same time the resource cost and the learning time. Figure 3.12 shows the comparison of our proposed method with respect to bandwidth usage cost in training the FL model. As shown, MCORANFed performs better than MFL, ORANFed, and FedAvg in terms of both aspects of the objective function.

Through these results, the superior performance of our proposed method is confirmed. The differentiating factor that improves the FL method can also be clearly seen. In practical implementations, MCORANFed can be run offline in the SMO layer to optimally train FL models which can be executed by an rApp of the Non-RT-RIC for policy decisions on slice resource management. For example, such a model can be used for the inference in control-loops of ORAN or for the QoS optimization through dynamic radio resource management of ORAN in which predictive models are executed frequently. In such cases, a trained FL model will allocate radio resources for each type of slice user groups from the overall radio resources of a Mobile Network Operator (MNO). Moreover, MCORANFed is scalable to accommodate even more than three types of slices as our system model can be readily generalized for N slice types. The processing capacity of VMs can also be adjusted by the MNO. This will allow the system to process operational data of bigger sizes. However, the training time will increase leading to less frequent policy change decisions.

3.8 Conclusion and Future work

In this paper, we proposed a communication efficient federated learning method designed for ORAN systems. Our model takes into account the importance of faster convergence through momentum gradient descent and compressed communication, deadline aware local trainers' selection, and an optimal resource allocation for training FL models. The proposed model outperforms state-of-the-art FL methods in terms of learning time and resource cost in the experimental settings. The simulation results show that an FL model trained with MCORANFed can save resource costs which is substantial for smart radio resource allocation in ORAN. Therefore, it can be deployed in the control loops of ORAN for different use cases such as to guarantee the slice QoS. In future, we will investigate the location of the distributed data collection points to improve further MCORANFed in a highly complex yet realistic environment.

Acknowledgment

The authors thank Mitacs, Ciena, and ENCQOR for funding this research under the IT13947 grant.

CHAPTER 4

USER HANDOVER AWARE HIERARCHICAL FEDERATED LEARNING FOR OPEN RAN BASED NEXT-GENERATION MOBILE NETWORKS

Amardip Kumar Singh[1] , Kim Khoa Nguyen[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper published in *IEEE Transactions on Machine Learning in Communications and Networking*, July 2025

Abstract The Open Radio Access Network (O-RAN) architecture, enhanced by its AI-enabled Radio Intelligent Controllers (RIC), offers a more flexible and intelligent solution to optimize next generation networks compared to traditional mobile network architectures. By leveraging its distributed structure, which aligns seamlessly with O-RAN's disaggregated design, Federated Learning (FL), particularly Hierarchical FL, facilitates decentralized AI model training, improving network performance, reducing resource costs, and safeguarding user privacy. However, the dynamic nature of mobile networks, particularly the frequent handovers of User Equipment (UE) between base stations, poses significant challenges for FL model training. These challenges include managing continuously changing device sets and mitigating the impact of handover delays on global model convergence. To address these challenges, we propose MHORANFed, a novel optimization algorithm tailored to minimize learning time and resource usage costs while preserving model performance within a mobility-aware hierarchical FL framework for O-RAN. Firstly, MHORANFed simplifies the upper layer of the HFL training at edge aggregate servers, which reduces the model complexity and thereby improves the learning time and the resource usage cost. Secondly, it uses jointly optimized bandwidth resource allocation and handed over local trainers' participation to mitigate the UE handover delay in each global round. Through a rigorous convergence analysis and extensive simulation results, this work demonstrates its superiority over existing state-of-the-art methods. Furthermore, our findings underscore significant improvements in FL training efficiency, paving the way for advanced applications such as autonomous driving and augmented reality in 5G and next-generation O-RAN networks.

Keywords: hierarchical federated learning, FL, open RAN, O-RAN intelligent controllers, handover, mobility, B5G

4.1 Introduction

The next generation of wireless communication, 5G and beyond, is expected to support cutting-edge, low-latency applications such as autonomous vehicles, immersive augmented reality (AR), and remote medical surgeries (Qureshi *et al.*, 2021),(Marinova & Leon-Garcia, 2024). The effective delivery of these services heavily depends on the ability of radio access networks (RANs) to ensure consistent Quality of Service (QoS) while efficiently managing radio spectrum and energy resources (Polese *et al.*, 2023a). To meet these demands, 5G-compatible radio and baseband processing units, commonly referred to as gNBs, are increasingly embedding Artificial Intelligence (AI) models that enable dynamic traffic handling and intelligent resource allocation. Leveraging the extensive data generated by edge devices, RANs can refine operations using adaptive learning mechanisms (O-RAN.WG1, 2023).

The Open Radio Access Network (O-RAN) architecture has emerged as a transformative approach, offering flexibility, intelligence, and efficiency to fully unlock the potential of 5G New Radio (NR), as outlined by 3GPP and 5GPPP standards (Lacava *et al.*, 2024). A pivotal component of this architecture is the Radio Intelligent Controller (RIC), which incorporates advanced machine learning techniques through standardized interfaces and the Radio Network Information Base (RNIB) to optimize network performance (Marinova & Leon-Garcia, 2024).

One of the most promising machine learning techniques in this architecture is Federated Learning (FL), a decentralized approach that enables training machine learning models across distributed devices or servers without transferring raw data (McMahan *et al.*, 2017). A hierarchical variant of FL, known as Hierarchical FL (HFL), is particularly well-suited for the mobile-edge-cloud ecosystem (Zhou, Zheng, Huang, Shu & Jia, 2023; Xu *et al.*, 2023b; Luo *et al.*, 2020). When integrated with the layered structure of O-RAN, HFL enhances network performance, reduces resource consumption, and improves the QoS of user equipment (UE) while preserving

data privacy (Xu *et al.*, 2023b). Unlike traditional approaches, FL minimizes communication overhead by eliminating the need to centralize UE data, thus safeguarding user privacy. Advanced techniques such as MCORANFed (Singh & Nguyen, 2022a) and ORANFed (Singh & Nguyen, 2022b) have further optimized learning efficiency and resource utilization for single-layer FL in latency-sensitive O-RAN applications. However, these methods do not account for the mobility of local model trainers.

Deploying HFL in O-RAN is further complicated by the dynamic nature of mobile networks. Frequent handovers of UEs between base stations challenge the ability of rApps (applications running on the Non-Real-Time RIC) to manage an ever-changing set of devices during FL model training (Gao, Zhao & Yu, 2023; O-RAN.WG1, 2023). Although restarting FL training for a UE post-handover may enhance overall model convergence and efficiency, it introduces delays due to handover execution, potentially increasing the time required to train the global model (Prananto *et al.*, 2023). This challenge is particularly acute in scenarios involving autonomous vehicles, where wireless resources must be allocated efficiently under stringent latency and mobility constraints (Zhou *et al.*, 2023). High mobility, interference during handovers, fragmented spectrum, and the increasing number of communication devices further exacerbate these issues (Noor-A-Rahim *et al.*, 2022).

Unlike static networks, managing HFL in O-RAN, as illustrated in Fig. 4.1, requires addressing unique challenges associated with handover (HO) management (Karmakar *et al.*, 2023). These include: (i) dynamically selecting local trainers for each global round, (ii) deciding whether UEs transitioning to new base stations should continue participating in FL training, (iii) handling UEs that prematurely exit the training process, and (iv) incorporating HO-induced delays into the aggregation algorithm to synchronize local model updates effectively. Retaining all UEs throughout training may enhance model accuracy but significantly prolongs learning time, especially in environments with frequent handovers, such as UAVs or public transit systems. Delays introduced by resuming training at new base stations further slow the overall learning process.

Although recent works (Singh & Nguyen, 2022b,a) propose Non-Real-Time RIC-based frameworks to reduce communication costs and enhance data privacy by localizing information within Near-RT RICs, they do not address the extended learning delays caused by UE handovers across gNB-DUs in hierarchical training. Optimizing HFL training time and resource efficiency in the presence of dynamic UE mobility remains a critical and unresolved challenge.

To bridge this gap, this paper presents MHORANFed, an innovative HFL framework that mitigates overall training delay and communication resource costs through the joint optimization of local trainer selection and training resource allocation. Our key contributions are:

- A mathematical formulation of the optimization problem to jointly minimize training time and resource cost in a mobility-aware hierarchical FL framework for O-RAN;
- The development of MHORANFed, a novel algorithm designed to accommodate UE mobility within a hierarchical learning paradigm;
- A thorough convergence analysis of the proposed method, providing insights into solving the underlying non-convex optimization problem;
- Comprehensive simulations that demonstrate the effectiveness of MHORANFed compared to state-of-the-art approaches.

To the best of our knowledge, this is the first work to address the impact of soft inter-gNB-DU handovers on local trainers during HFL training. The remainder of this paper is organized as follows: Section 4.2 reviews related work, Section 4.3 presents the system model, Section 4.4 describes the proposed approach, Section 4.5 provides simulation results, and Section 4.6 concludes with directions for future research.

4.2 Related Works

Prior studies (Liu *et al.*, 2023), (Dinh *et al.*, 2021) have highlighted the unique characteristics and challenges of Federated Learning (FL) architectures, offering comprehensive overviews of existing approaches. The deployment of FL in O-RAN has been explored in works such as (Cao *et al.*, 2022), with particular attention to the open, software-defined, virtualized, and intelligent

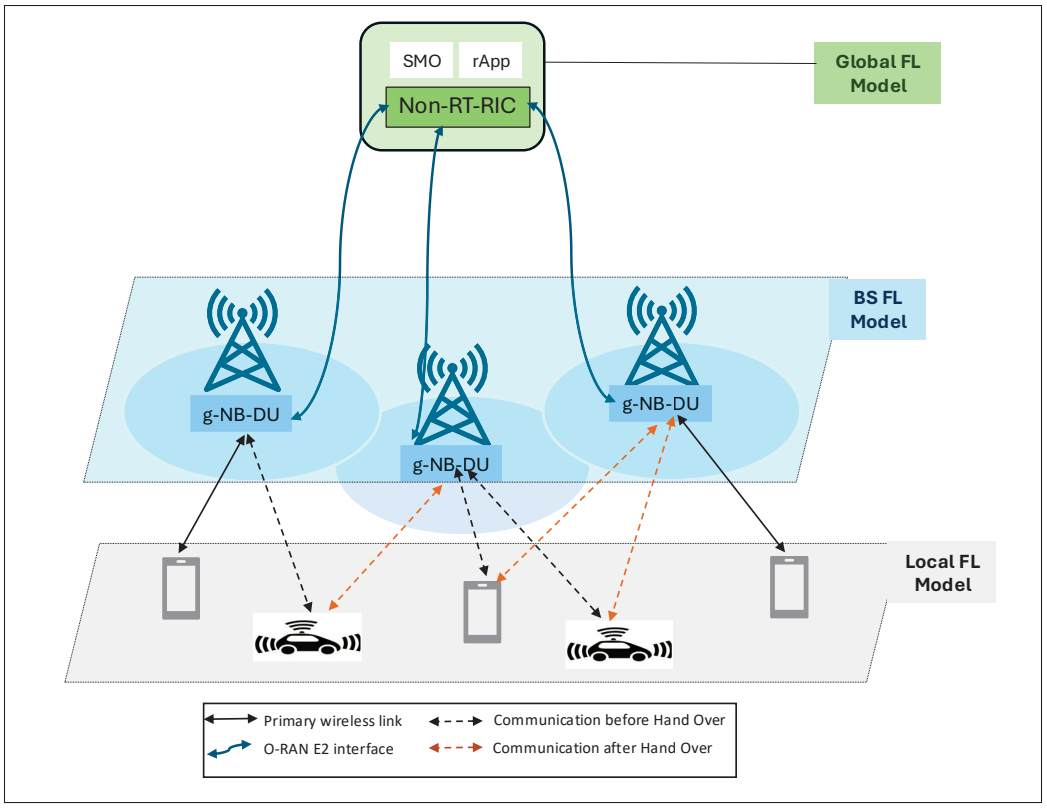


Figure 4.1 Federated Learning model with UE-BS handover and hierarchical aggregation at BS and Non-RT-RIC in O-RAN

design of the architecture. The challenge of imbalanced data distributions and learning latency in Hierarchical FL (HFL) across wireless multi-cell networks is addressed in (Liu *et al.*, 2022). This work proposes joint user association and wireless resource allocation algorithms for both IID and non-IID data, demonstrating improvements in convergence rates and learning accuracy. In contrast, (Xu *et al.*, 2023a) investigates FL deployment in decoupled O-RAN architectures to enhance computing capabilities. By enabling collaboration among Mobile Virtual Network Operators (MVNOs) for FL model training, it reduces the cost of data collection while boosting performance. A robust HFL framework tailored to the Internet of Vehicles (IoV) is presented in (Zhou *et al.*, 2023), which counters poisoning attacks through a reputation-based aggregation strategy and logarithmic normalization, offering enhanced robustness. Similarly, (Xu *et al.*, 2023b) optimizes worker aggregator placement and User Equipment (UE) assignment in Mobile Edge Computing (MEC) networks. Its proposed optimization framework and approximation

Table 4.1 Comparison with related literature on FL

Work	System Model Participants	FL Performance Objective(s)	Hierarch. FL	Resource Allocation	Handover of UEs	Converge. Analysis
Lin, Hosseinalipour, Michelusi & Brinton (2024)	BS, UE, Remote cloud	Model Loss, Delay	✓	✓	✗	✓
Xu *et al.* (2023a)	UE, BS (RIC)	Compute acceleration	✗	✗	✗	✗
Liu *et al.* (2022)	UE, BS, Cloud Server	Data Distribution, Delay	✓	✓	✗	✓
Singh & Nguyen (2022a)	O-RAN RICs	Resource Cost, Learning Time	✗	✓	✗	✓
Xu *et al.* (2023b)	BS, UE, Cloudlets	Resource Cost	✓	✓	✗	✓
Zhou *et al.* (2023)	Vehicles, BS, RSU, Cloud Server	Robust Model Aggregation	✓	✗	✗	✓
Feng *et al.* (2022)	UEs, Edge, Cloud Server	Convergence Accuracy	✓	✗	✗	✓
Our Work	**O-RAN RIC, BS, UE**	**Learning Time, Resource Cost**	✓	✓	✓	✓

algorithm substantially reduce FL implementation costs, as demonstrated through simulations and real-world testbeds. To address communication delays between edge and cloud servers, (Lin *et al.*, 2024) introduces a delay-aware HFL method that applies multiple stochastic gradient descent iterations alongside an adaptive control algorithm, achieving faster global model convergence and reduced resource consumption. In addressing mobility-related performance degradation, (Feng *et al.*, 2022) proposes a cluster-based HFL approach. More recently, (Zhao *et al.*, 2025) introduced a Reinforcement Learning (RL)-based client selection mechanism that adapts FL training to dynamic environments. Collectively, these studies tackle key challenges of FL implementation in 5G and beyond Radio Access Network (RAN) environments. Their use cases span resource optimization, hierarchical architectures, and resilience against adversarial threats. However, none of these works address the critical challenge of training HFL models

under dynamic handover (HO) conditions—an increasingly common scenario in mobile networks. This gap is especially pressing for high-mobility applications spanning terrestrial, aerial, and underwater domains, where rapid and reliable decision-making is vital. To fill this gap, we propose MHORANFed, a novel framework that explicitly incorporates HO dynamics into the HFL training process. Table 4.1 presents a detailed comparison between MHORANFed and existing state-of-the-art HFL and FL frameworks implemented in O-RAN and edge-cloud systems.

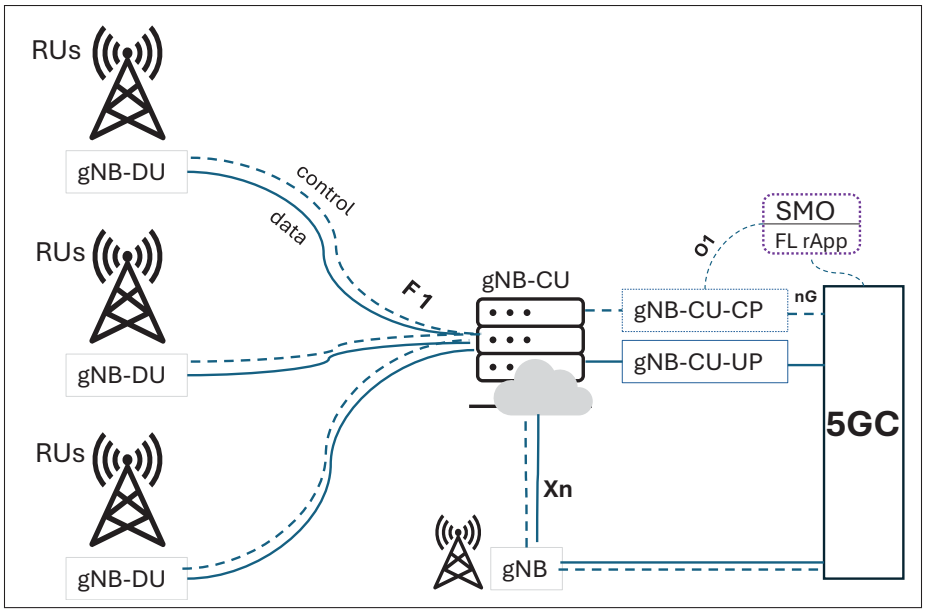


Figure 4.2 System components and standard interfaces in O-RAN

4.3 System Model and Problem Formulation

We consider a hierarchical O-RAN system disaggregated into Open Radio Unit (O-RU) at BS, Open Distributed Unit (O-DU) provided close to BS as gNB-DU, and Open Control Unit (O-CU) that controls multiple O-RUs and O-DUs. O-CU control plane (O-CUCP), as a separate entity from O-CU user plane (O-CUUP), monitors the performance KPIs of each UEs via their corresponding BSs. An r-App placed in Non-RT-RIC layer of O-RAN executes the FL training globally across the multiple BSs as illustrated in Fig. 4.2. In each time slot, four operations are implemented to train an FL model: (1) At the UE level, locally collected raw data by the UEs

are processed, (2) the model update parameters such as gradients are uploaded at respective BSs for local aggregation, (3) some of the UEs are handed over from one BS to another, and (4) global model update parameters from all the BSs are aggregated at the Non-RT-RIC.

The system consists of a set $\mathcal{N} = \{1, 2, ..., N\}$ of Base Stations (BS) each of which is associated with a gNB-DU. Please note that in this paper, the term BS is used interchangeably with gNB-DU since they are positioned together as depicted in Fig. 4.1. These gNB-DUs have limited processing power. Let $\mathcal{M} = \{1, 2, ..., M\}$ be the set of UEs in this network. The mobility of these devices results in frequent change of its associated BS invoking several HOs. However, a UE is allowed to be connected to a single BS in one time slot. Let $\mathcal{M}_n^t$ be the set of UEs connected with the BS n in the time slot t. The association between an instance of BS and the UEs follow O-RAN hierarchical definition, which can be modeled as $\sum_{n \in \mathcal{N}} \zeta_{nm} \leq 1 \ \ ; \forall m \in \mathcal{M}$. All the UEs participate in model training through FL using their available computational power on a pay per usage cost basis. In this model, we consider a generalized notion of cost that can be specified for a particular type of cost such as energy, battery consumption, etc. Further, all the BSs are connected with an instance of Non-RT-RIC in the Service and Management Orchestration (SMO) layer of O-RAN via a fiber link serving as the E2 interface of O-RAN. This dedicated link is separated from the data plane of Open Control Unit (O-CU). The Non-RT-RIC is hosted on a VM on pay per usage cost basis.

4.3.1 Hierarchical Federated Learning in O-RAN

In such a hierarchical federated learning model, the UEs are local trainers, the BSs are the edge model aggregators, and the Non-RT-RIC is the global model aggregator. Model update parameters are exchanged between the UEs and its associated BS periodically during the FL model training as in the traditional FL structure named FedAvg (McMahan *et al.*, 2017). For the sake of distinction, we call the model trained at the UEs as local FL models, at the BSs as edge FL models, and at the Non-RT-RIC as the global FL model. The aggregated model update parameters from each BS is sent to the Non-RT-RIC after every edge FL model aggregation.

Table 4.2 Summary of key notations

Notation	System Model Parameters
$\mathcal{M}$	Set of UEs
$\mathcal{N}$	Set of BS (g-NBs)
D_m	Dataset at m^{th} UE
D	Total size of all training data
R^m	Resource usage cost for one local iteration
R_m^{lc}	Total resource cost of local processing at all UEs
R^{bs}	Resource cost for model aggregation at BS
R^{lf}	Resource cost for communication
R^{bs}	Total resource usage cost of HFL
c_m	CPU cycles required to process per bit data
p_m	Processing power of m^{th} UE
p_{tr}	Uniform per unit transmission cost
R_n^g	Backhaul data rate from n^{th} BS to Non-RT-RIC
$F(w)$	Global Model Loss function at the Non-RT-RIC
T_{lc}^m	Local compute latency from m^{th} UE
T_m^n	Transmission delay from m^{th} UE to n^{th} BS
T_{ec}^n	Edge Model aggregation latency at n^{th} BS
T_m^{HO}	HO execution time for m^{th} UE
$\mathcal{T}_{cost}$	Total HFL model learning time
B_n	Available Bandwidth for allocation at n^{th} BS
Notation	**Input Parameters**
$s(d_m)$	Size of the local model vector at m^{th} UE
θ	Local Accuracy
ϵ	Prefixed global accuracy
ρ	Pareto parameter
$H_m^{n\to n'}$	Binary Hand Over (HO) parameter
L	Number of local iterations per global round
G	Number of global rounds
Notation	**Decision Variables**
x_m^g	Participation of m^{th} UE in g^{th} global round
a_m^n	Association of m^{th} UE with BS n
b_m	Bandwidth fraction allocation

Let $F(\mathbf{w})$ be the loss function of the global FL model. To obtain the optimized model $\mathbf{w}^*$, we perform iterative gradient descent method to minimize $F(\mathbf{w})$ over the model vector $\mathbf{w}$. At each BS, the loss function value of an edge FL model is calculated by aggregating its local model updates. Let $f_m^t(w)$ be the loss function of the local FL model associated with UE m and $f_n^t(w)$

be the loss function of the edge FL model associated with BS n at time slot t. Then,

$$F^t(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} f_n^t(w), \tag{4.1}$$

where

$$f_n^t(w) = \frac{1}{M} \sum_{m=1}^{M} f_m^t(w). \tag{4.2}$$

4.3.1.1 UE-BS Edge Model

At time slot t, the UEs serving as local trainers, process their individual raw data. Let $\mathcal{D}_m$ represent the dataset of UE m. To reduce communication overhead, instead of uploading the local model after every local iteration, the model parameters are uploaded to the associated BS only after every L rounds of local iterations. Let w_m^L denotes local FL model of m^{th} UE at L local iterations. Then the model update equation is:

$$w_m^L = w_m^{L-1} - \eta \nabla f_m(w_m^{L-1}); \forall m \in \mathcal{M}, \tag{4.3}$$

where η is the learning rate. The model update aggregation in each time slot at the n^{th} BS is performed as:

$$w_n^L = \frac{|\mathcal{D}_m|}{\mathbf{D}} \sum_{m \in \mathcal{M}_n} w_m^L; \forall n \in \mathcal{N}, \tag{4.4}$$

where $\mathbf{D} = \sum_{m \in \mathcal{M}_n} |\mathcal{D}_m|$.

4.3.1.2 BS – Non-RT-RIC Global Model

After each edge model aggregation, the BS sends its updated model to the Non-RT-RIC. The Non-RT-RIC receives the updated model from each BS and then aggregates as per the following

rule:

$$w^G = \frac{1}{N} \sum_{n \in \mathcal{N}} w_n^L; \forall n \in \mathcal{N}, \tag{4.5}$$

The global model is aggregated after every L local iterations and then communicated back to all the BS nodes. This process is called global round of FL model training.

4.3.2 Base Station Association and Bandwidth Assignment

The channel gain ($C_{m,n}^k(t)$) between a UE m and BS n in a particular time slot t for the radio Resource Block (RB) k is determined by two factors: large scale fading component ($l_{m,n}(t)$) and small scale fading component ($h_{m,n}^k(t)$). While the large scale fading is regulated by the distance between a UE and the corresponding BS it is connected to during one time slot and remains unchanged within this time slot, the small scale fading component is regulated by the variation between two contiguous time-slots. These time-varying components are related as:

$$C_{m,n}^k(t) = l_{m,n}(t)|h_{m,n}^k(t)|^2. \tag{4.6}$$

Co-channel interference may also occur when the same RB is allocated to multiple UEs. Therefore, as calculated in (Cao *et al.*, 2022), let $\delta_{m,n}^k(t)$ be the SINR at UE m from BS n in RB k then its uplink spectrum efficiency for time slot t can be given by

$$r_{m,n}(t) = log_2(1 + \delta_{m,n}^k(t)). \tag{4.7}$$

We define a binary variable a_m^n to denote the association of user m with BS n.

$$a_m^n = \begin{cases} 1 & if\ UE\ m\ is\ associated\ with\ BS\ n, \\ 0\ , & otherwise. \end{cases}$$

Let B_n be the available bandwidth for communicating the FL model training tasks of the associated UEs at the BS n. We define a decision variable $b_m^n(t) \in (0, 1) \subset \mathbb{R}$ to denote

the fraction of bandwidth of BS n allocated to the UE m within time slot t. Therefore, the instantaneous data rate between UE m and BS n in time slot t can be given as:

$$R_m^n(t) = a_m^n.b_m^n(t).B_n.r_m^n(t) \tag{4.8}$$

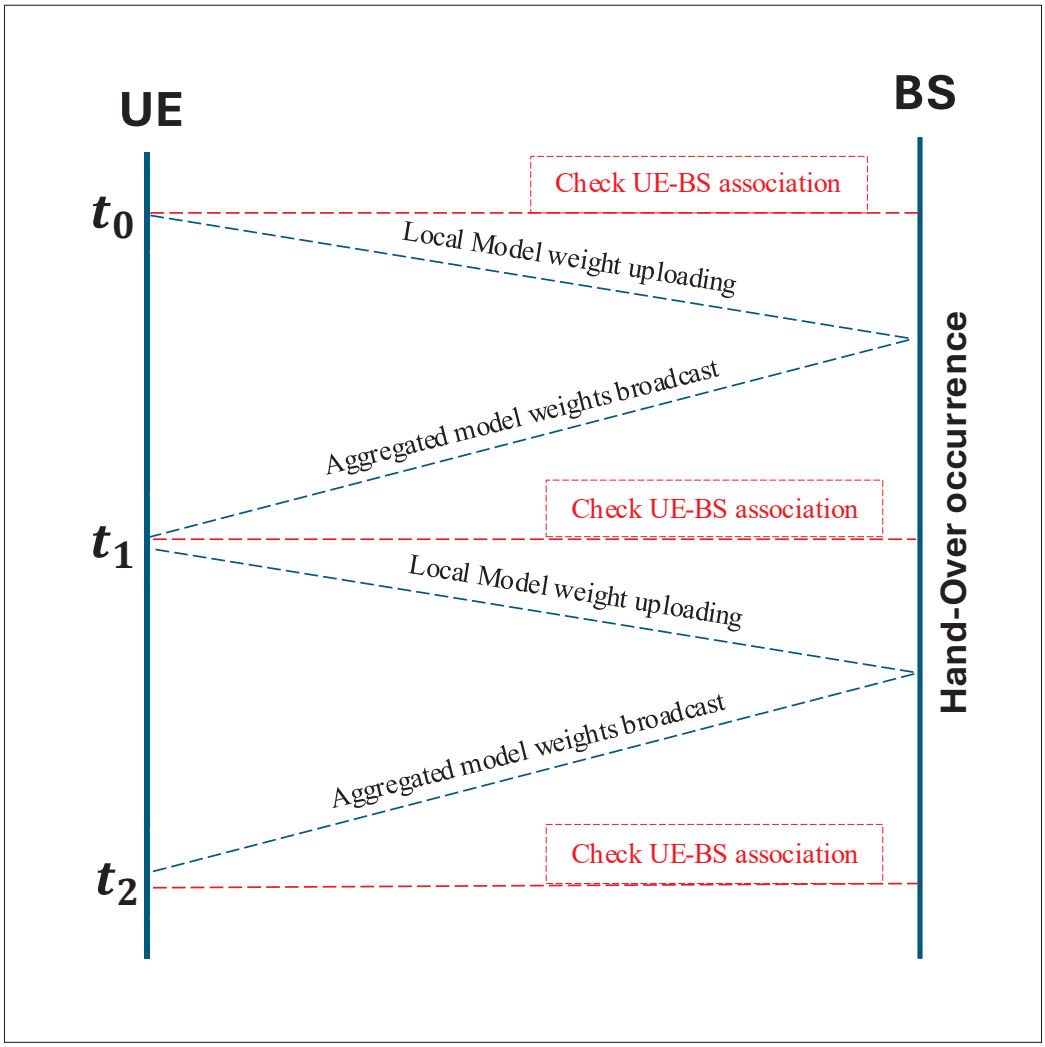


Figure 4.3 Sequence of steps among the HFL training nodes in every communication round

4.3.3 Inter gNB-DU Handover

We consider a soft handover in which the source and destination cells are associated with different gNB-DUs while the 5G core (5GC), which makes the HO decision, remains the same as illustrated in Fig. 4.2. This definition follows the technical specification of (3GPP, 2024) which is also analysed by (Kim *et al.*, 2021). In terms of physical deployment, the User Plane Function (UPF) of the 5GC can either be moved within the same UPF instance (intra-UPF) or across different UPF instances (inter-UPF). Although there is a dedicated m-Plane (defined in the O-RAN specifications) for such network management operations, this conditional inter-gNB-DUs handover results in change of associated UEs temporarily. In turn, this leads to fluctuations in channel gains and the BS throughput. Consequently, the bandwidth allocation of the updated UEs changes. A binary handover parameter $H_m^{n\to n'}$ indicates whether UE m is handed over from BS n to BS n'.

$$H_m^{n\to n'} = \begin{cases} 1 & if\ HO\ occurs\ from\ BS\ n\ to\ BS\ n', \\ 0\ , & otherwise\ (No\ HO). \end{cases}$$

Following the HO, the resulting association of a UE with a BS will also change and hence the effective data transmission rate changes. So, the updated subset of the UEs becomes $\mathcal{M}_{n'}(t)$ instead of $\mathcal{M}_n(t)$. Moreover, since the HO occurs during an ongoing model training, an overhead of handover time also needs to be accounted in the total FL training time. The FL rApp (located in the O-RAN SMO) records this HO event and calculates the time elapsed in rejoining the UE back into the FL model training. Let T_m^{HO} be the time elapsed in the handover execution of UE m. T_m^{HO} consists of internal delay caused by measurement reports exchange, TTT (Time-To-Trigger) parameters, etc. It should be noted that T_m^{HO} also includes delay arising from channel interference caused by high mobility of UEs in the adjacent BS.

4.3.4 HFL Resource Model

The FL model training incurs compute and communication resources consumption. In the first layer of HFL, training local FL model requires UE's compute resource, and bandwidth is required between the UEs and the corresponding BS for transmitting the model parameters.

4.3.4.1 Compute Resource

Let $s(D_m)$ be the size of the raw dataset to be processed at m^{th} UE, p_m be its processing power in cycles/sec, and c_m be the CPU cycles per bit to process the data. Then, the resource usage cost for one local iteration is given by:

$$R_m = s(D_m).c_m.(p_m)^2 \tag{4.9}$$

So, for L local iterations, the computation cost for UE m is $L.R_m$. Now, at the BS level, model parameters' aggregation is performed after every L local iterations (i.e., or a global round G). Naturally, $G.k = L$, meaning the total number of local rounds is always a whole number multiple of the number of global iterations. Therefore, the total resource cost of the HFL required for the local processing at all UEs is given by:

$$R^{lc} = G. \sum_{m \in \mathcal{M}} L.R_m \tag{4.10}$$

Further, let $s(d_m)$ be the size of the model parameters from the m^{th} UE. Then, at the BS level, the resource usage cost required for processing the local FL model aggregation is:

$$R^{bs} = \sum_{m \in \mathcal{M}, n \in \mathcal{N}} G.s(d_m).p_n \tag{4.11}$$

The compute resource usage cost at the Non-RT-RIC is negligible as the processing power there is not scarce. So, we do not model that part.

4.3.4.2 Communication Resource

Wireless resources are consumed for transmitting the model updates from UEs to their associated BS. Therefore, the communication resource usage cost incurred by the local FL model is given by:

$$R^{lf} = G. \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} a_m^n . b_m^n . B_n . r_m^n . p_{tr} \tag{4.12}$$

where, $p_{tr} \in \mathbb{R}^+$ is the uniform per unit bit transmission cost. The communication resource usage cost between BS and the Non-RT-RIC is comparatively negligible as it is provided by the non-scarce backhaul fiber link. Hence, the overall resource usage cost of this HFL can be summed up as;

$$\mathcal{R}_{cost} = R^{lc} + R^{bs} + R^{lf} \tag{4.13}$$

4.3.5 Learning Time Model

Due to the limited resources in UE-BS layer, both compute and transmission latency are the main performance factors. These latency occur at both the layers of HFL.

4.3.5.1 UE-BS Edge Layer

In this layer, the latency consists of local model processing at each participating UE and model parameters uploading by all of them to the respective BSs. The local compute latency can be modelled as:

$$T_m^{lc} = \frac{s(D_m).c_m}{p_m}. \tag{4.14}$$

Since a BS will have to wait for all its participating UEs to receive the model update before it can start the aggregation, the effective compute latency at BS n over all its UEs in time slot t is given by:

$$\max_{m \in \mathcal{M}_n(t)} \{T_m^{lc}\}. \tag{4.15}$$

Further, the delay in its model parameters' transmission to the corresponding temporal BS n can be modelled as:

$$T_m^n = \frac{s(d_m)}{R_m^n(t)}. \tag{4.16}$$

4.3.5.2 BS–Non-RT-RIC Layer

In this layer, the local FL model aggregation at BS and the global FL model aggregation at the Non-RT-RIC incur compute delays whereas the model parameters exchange adds transmission delay. A wired backhaul, i.e., a dedicated fiber link can be used for this communication, which offers a much higher speed than the wireless links between BSs and UEs. However, due to the straggler effect of FL training, the synchronous model aggregation may become a significant part of the learning time. Because of the model integrity, the size of the model parameters remain the same after the aggregation. Let $s(d)$ be the size of this model update and p_n be the processing power in bit per sec of the VM associated with the n^{th} BS. Then, the time required for processing the model is:

$$T_n^{ec} = \frac{s(d)}{p_n}. \tag{4.17}$$

The delay of transmitting the model from the n^{th} BS to the Non-RT-RIC is:

$$T_n^g = \frac{s(d)}{R_n^g}, \tag{4.18}$$

where R_n^g is the backhaul data rate from the n^{th} BS to the Non-RT-RIC. Since the Non-RT-RIC aggregates the model in the core network where compute resource is not scarce, the corresponding compute delay is negligible. Therefore, the overall learning time of this HFL can be summed up as:

$$\mathcal{T}_{cost} = \max_{m \in \mathcal{M}_n} \left\{ H_m^{n \to n'} . T_m^{HO} + T_m^{lc} + T_m^n \right\} + \max_{n \in \mathcal{N}} \left\{ T_n^{ec} + T_n^g \right\} \tag{4.19}$$

4.3.6 Problem Formulation

We define x_m^g a binary variable that decides whether the m^{th} UE which has been handed over from one BS to another will continue to take part in the g^{th} global round of HFL training or it will be dropped out of this particular round.

$$x_m^g = \begin{cases} 1\ ; & if\ m^{th}\ UE\ takes\ part\ in\ g^{th}\ global\ round, \\ 0\ ; & otherwise\ . \end{cases}$$

While a handed over UE may contribute to the global model in terms of accuracy, data heterogeneity, and convergence, it also increases the overall model learning time because of the time elapsed in handover and resuming the model update. In our proposed architecture, a rApp running in the Non-RT-RIC monitors the accuracy of the in-training model and the elapsed learning time following the completion of every global round to make the decision of accepting such UEs in the FL training as depicted in Fig. 4.3. The objective of this system is to minimize the compute and communication costs and the training latency of all the participating UEs over the required global rounds, which is modeled as follows:

$$\begin{aligned} cost(t) = \sum_{m \in \mathcal{M}} x_m^g \Big\{ & \sum_{n' \in \mathcal{N}} \sum_{m \in \mathcal{M}_{n'}(t)} H_m^{n \to n'} . T_m^{HO} \\ & + \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}_m(t)} (\rho . \mathcal{T}_{cost} + (1 - \rho) . \mathcal{R}_{cost}) \Big\} \end{aligned} \tag{4.20}$$

The first term of this cost function represents the delay incurred solely due to HO execution, while the second term captures the training delay along with the associated resource usage cost. Notably, the impact of HO execution delay also extends to the second term, resulting in a compounded effect on the overall cost function. This function serves as the objective function

for the subsequent optimization problem.

$$\min_{\mathbf{x,\,a,\,b}} \sum_{t=1}^{G} cost(t) \tag{4.21}$$

subject to:

$$\sum_{m \in \mathcal{M}_n} a_m^n(t).b_m^n(t) \leq 1;\ \forall n \in \mathcal{N}, t = \{1, 2, ..., G\}, \tag{4.21a}$$

$$\sum_{m \in \mathcal{M}_n}^{M} b_m^n(t) = 1;\ \ \forall n \in \mathcal{N}, \tag{4.21b}$$

$$\sum_{n \in \mathcal{N}} a_m^n(t) = 1;\ \ \forall m \in \mathcal{M}_n, \tag{4.21c}$$

$$\sum_{n,n' \in \mathcal{N}} a_m^n(t).H_m^{n \to n'} \leq 1\ \ ;\forall m \in \mathcal{M}, \tag{4.21d}$$

$$\sum_{m \in M, n \in N} R_m^n(t).x_m^g \leq r_{m,n}(t)\ \ ;\forall g \leq G, \tag{4.21e}$$

$$\sum_{n \in \mathcal{N}} \zeta_{nm} \leq 1\ \ ;\forall m \in \mathcal{M}, \tag{4.21f}$$

$$a_m^n, x_m^g \in \{0, 1\}, \tag{4.21g}$$

$$b_m^n(t) \in (0, 1) \tag{4.21h}$$

Constraint (4.21a) guarantees that the sum of bandwidth fractions allocated to all the local trainers does not exceed the available bandwidth at each BS. Constraint (4.21b) defines the integrity of bandwidth allocation for each BS. Constraint (4.21c) ensures that a UE is associated with a single BS in each time slot. Constraint (4.21d) states that a UE can be handed over to at maximum one BS in each time slot. Constraint (4.21e) denotes that the sum of selected UEs' data rates obtained throught he bandwidth allocation must not exceed the spectrum uplink efficiency in any global round. (4.21f) reflects the association constraint of O-RAN hierarchy. (4.21g and 4.21h) are the defining conditions on the decision variables.

Algorithm 4.1 Recovering Feasible Integer Solution using PCA

Input: Relaxed SDP solution W_m (positive semi-definite matrix)
Output: Approximate integer solution $\hat{x}, \hat{a}, \hat{b}$

Step 1: Compute Principal Components;
Compute the eigenvalue decomposition: $W_m = U\Lambda U^T$;
Select the top principal component: $v_1 = U[:, 1]$ (corresponding to the largest eigenvalue);
Step 2: Extract Initial Integer Approximation;
for $i = 1$ ***to*** n **do**
    Assign $\tilde{x}_m^g(i) = \text{sign}(v_1(i))$ (Convert to binary by thresholding at 0);
end for
Step 3: Randomized Rounding;
for *each* m, n **do**
    Compute probability $p_m^n = \frac{|v_1(m,n)|}{\sum_{m'} |v_1(m',n)|}$;
    Set $\hat{a}_m^n$ as a Bernoulli random variable with probability p_m^n;
end for
Step 4: Feasibility Refinement;
for *each constraint in (4.21)* **do**
    Project the solution onto the feasible set;
    Adjust $\hat{a}, \hat{b}, \hat{x}$ using constraint enforcement;
end for
Step 5: Return Recovered Integer Solution;
Return $\hat{x}, \hat{a}, \hat{b}$;

4.4 Proposed Solution: MHORANFed

The problem (4.21) is a mixed-integer non-linear programming (MINLP) model. The objective function is non-convex and the constraint (4.21a) contains bilinear terms (product of decision variables **a** and **b**). Obtaining the exact optimal solution of this NP-Hard problem is intractable with existing mathematical solvers. Nonetheless, Semi Definite Programming (SDP)(Vandenberghe & Boyd, 1996) relaxation is known to provide tight bounds. Therefore, we relax the constraints and the objectives to a convex semi-definite form.

Fig. 4.4 outlines our proposed solution approach to the joint optimization problem (4.21) aimed at minimizing the total HFL training cost under bandwidth and handover-aware local trainers' selection constraints. The problem is first reformulated as a SDP problem (4.22), which is then approximately solved using Algorithm 4.1 to obtain a feasible solution. Following this, the system performs post-handover local trainer selection and allocates bandwidth among the selected devices. Finally, Algorithm 4.2 resumes to continue with the learning rounds based on the optimized configuration.

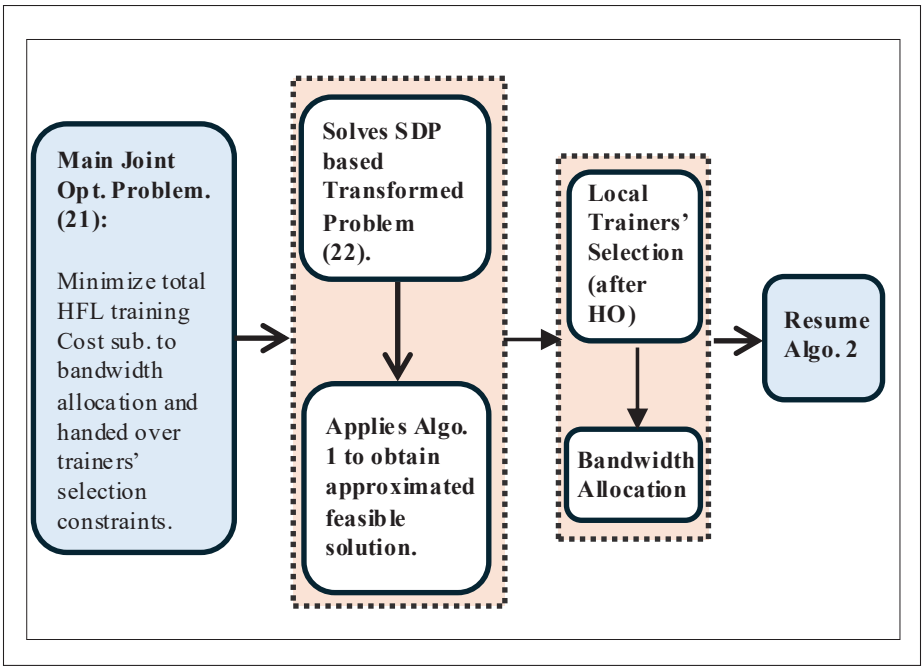


Figure 4.4 Proposed solution schema

To relax the bilinear term $(a_m^n \cdot b_m^n)$, we introduce a new variable W_m defined as:

$$W_m = \begin{bmatrix} 1 & a_m^n \\ a_m^n & b_m^n \end{bmatrix}, \quad W_m \succeq 0.$$

So, the constraint (4.21a) transforms to

$$\sum_{m \in \mathcal{M}_n} W_m \leq 1, \quad \forall n \in \mathcal{N}, t = \{1, 2, \ldots, G\}$$

We also relax the binary decision variables into continuous real variables as: $x_m^g \in [0, 1], \quad a_m^n \in [0, 1], \quad b_m^n \in [0, 1]$ to make it solvable through a mathematical solver (Diamond & Boyd, 2016). Then, a rounding technique is applied to obtain the binary values of these variables. In particular, we performed principal component analysis (PCA) on the obtained matrix W_m solution and iteratively adjusted its feasibility as shown in Algorithm 4.1. The PCA based rounding is preferred over other rounding methods as it offers a principled and computationally efficient way to project high-dimensional approximated solutions onto a lower-dimensional feasible space. (Lee, Song & Honorio, 2022)

4.4.1 Optimality Gap Analysis

The SDP based relaxation leads to a sub-optimal approximation which is a lower bound on the exact solution as can be directly inferred from its definition as follows:

The non-convex bilinear inequality constraint (4.21a):

$$a_m^n(t) \cdot b_m^n(t) \leq 1$$

is relaxed using a positive semidefinite matrix:

$$W_m^n(t) = \begin{bmatrix} 1 & a_m^n(t) & b_m^n(t) \\ a_m^n(t) & a_m^n(t)^2 & \theta_m^n(t) \\ b_m^n(t) & \theta_m^n(t) & b_m^n(t)^2 \end{bmatrix}, \quad W_m^n(t) \succeq 0$$

where $\theta_m^n(t)$ serves as a convex surrogate for the product $a_m^n(t) \cdot b_m^n(t)$.

We replace the rank-1 matrix constraint

$$W_m^n(t) = \mathbf{v}_m^n(t)\mathbf{v}_m^n(t)^\top, \quad \text{with} \quad \mathbf{v}_m^n(t) = \begin{bmatrix} 1 \\ a_m^n(t) \\ b_m^n(t) \end{bmatrix}$$

by a relaxed convex constraint $W_m^n(t) \succeq \mathbf{v}_m^n(t)\mathbf{v}_m^n(t)^\top$, allowing $W_m^n(t)$ to lie in the convex hull of rank-1 matrices. This enlargement of the feasible region leads to the SDP:

$$\min_{\mathbf{x},\mathbf{a},\mathbf{b},\mathbf{W}} \quad \sum_{t=1}^{G} \text{cost}(t) \tag{4.22}$$

$$\text{s.t.} \quad W_m^n(t) \succeq 0, \quad \forall m, n, t \tag{4.22a}$$

(plus relaxed versions of other constraints from (4.21))

The optimal value f_{sdp}^* of this convex relaxation satisfies:

$$f_{\text{sdp}}^* \leq f^*$$

where f^* is the optimal value of the original non-convex problem. This is because the relaxed SDP includes all feasible integer points and additional fractional ones, resulting in a lower bound on the true optimum. The gap between f^* and f_{sdp}^* is the optimality gap.

In Lemma 1 of our convergence analysis, we show that under a set of conditions on the model loss functions and its learning rate, this optimality gap lies within the controllable range of convergence.

Using this approximated solution, we design an algorithm called Mobility Aware Hierarchical Federating Learning for O-RAN (MHORANFed). The global training loop iterates until the global model accuracy ($\epsilon \in (0, 1)$) reaches a predefined threshold ϵ^* as outlined in Algorithm (4.2). By considering the mobility of UEs and the hierarchical structure of O-RAN,

MHORANFed aims to achieve efficient and accurate federated learning in dynamic network environments.

Algorithm 4.2 Mobility-aware Hierarchical FL for O-RAN (MHORANFed)

Input: Untrained global FL model w; Set of participating UEs $\mathcal{M}$ and BS-gNB servers $\mathcal{N}$; HO occurrence at concerned UEs $H_m^{n\to n'}$; Stopping criteria ϵ^*
Output: Trained global HFL model w^*

while $\epsilon \geq \epsilon^*$ **do**
  for *global round* $g = 1$ ***to*** G **do**
    Non-RT-RIC rApp solves (4.22);
    Non-RT-RIC assigns bandwidth $\hat{\mathbf{b}}$ to UEs $\hat{\mathbf{x}}$, $\hat{\mathbf{a}}$;
    for *local iteration* $\ell = 1$ ***to*** L **do**
      UE m downloads the edge model $w_m^{\ell-1}$ from its associated BS-gNB;
      UE trains local model w_m^{ℓ} using Equation (4.3);
      BS n aggregates local models to get w_n^{ℓ} via Equation (4.4);
    end for
    BS-gNB servers upload aggregated models to the Non-RT-RIC;
    Non-RT-RIC aggregates models to get w^g using Equation (4.5);
    Non-RT-RIC updates global model accuracy;
    rApp utilizes $H_m^{n\to n'}$ to update current UE associations;
  end for
  Evaluate convergence criterion ϵ;
end while

By utilizing the optimal set of local trainers associated with a BS and the handed-off UEs allowed back in the model training in each global round, MHORANFed mitigate the bias caused by handover and in favour of the BS having a relatively higher number of local trainers. Moreover, the synchronous communication resulting in the delay calculated as modeled by (4.15), ensures that the optimal solution has the total learning time with ρ times Pareto importance as defined in (4.20).

4.4.2 Complexity Analysis

Algorithm 4.2 consists of two main loops. The outer-loop depends on the number of global rounds (G) having a time complexity of $\mathcal{O}(G)$. According to (Ma *et al.*, 2017), the MHORANFed

performance will be unaffected as long as the convergence time of FL global iterations is upper-bounded by $O(log(1/\epsilon^*)$, where ϵ^* is the model accuracy. The inner-loop has the complexity of $O(L)$, where L is the number of local iterations. Let K_{SDP} be the approximated iterations of SDP then $O(K_{SDP})$ Vandenberghe & Boyd (1996) is the computational complexity of the optimization problem in (4.21), as drawn in Algorithm 4.2. Therefore, the overall complexity of MHORANFed is $\{M.O(L) + N.O(K_{SDP})\}.O(log(1/\epsilon^*))$. Here, $M = |\mathcal{M}|$ is the number of UEs and $N = |\mathcal{N}|$ is the number of BSs.

4.4.3 Convergence Analysis

Lemma 1 (Boundary of Convergence): *If $F(w^*)$ and $F(w^G)$ be the final loss function values corresponding to f^* and f^*_{sdp} i.e. the optimal and the approximated solutions respectively, then*

$$f^* - f^*_{\text{sdp}} \geq \mathbb{E}[F(w^G) - F(w^*)]$$

provided the following conditions hold:

- $\eta < \frac{1}{R}$,
- $\eta\mu > 1 - \sqrt[L]{\frac{1}{4M}}$.

Proof: After G global rounds of communication between the Non-RT-RIC and the set of BSs, the expected optimality gap is bounded by:

$$\mathbb{E}[F(w^G) - F(w^*)] \leq \beta[F(w^0) - F(w^*)] + (1-\beta)\frac{L\eta p}{4\gamma}$$

where $\gamma = (1-\eta\mu)^L$, and $\beta = \left(\frac{2M\gamma}{L}\right)^G$.

In each round, the left-hand side of the above inequality is governed by two key terms from the SDP-based approximated solution of (4.21):

- the geometric decay term: $\beta[F(w^0) - F(w^*)]$,
- and the residual error floor: $(1-\beta)\frac{L\eta p}{4\gamma}$.

The term $\beta = \left(\frac{2M\gamma}{L}\right)^G$ depends exponentially on the number of global rounds G, and shrinks rapidly as G increases, provided that:

$$\gamma = (1 - \eta\mu)^L < 1$$

This holds true when the learning rate η and the local smoothness constant μ satisfy:

$$\eta\mu > 1 - \sqrt[L]{\frac{1}{4M}}$$

Together, these conditions ensure that:

- the geometric decay dominates initially, reducing the gap quickly,
- the residual floor becomes small with careful tuning of η,
- and the SDP relaxation of the original problem provides a tight lower bound on $F(w^*)$, enabling estimation of the gap:

$$\mathbb{E}[F(w^G) - F(w^*)] \leq f^* - f^*_{\text{sdp}}$$

Hence, this theoretical bound directly connects the choice of algorithmic parameters to the rate of convergence and the quality of the approximate solution.

Theorem 1 (UE-BS Edge Layer): *Assuming the global loss function $F(w)$ to be R-smooth and μ-strongly convex (true for the standard FL convergence analysis (Wang* et al.*, 2019a), and that each UE performs L local updates before uploading its updated model weights to the corresponding BS, if the following conditions are satisfied:*

- $\eta < \frac{1}{R}$,
- $\eta.\mu > 1 - \sqrt[L]{\frac{1}{4M}}$,
- $F(w^o) - F(w^*) > \frac{L.M.p}{4.(1-\eta.\mu)^L}$,

then we have:

$$\begin{cases} \frac{2M}{L}(1-\eta\cdot\mu)^L \in (0,1), \\ \mathbb{E}[F(w^L)-F(w^*)] \leq (1-\eta\cdot\mu)^L[F(w^0)-F(w^*)] \\ +\frac{L\cdot\eta\cdot p}{2} \end{cases} \tag{4.23}$$

where w^L denotes the model weights after L rounds of local updates and w^o denotes the initial weight parameter.

Proof: We derive the convergence of L local updates first. By utilising the assumptions of strong convexity and smoothness for each local update $l \in 1, 2, ...L$, we have:

$$\begin{aligned} &\mathbb{E}[F(w^l)-F(w^*)] \\ &\leq F(w^{l-1})-F(w^*) \\ &-\eta\mathbb{E}[<\nabla F(w^{l-1}), \nabla f(w^{l-1};q_l)>] \\ &\quad+\frac{L\eta^2}{2}\mathbb{E}[||\nabla f(w^{l-1};q_l)||^2] \\ &\leq F(w^{l-1})-F(w^*)-\frac{\eta}{2}||\nabla F(w^{l-1})||^2 \\ &\quad+\frac{\eta}{2}\mathbb{E}[||\nabla F(w^{l-1})-\nabla f(w^{l-1};q_l)||^2] \\ &\leq F(w^{l-1})-F(w^*)-\frac{\eta}{2}||\nabla F(w^{l-1})||^2 \\ &\quad+\frac{\eta p}{2} \text{ (using the graded variance)} \\ &\leq F(w^{l-1})-F(w^*)-\eta\mu[F(w^{l-1})-F(w^{)}]+\frac{\eta p}{2} \\ &\leq (1-\eta\mu)[F(w^{l-1})-F(w^*)]+\frac{\eta p}{2} \end{aligned} \tag{4.24}$$

This gives us a convergence bound for each local iteration $l \in 1, 2, ..., L$. By extending this for all the local updates and telescoping, we have after L local updates:

$$
\begin{aligned}
&\mathbb{E}[F(w^L) - F(w^*)] \\
&\leq (1 - \eta\mu)[F(w^{L-1}) - F(w^*)] + \frac{\eta p}{2} \\
&\leq (1 - \eta\mu)[(1 - \eta\mu)[F(w^{L-2}) - F(w^*)] \\
&+ \frac{\eta p}{2}] + \frac{\eta p}{2} \\
&\ldots\ldots\ldots \textit{by telescoping with eq.}\ (4.24) \\
&\leq (1 - \eta\mu)^L[F(w^0) - F(w^*)] + \\
&\frac{\eta p}{2} \sum_{l=1}^{L} (1 - \eta\mu)^{l-1} \\
&\leq (1 - \eta\mu)^L[F(w^0) - F(w^*)] + \\
&\frac{\eta p}{2} \frac{1 - (1 - \eta\mu)^L}{1 - (1 - \eta\mu)} \\
&\leq (1 - \eta\mu)^L[F(w^0) - F(w^*)] + \\
&\frac{\eta p}{2} \frac{L\eta\mu}{1 - (1 - \eta\mu)} \\
&\leq (1 - \eta\mu)^L[F(w^0) - F(w^*)] + \frac{L\eta p}{2}
\end{aligned} \tag{4.25}
$$

At the Non-RT-RIC, the concerned FL Manager rApp performs global model aggregation after every m updated model weights at time-slot t.
Following it we have: $w^t = \frac{L}{M} \sum_{i=1}^{\frac{M}{L}} w_i^L$, where w_i^L denotes the local model update in the i^{th} UE after L local iterations.

Theorem 2 (BS-Non-RT-RIC Layer): *After G global rounds between the Non-RT-RIC and the set of BSs, the convergence bound of MHORANFed is:*

$$
\mathbb{E}[F(w^T) - F(w^*)] \leq \beta[F(w^0) - F(w^*)] + (1 - \beta)\frac{L\eta p}{4\gamma} \tag{4.26}
$$

where $\gamma = (1-\eta\mu)^L$, *and* $\beta = (\frac{2M\gamma}{L})^G$.

Proof: Following the definition of the weight update rule of w^t for every global round, we have:

$$
\begin{aligned}
&\mathbb{E}[F(w^t) - F(w^*)] \\
&\leq \frac{L}{M}\sum_{i=1}^{M/L}[F(w_i)^L - F(w^*)] \\
&\leq \frac{L}{M}\sum_{i=1}^{M/L}[(1-\eta\mu)^L(F(w_i) - F(w^*)) + \frac{LMp}{2}] \\
&\leq \frac{M}{L}[(1-\eta\mu)^L(F(w^0) - F(w^*)) + \frac{L\eta p}{2}] \\
&\leq \frac{M}{L}[(1-\eta\mu)^L(F(w^0) - F(w^{t-1}) \\
&+ F(w^{t-1}) + F(w^*)) + \frac{L\eta p}{2}] \\
&\leq \frac{M}{L}(1-\eta\mu)^L(F(w^0) - F(w^{t-1})) \\
&+ \frac{M}{L}(1-\eta\mu)^L(F(w^{t-1}) - F(w^*)) + \frac{ML\eta p}{2} \\
&\leq \frac{M}{L}(1-\eta\mu)^L(F(w^{t-1}) - F(w^*)) \\
&+ \frac{M}{L}(1-\eta\mu)^L(F(w^{t-1}) - F(w^*)) + \frac{ML\eta p}{2} \\
&\leq (\frac{2M}{L}(1-\eta\mu)^L)[F(w^{t-1}) - F(w^*)] + \frac{ML\eta p}{2}
\end{aligned}
\tag{4.27}
$$

Therefore, the after G global rounds, the convergence bound comes down to:

$$
\begin{aligned}
&\mathbb{E}[F(w^G) - F(w^*)] \\
&\leq (\frac{2M}{L}(1-\eta\mu)^L))[F(w^{G-1}) - F(w^*)] + \frac{ML\eta p}{2} \\
&.....telescoping\ by\ eq.(4.27) \\
&\leq (\frac{2M}{L}(1-\eta\mu)^L))^G[F(w^0) - F(w^*)] \\
&+ \frac{(1-\frac{2M}{L}(1-\eta\mu)^L)^G ML\eta p}{2(1-\frac{2M}{L}(1-\eta\mu)^L)} \\
&\leq (\frac{2M}{L}(1-\eta\mu)^L)^G[F(w^0) - F(w^*)] \\
&+ \frac{(1-\frac{2M}{L}(1-\eta\mu)^L)^G ML\eta p}{\frac{4M}{L}(1-\eta\mu)^L)} \\
&\leq (\frac{2M}{L}(1-\eta\mu)^L)^G[F(w^0) - F(w^*)] \\
&+ \frac{L\eta p}{4(1-\eta\mu)^L}.(1-(\frac{2\eta}{L}(1-\eta\mu)^L)^G)
\end{aligned} \tag{4.28}
$$

To simplify eq. (4.28), replacing $\gamma = (1-\eta\mu)^L$ and $\beta = (\frac{2M\gamma}{L})^G$. Then, we have:

$$
\mathbb{E}[F(w^T) - F(w^*)] \leq \beta[F(w^0) - F(w^*)] + \frac{L\eta p}{4\gamma}(1-\beta) \tag{4.29}
$$

4.5 Numerical Results and Analysis

4.5.1 Simulation Settings

We simulate 5 BSs and 100 UEs randomly in a distributed coverage region. At the beginning, a random subset of 20 UEs are connected to each BS such that the load of all BSs are balanced

Table 4.3 Simulation Parameters

Parameter	Value
System Model Parameters	
Number of BSs	5
Number of UEs	100 (random placement)
Initial UE-BS Association	20 UEs per BS (balanced load)
HO Events	After 20, 40, 70 rounds: 5, 3, and 6 UEs switch
HO Repetitions	10 runs (average results)
Channel Model	Small- and large-scale fading
Path Loss Model	$128.1 + 37.6 \log_{10}(d)$
Fading Type	Rayleigh
UE CPU	$\sim \mathcal{U}(1.0,\ 2.5)$ GHz
BS CPU	$\sim \mathcal{U}(2.0,\ 3.0)$ GHz
Uplink Bandwidth	50 MHz per BS
Fiber Bandwidth	20 MHz (2nd HFL layer)
Loss Function	Cross-Entropy (Non-convex), Mean Square Error (Convex)
Input Parameters (Datasets and Accuracy)	
CIFAR-10	50k train, 10k test
MNIST	60k train, 10k test
Network Traffic	160k train, 40k test
Data Allocation	IID and non-IID across UEs
$\epsilon^*_{\text{MNIST}}$	93% (CNN), 91% (SVM)
$\epsilon^*_{\text{CIFAR-10}}$	78% (CNN), 81% (SVM)
$\epsilon^*_{\text{Traffic}}$	90% (LSTM)

in terms of the bandwidth required for model update parameters' transmission. After the completion of 20, 40, and 70 global rounds, a subset of 5, 3, and 6 UEs respectively opt for HO to their nearest BS. A similar occurrence of HO is set in the experiment for a total of 10 times and the average performance of MHORANFed is plotted to counter the randomness bias. A wireless channel model between the UEs and the respective BSs is considered with both small and large scale fading components. We set the path loss model of $128.1 + 37.6 log_{10}(d)$ for the large scale fading and set Rayleigh distribution for the small scale fading. The processing power of UEs is set uniformly in the range from 1.0 GHz to 2.5 GHz and that of BSs is in the range from 2.0 GHz to 3 GHz. Uplink bandwidth capacity is 50 MHz for each BS. For the second layer of the HFL, the available fiber bandwidth is 20 MHz. A summary of the simulation settings are presented in Table 4.3.

Table 4.4 Number of global communication rounds to achieve the target accuracy with different Pareto trade-offs and learning rates

Dataset	**Method**	ρ=0.5 η=0.01	ρ=0.5 η=0.05	ρ=0.01 η=0.01	ρ=0.01 η=0.05	ρ=0.99 η=0.01	ρ=0.99 η=0.05
CIFAR-10	All-Dropped	114	116	112	115	118	117
CIFAR-10	MHORANFed	**101**	**102**	**103**	**100**	**104**	118
CIFAR-10	Random	113	115	110	114	116	119
CIFAR-10	HFL-0	112	114	108	113	117	115
CIFAR-10	DFL	110	117	111	116	115	**110**
CIFAR-10	MACFL	109	113	110	112	114	116
CIFAR-10	FedPPO	108	111	109	110	113	114
MNIST	All-Dropped	133	130	127	134	132	131
MNIST	MHORANFed	**113**	**115**	**114**	**116**	126	**117**
MNIST	Random	129	128	125	132	130	129
MNIST	HFL-0	124	126	121	128	129	128
MNIST	DFL	126	131	123	130	128	127
MNIST	MACFL	121	127	122	129	127	125
MNIST	FedPPO	119	123	120	124	**119**	122
Traffic	All-Dropped	56	55	52	53	58	57
Traffic	MHORANFed	**39**	**40**	43	**41**	**36**	46
Traffic	Random	50	51	48	49	54	53
Traffic	HFL-0	47	49	46	48	52	50
Traffic	DFL	44	53	45	50	51	54
Traffic	MACFL	42	46	**38**	47	49	51
Traffic	FedPPO	41	44	42	45	47	**42**

The standard CIFAR-10 dataset (Krizhevsky, Nair & Hinton) consisting of 50,000 training images and 10,000 train images and the MNIST dataset (Deng, 2012) containing 60,000 training and 10,000 test images are adopted for the multi-class classification problem. These datasets are distributed onto all 100 UEs with skewed class labels to represent non-IID scenario and uniformly using stratified sampling method to represent IID scenario at the FL training initialization. The stopping criteria for MNIST, defined as the model accuracy ϵ^*, is set as 93% which is obtained through centralized ML model training with a 2 layer CNN model and 91% model accuracy with SVM. The ϵ^* for CIFAR-10 is 78%, as obtained through centralized CNN model with 2 layers and 81% with SVM. A time series dataset taken from (Rojas, 2020) is also processed for predicting network traffic. The categorical cross-entropy is the loss function for

the classification problem while the Mean Square Error (MSE) is the loss function for the traffic prediction problem.

We trained the models under two different dataset distributions to incorporate the impact of data heterogeneity. (i) IID case: the whole train and test dataset as described above us uniformly distributed across the associated local trainers. We also employ stratified sampling to maintain a proportional class label distribution for the MNIST and CIFAR-10 dataset. Similarly, we maintain the same time duration for the traffic volume dataset. (ii) Non-IID case: we utilized skewed class distribution for the image classification task and non-uniform time-span distribution for the traffic prediction task.

4.5.2 Baselines

The following baseline schemes are implemented for this comparative performance analysis:

- **'All-Dropped':** This scheme excludes all the UEs which undergo HO in HFL.
- **'MHORANFed':** The near optimal solution obtained through our proposed algorithm in this paper.
- **'Random':** In this scheme, a random set of UEs are selected consisting of one or more handed over UEs and dropped UEs.
- **'HFL-0':** This is the algorithm derived from (Liu *et al.*, 2022) and adapted to suit the comparable framework in this paper. In this variant of HFL, UEs' handovers are not considered.
- **'DFL':** This is a delay aware HFL (Lin *et al.*, 2024) having sub-linear convergence rate and no consideration of handovers.
- **'MACFL':** Taking into account the impact of users' mobility on the FL training in wireless networks, MACFL (Mobility-Aware Cluster FL) (Feng *et al.*, 2022) serves as the closest baseline.
- **'FedPPO':** We trained a reinforcement learning (RL) based client selection strategy for the hierarchical FL by utilizing Federated Proximal Policy Optimization (FedPPO) (Zhao *et al.*, 2025).

4.5.3 Key Metrics

The main metrics used for comparison are as follows.

- **Convergence:** This fundamental property assesses an FL method's ability to attain the threshold performance over the number of global rounds. A higher convergence rate results in a smaller number of global rounds as it directly translates to lower training time. We compare the convergence rate of all the baselines for both types of learning tasks over the IID and non-IID cases.
- **Training Cost:** As defined in (4.21), there are two main components of the training cost: i) learning time shich is the total time taken to train the final global model, and ii) total cost which is calculated based on compute and bandwidth resources in both the layers of HFL. A lower training cost is preferred as it reduces the burden on already constrained edge system with O-RAN's tight closed loop timescale.

4.5.4 Performance Evaluation

Now, we examine the performance of MHORANFed in light of the defined metrics, the relevant baselines, learning tasks, and data distributions under the described experimental settings.

4.5.4.1 Impact of local trainers' participation method

Keeping the value of $\rho = 0.5$ (implying a balanced trade-off between learning time and resource usage costs) for MHORANFed, we compared the performance of the baselines in terms of training costs (Figure 4.6) and convergence rate (Fig. 4.5). The key differentiator is their local trainers' selection method. We can also observe the exact number of global rounds required by each method in Table 4.4. In general MHORANFed outperforms clustering based MACFL and RL based FedPPO among other HFL variants, advocating the importance of our proposed joint optimization based selection.

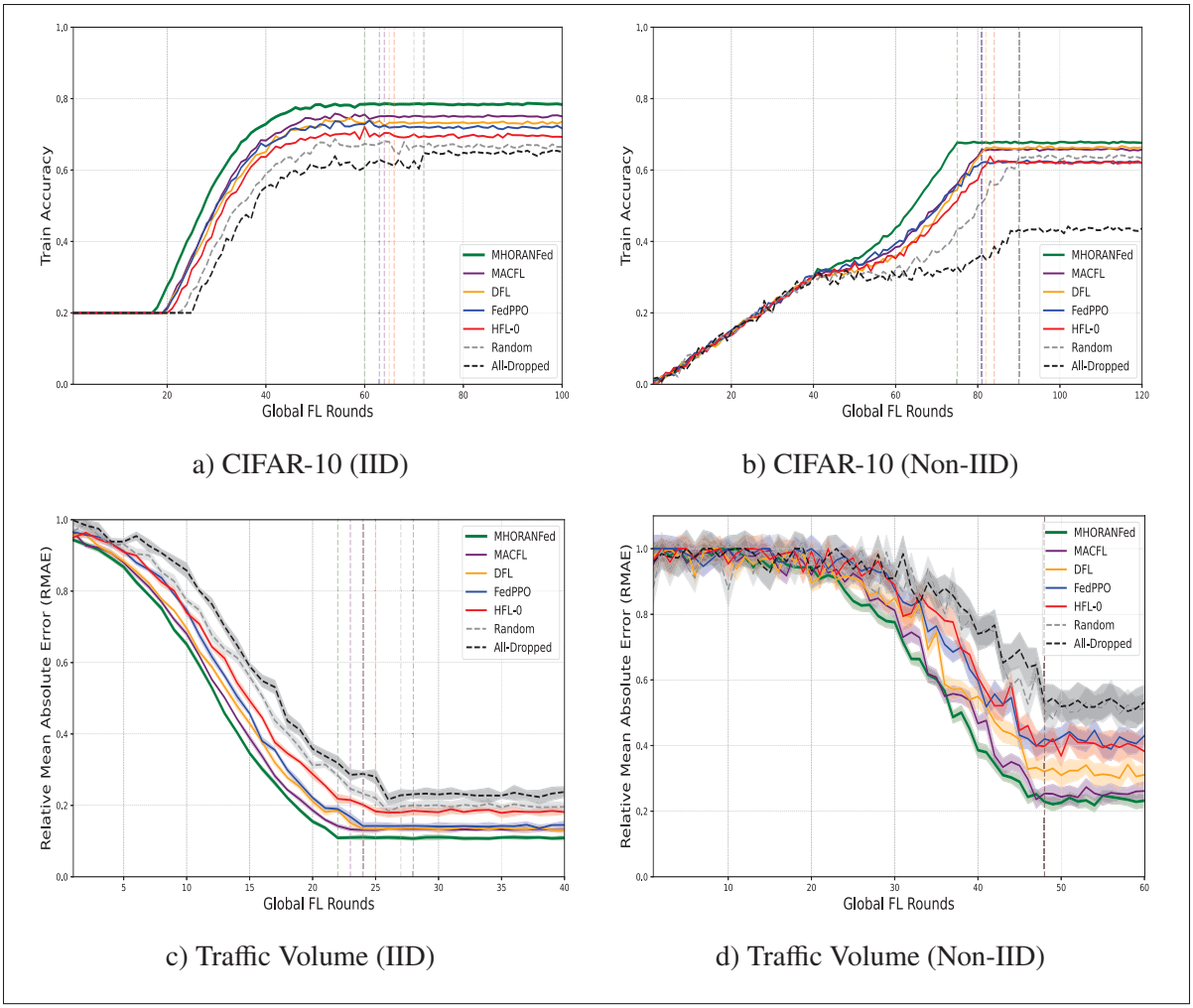


Figure 4.5 Comparison of convergence rate under IID [(a), (c)] and Non-IID [(b), (d)] data distributions for two learning tasks with convex [(c), (d)] and non-convex [(a), (b)] loss functions for the traffic prediction and the image classification respectively

4.5.4.2 Impact of Loss function type

The convergence in the non-convex case (CIFAR-10) appears to be more sensitive to the non-IID data distribution, as seen by the more pronounced differences in the convergence patterns between the IID and Non-IID CIFAR-10 plots in Figures 4.5a and 4.5b.

In contrast, the RMAE curves for the Traffic dataset show a generally smoother and more rapid decrease, particularly in the IID setting as can be seen in Figure 4.5d. MHORANFed effectively

handles the non-convex loss function in the CIFAR datasets, achieving high accuracy in both IID and Non-IID settings. However, like other methods, its convergence is somewhat affected by the non-IID distribution, showing a more gradual increase in accuracy. With the convex loss function in the Traffic datasets, Figure 4.5c validates that MHORANFed shows a rapid convergence and achieves the lowest RMAE.

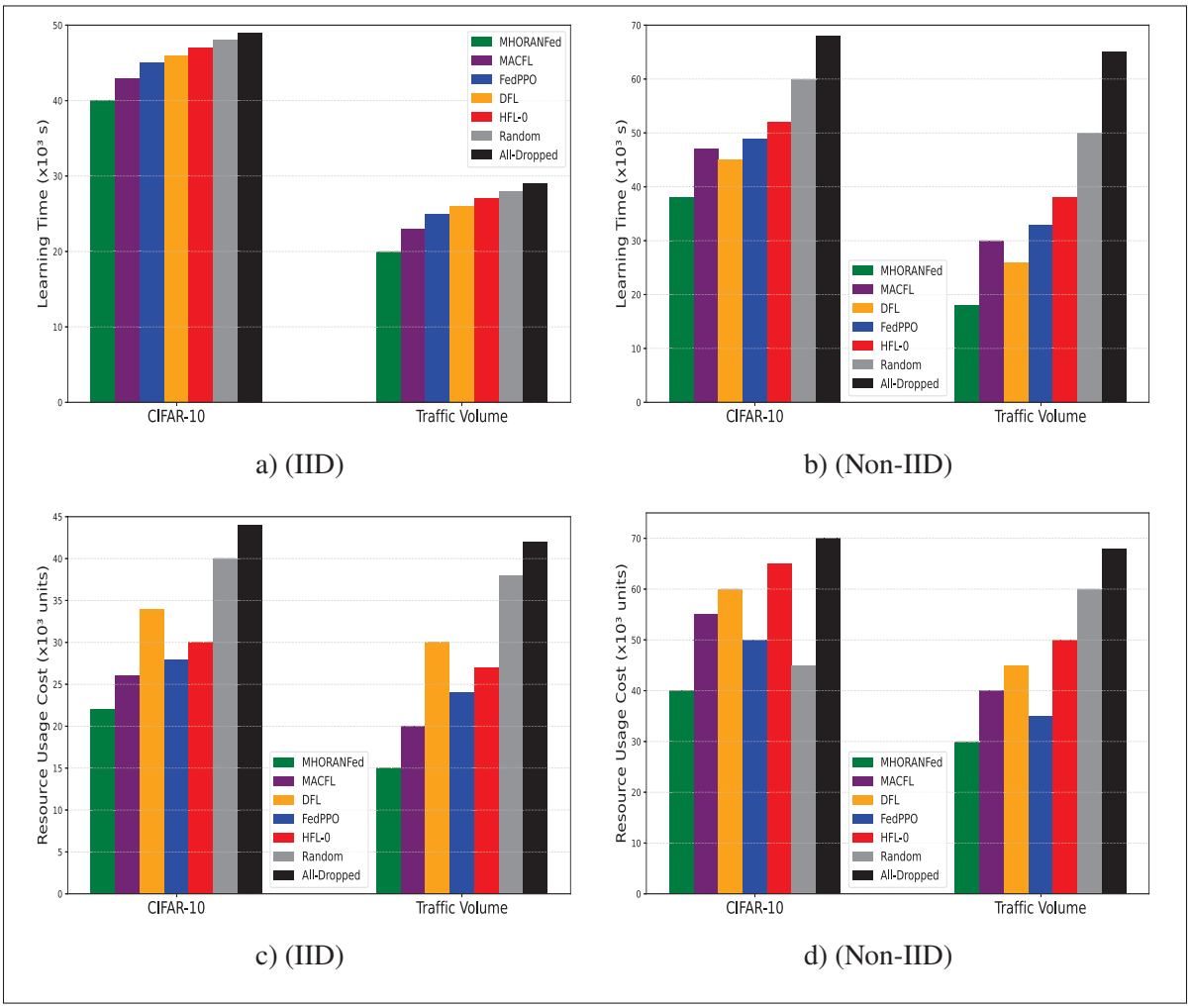


Figure 4.6 Comparison of Training Costs in terms of total Learning Time [(a), (b)] and Resource Usage [(c), (d)] under IID and Non-IID data distributions

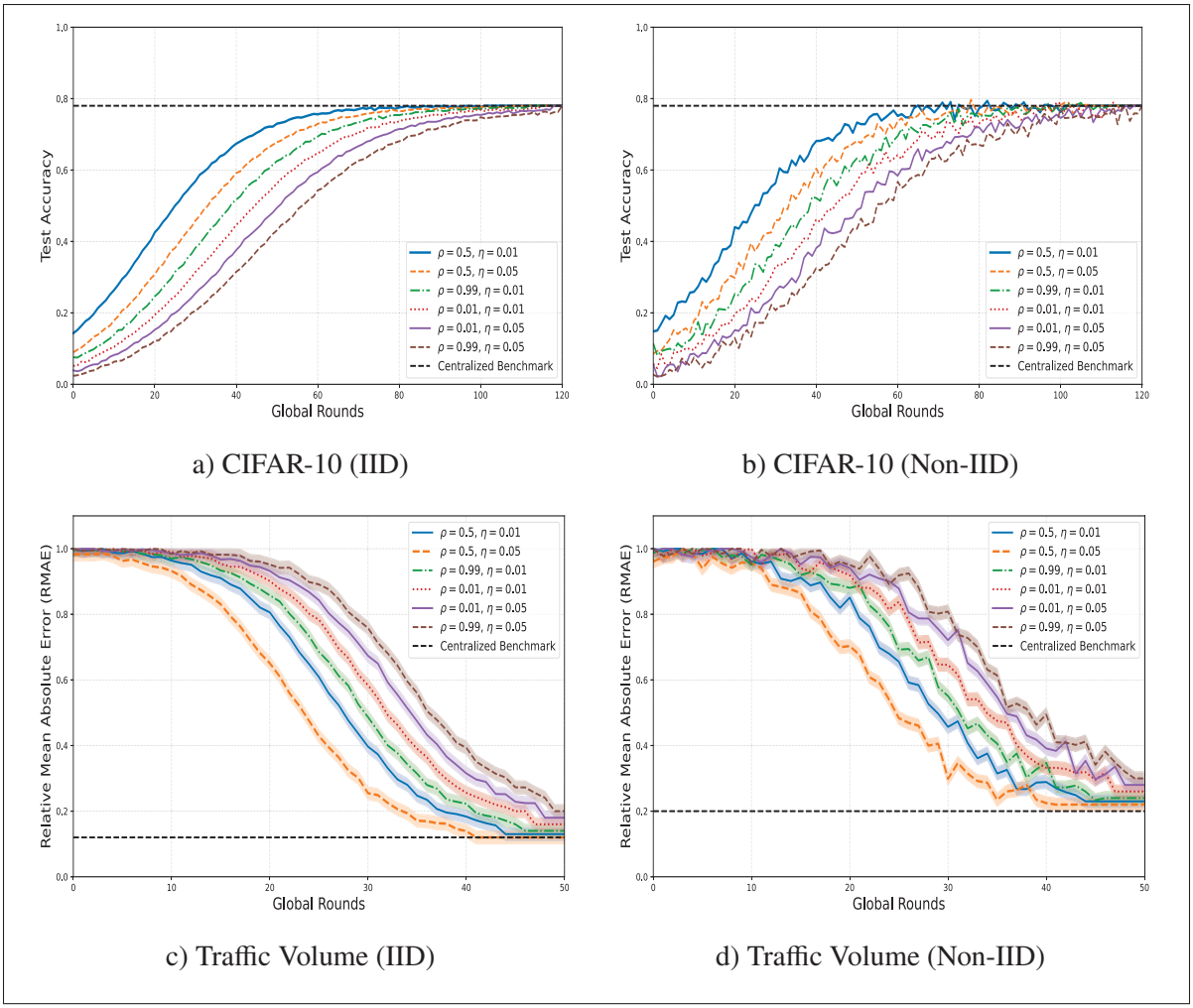


Figure 4.7 Performance of MHORANFed under different data distributions, learning tasks, and system hyper-parameters. (a) and (b) are with a Non-Convex Loss Function, while (c) and (d) are with a Convex Loss Function

4.5.4.3 Impact of data heterogeneity

The Non-IID settings significantly impact the convergence behavior and performance of all methods. As illustrated in Fig. 4.5, the results across the CIFAR and Traffic datasets reveal a consistent trend: data heterogeneity, as represented by the non-IID settings, significantly impacts the convergence and final performance of all federated learning methods compared to the IID scenarios. In both the CIFAR and Traffic tasks, the convergence curves for all methods are generally slower and sometimes more erratic under non-IID conditions. Furthermore, the

final performance metrics (Train Accuracy for CIFAR and RMAE for Traffic) achieved by the methods are typically worse in the non-IID settings than in their IID counterparts. Nonetheless, MHORANFed consistently outperforms other methods across both IID and Non-IID settings and for both CIFAR and Traffic datasets.

On the other hand, Fig. 4.6 shows that the presence of heterogeneous data distribution leads to higher training costs. While MHORANFed incurs significantly lower learning time and resource usage costs, other baselines show a relatively higher costs with fluctuating performance. This suggests that MHORANFed is more robust and effective in various hierarchical federated learning scenarios.

4.5.4.4 Impact of key hyperparameters setting (Sensitivity)

Fig. 4.7 shows a comparative behaviour of MHORANFed on how well it performs with respect to the centralized training benchmark and multiple combinations of ρ and η. We can see that varying the Pareto trade-off (ρ) affects convergence and higher values don't always guarantee the best outcome. On the other hand, lower values of ρ (e.g., 0.01) can lead to slower convergence. Another key observation is that the performance gap between MHORANFed and the centralized benchmark is often wider in Non-IID scenarios. We can infer through Fig. 4.7a, 4.7c, 4.7b, and 4.7d that the best performance corresponds to $\rho = 0.5$ and $\eta = 0.01$.

4.5.4.5 Impact of the number of UEs and BSs (Scalability)

We applied two scenarios corresponding to small (2 BSs and 20 UEs) and medium (5 BSs and 100 UEs) scale connections to bring out the impact of scalability. The result is shown in Fig. 4.8. While the methods converge in both the cases, they are affected by the presence of heterogeneity (non-IID). Nonetheless, MHORANFed requires less number of global rounds to attain the threshold model performance for both the learning tasks.

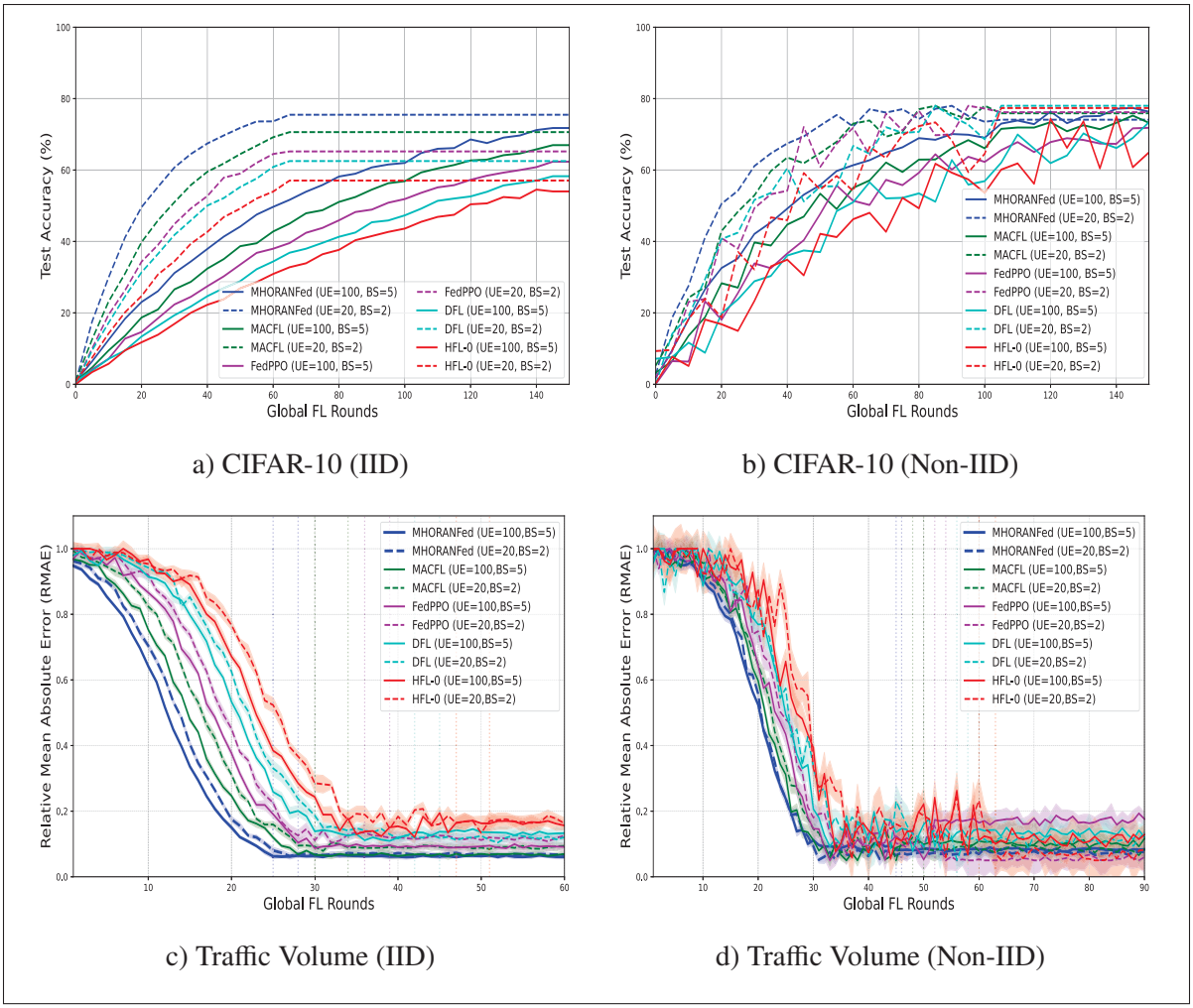


Figure 4.8 Comparison of scalability with two settings: (i) small scale (2 BSs and 20 UEs), (ii) medium scale (100 UEs and 5 BSs) under IID and Non-IID data distributions, convex and non-convex loss functions for the image classification and traffic prediction learning tasks respectively

4.5.5 Analysis

The superior performance of MHORANFed is due to its optimal selection of participating UEs in each global FL round and the corresponding bandwidth resource allocation to only the contributing UEs in the ongoing model training. MHORANFed converges faster because it is biased towards those temporally associated set of local trainers that require less processing time. So, the UEs with higher compute tends to be favoured over increasing global rounds. On the other

hand, the other baselines do not employ any such technique. *DFL* while reduces the training time in each global round by fixing a deadline for the local trainers, it also requires more number of global rounds to converge. This results in overall a higher learning time. In terms of the resource cost, MHORANFed critically assigns the bandwidth for the communication channel between a BS and its associated UEs for uploading the updated model parameters. Therefore, it minimizes the total resource usage cost. By outperforming clustering-based MACFL and RL-based FedPPO, MHORANFed's HO delay aware optimization based selection of local trainers proves to be more effective, highlighting the importance of this proposed approach. Despite the increased training costs associated with heterogeneous data for all methods, MHORANFed incurs significantly lower learning time and resource usage costs compared to the baselines.

MHORANFed effectively handles both non-convex (CIFAR) and convex (Traffic) loss functions, achieving high accuracy and rapid convergence, respectively. While its convergence is somewhat affected by non-IID distribution in the non-convex case, it still maintains superior performance. MHORANFed requires fewer global rounds to reach the performance threshold in both small and medium-scale network settings, indicating better scalability compared to other methods, especially in the presence of data heterogeneity. MHORANFed consistently achieves better final performance metrics (Train Accuracy for CIFAR and RMAE for Traffic) compared to other methods across both IID and Non-IID data distributions and for both the CIFAR and Traffic datasets.

Based on the evaluation outcome, we can infer that MHORANFed demonstrates superior performance and robustness in hierarchical federated learning scenarios, particularly under challenging conditions such as data heterogeneity. These results can be extrapolated to infer the generalization ability of our proposed method as it solves the optimization problem (4.21) in every global round to get an approximated near-optimal solution. So, it should work for real world use cases where the models are required to be executed with UEs' dynamically changing data. Moreover, since the model is trained on such large datasets, it also has the potential to be scaled over a large number of UEs with smaller dataset sizes. Applications such as fulfilling the

QoS of connected cars, the key network traffic indicators can be used to predict the required radio resource using MHORANFed exploiting its fast and reliable convergence property.

4.6 Conclusion

In this paper, we have introduced MHORANFed, a novel optimization algorithm tailored for HFL within the O-RAN architecture. By addressing the dynamic nature of mobile networks, particularly UE handovers, our proposed MHORANFed algorithm minimizes FL model training time and resource usage costs while maintaining high model accuracy. By addressing mobility-related challenges in FL for O-RAN, this research not only enhances the practicality and efficiency of HFL models but also lays the foundation for enabling a broad range of next-generation use cases. This work bridges a critical gap, pushing the boundaries of what intelligent, privacy-preserving, and resource-efficient networks can achieve in highly dynamic environments. These improvements are critical for enabling advanced 5G applications, such as autonomous driving and augmented reality, which demand both high performance and stringent privacy standards. By effectively managing the challenges posed by UE handovers and the dynamic sets of associated devices, MHORANFed paves the way for more flexible, intelligent, and efficient 5G network optimization. Future research will delve into further refining our algorithm and exploring its application in other evolving network paradigms, ensuring that the potential of FL in enhancing O-RAN architectures is fully realized.

Acknowledgment

The authors thank NSERC, VMware Canada, and Mitacs for supporting this research through under Grant ALLRP 577577-22.

CHAPTER 5

O-FL rApp: ORCHESTRATING xApps RESOURCES FOR MULTIPLE FEDERATED MARL TASKS IN O-RAN

Amardip Kumar Singh[1] , Kim Khoa Nguyen[1]

[1] Department of Electrical Engineering, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Paper submitted in *IEEE Transactions on Networking*, November 2025

Abstract The convergence of Open Radio Access Network (O-RAN) and Federated Multi-Agent Reinforcement Learning (MARL) promises intelligent, distributed network optimization, but introduces critical orchestration challenges when multiple learning tasks compete for limited compute and communication resources. This paper presents O-FL rApp, a novel orchestration framework hosted in the Non-Real-Time RAN Intelligent Controller (Non-RT-RIC) that manages the lifecycle of concurrent Federated MARL xApps. We formulate the orchestration problem as a multi-objective constrained optimization that jointly optimizes resource utilization, minimizes learning latency, and reduces Quality of Service (QoS) violations across heterogeneous network slices. To address the computational intractability of the resulting mixed-integer non-linear program (MINLP), we propose a tractable two-stage decomposition approach: (i) Task and Slice Assignment (TSA) using priority-based greedy heuristics, and (ii) Resource Allocation and Routing (RAR) via piecewise linear approximation and Mixed-Integer Linear Programming (MILP). We prove the existence of optimal solutions and establish geometric convergence guarantees for our iterative feedback-driven orchestration scheme. Through relevant case studies with eMBB and uRLLC xApps, we demonstrate that O-FL rApp achieves 98.2% service delivery success, 24% resource savings, and $33 - 50\%$ faster policy convergence compared to state-of-the-art baselines including auction-based, priority-based, and independent learning approaches. The framework scales efficiently to high system loads (90% utilization) while maintaining strict latency guarantees through strategic task preemption and adaptive resource reallocation.

Keywords:

5.1 Introduction

The exponential growth in mobile data traffic and the burgeoning demand for diverse services, from ultra-reliable low-latency communication (uRLLC) for industrial automation to enhanced mobile broadband (eMBB) for immersive experiences, necessitate a more intelligent and adaptable Radio Access Network (RAN) (Alam *et al.*, 2025). Traditional RAN architectures, characterized by monolithic and proprietary implementations, struggle to meet these evolving requirements. The Open RAN (O-RAN) initiative represents a paradigm shift, promoting open interfaces, virtualization, and disaggregation of RAN functionalities (Polese, Bonati, D'Oro, Basagni & Melodia, 2023b). This architectural evolution enables the deployment of intelligent applications, named *xApps*, within the Near-Real-Time RAN Intelligent Controller (Near-RT-RIC), facilitating granular, near-real-time optimization of network performance.

At the heart of advanced network optimization lies the potential of Machine Learning (ML), particularly Reinforcement Learning (RL) and its multi-agent extension (MARL) (Nasir & Guo, 2019). RL agents can learn optimal control policies through interaction with the environment, making them suitable for dynamic tasks such as task offloading (Zhan, Guo, Li & Zhang, 2022), interference mitigation and handover optimization Nguyen, Nguyen & Nahavandi (2020), and even resource management (Lu, Huang, Liu, Liu & Zhang, 2022), (Shi *et al.*, 2022). MARL extends this capability to multiple agents that coordinate within a shared environment, mirroring the distributed nature of RAN operation.

The choice of a MARL paradigm, as opposed to a centralized single-agent approach, is fundamental to addressing the challenges of the modern RAN (Wang *et al.*, 2024a). A centralized controller managing thousands of UEs would be intractable due to the curse of dimensionality in its action and state space. Furthermore, the MARL paradigm offers several compelling advantages: scalability, by distributing the control problem; low latency, by allowing local agents to react in real-time; reduced communication overhead, by acting on local observations; and privacy, by obviating the need to share sensitive local data (Xu, Wang & Chen, 2022b). However, deploying RL agents at the edge is challenging due to the need for substantial data

and computational resources for training. Federated Learning (FL) mitigates this by enabling agents to train local models and contribute to a shared global model without sharing raw data (Singh & Khoa Nguyen, 2025), (Lim *et al.*, 2020).

Foundational work on communication-efficient Federated MARL (Krouka, Elgabli, Issaid & Bennis, 2022) has established the feasibility of distributed learning, while various heuristic solutions like auction-based mechanisms (Cheng *et al.*, 2022a) and priority-based schedulers (Nguyen, Tran, Tun, Han & Hong, 2023a) have been proposed for managing multi-task coexistence. However, they fall short in addressing an orchestration, where multiple tasks compete for shared resource pools (O-DU compute, Near-RT-RIC compute, transport bandwidth) and the lifecycle of xApps (instantiation, termination, preemption) must be actively managed in response to changing resource availability.

The existing approaches such as (Xu *et al.*, 2022b),(Nguyen *et al.*, 2023a) often rely on static priorities or simplified resource models, failing to provide a systematic framework for jointly optimizing the tightly coupled compute, bandwidth, and QoS constraints of heterogeneous tasks throughout its lifecycle. When tasks enter or exit the system, the available resource pools change, creating resource contention that requires coordinated decision-making across all active and pending tasks rather than independent per-task allocation. For example, when a new high-priority uRLLC task arrives at a system where available Near-RT-RIC compute is exhausted, the orchestrator must identify which active lower-priority task to preempt to free resources, a decision that cannot be made by any single task in isolation and requires global visibility of all task states, priorities, and resource consumption.

Thus, the convergence of Federated MARL and O-RAN presents three primary orchestration challenges when multiple xApps are deployed concurrently: (i) Multiple FL MARL xApps simultaneously compete for shared resource pools; (ii) Tasks supporting different slices (e.g., uRLLC vs. eMBB) impose conflicting demands. uRLLC requires low-latency paths and guaranteed compute, while eMBB tolerates higher latency but demands high throughput. These conflicts require coordinated resolution as naive independent allocation can lead to QoS

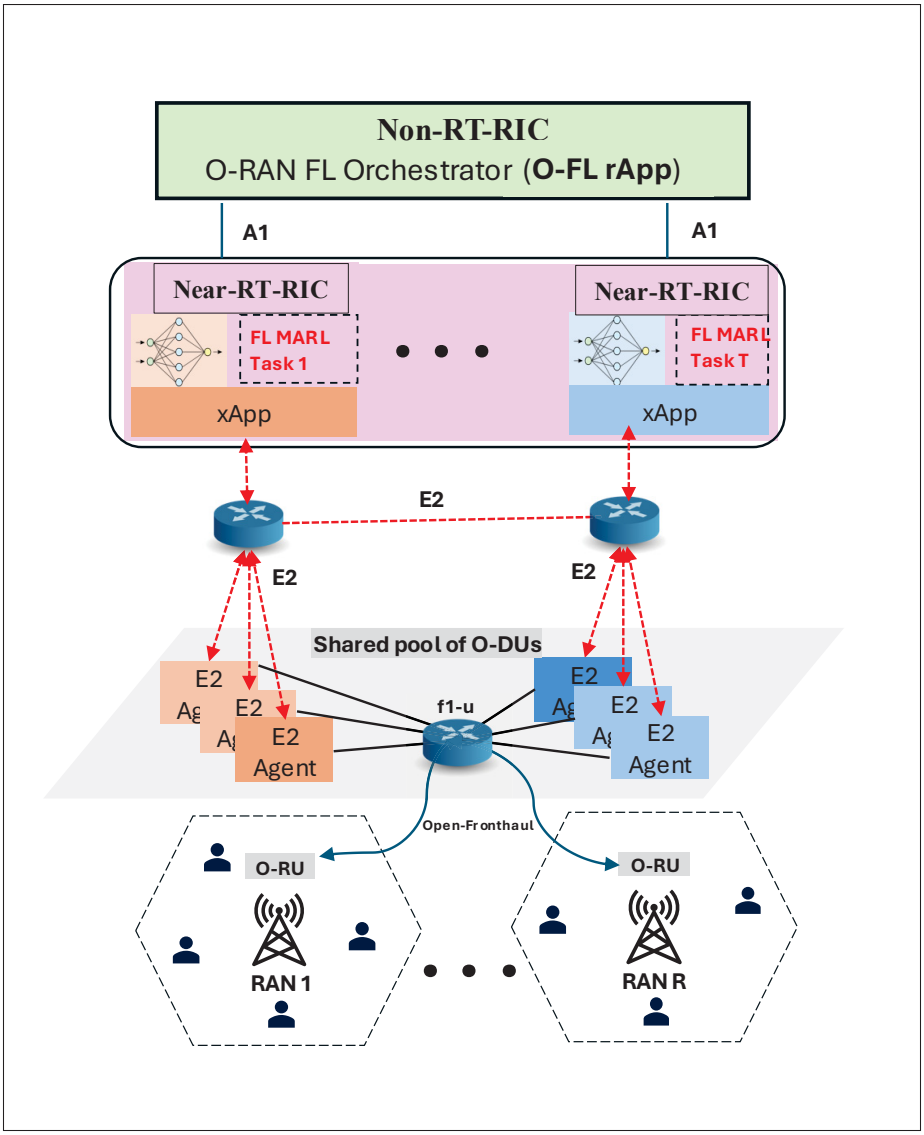


Figure 5.1 System architecture of Multiple Federated Learning xApps orchestrated by a FL Resource Manager rApp over O-RAN

violations; and (iii) Admitting a new task may require preempting an existing task to free resources. This creates temporal dependencies where admission decisions depend on the current state of all active tasks, and preemption of one task affects resource availability for all remaining tasks.

To address these challenges, this paper proposes an Orchestration Framework for Multiple Federated MARL Tasks in O-RAN. We introduce the O-FL rApp, hosted in the Non-RT-RIC, as a centralized orchestrator that manages the lifecycle of FL MARL xApps through joint optimization rather than independent per-task allocation. The O-FL rApp optimizes resource utilization (compute and communication), minimizes learning latency, and reduces QoS violations across network slices. We formalize this as a multi-objective constrained optimization and propose a tractable two-stage decomposition solution with convergence guarantees. The key contributions are:

Main Contributions

1. **O-FL rApp Orchestration Framework**: A novel orchestration framework at the Non-RT-RIC that coordinates multiple concurrent FL MARL xApps through joint optimization. The framework manages task lifecycles (admission, preemption, termination) in response to resource availability changes.
2. **Mathematical Problem Formulation**: A multi-objective constrained optimization model (Section 5.3.5) that explicitly couples all tasks through shared resource capacity constraints (5.20 - 5.22) for communication bandwidth, and Eq. (5.23 - 5.24) for the compute at Near-RT-RIC and O-DU layers.
3. **Two-Stage Decomposition Methodology**: A hierarchical solution (Section IV) with Task and Slice Assignment (TSA) for strategic decisions on task activation/preemption and Resource Allocation and Routing (RAR) for tactical resource scaling, aligned with O-RAN control loops.
4. **Convergence Analysis:** Rigorous mathematical analysis (Section IV-D) ensuring feasible solutions and geometric convergence to neighborhoods of the optimal solution.
5. **Case Study Validation**: A timeline-based case study (Section V-A) demonstrating how resource availability evolves as tasks enter and exit. The study shows orchestration events where a new high-priority task arrival triggers preemption of an active low-priority task to free resources. Explicit resource state tracking (Table III) shows compute and bandwidth

consumption before and after each event, illustrating why coordinated multi-task optimization is necessary.

The rest of the paper is organized as follows. Section 5.2 critically reviews related works. Section 5.3.5 develops the system model, including resource consumption equations, performance metrics, and the multi-objective optimization with shared capacity constraints coupling all tasks. Section 5.4 presents our two-stage decomposition solution and the event-driven orchestration loop that triggers re-optimization when tasks arrive, complete, or violate QoS. Section 5.5 provides numerical results, including a timeline-based case study showing resource state evolution across multiple orchestration events. Section 5.6 concludes the work with recommended future works.

5.2 Related Works

Our work is positioned at the intersection of three key domains: intelligent control in O-RAN, Federated Multi-Agent Reinforcement Learning (MARL), and multi-service orchestration.

Intelligence and Automation in O-RAN: The O-RAN architecture, with its disaggregated components and open interfaces, is fundamentally designed to host intelligent applications for network automation. The Near-RT RIC and Non-RT RIC are central to this vision, enabling data-driven control loops that operate at different timescales (Polese *et al.*, 2023b). Recent comprehensive surveys have detailed the O-RAN architecture, its deployment options, and its potential for enabling advanced use cases like network slicing and dynamic resource management (Alam *et al.*, 2025). Furthermore, the role of machine learning as the primary engine for this automation is well-established, with a growing body of work focused on applying ML techniques to optimize RAN performance, as highlighted by Hamdan et al. (Hamdan *et al.*, 2023b). These works establish the architectural foundation and the imperative for intelligent controllers, but often focus on the enablement of single, specific ML functions rather than the coordination of multiple, concurrent, and potentially conflicting ones.

Federated and Multi-Agent Learning in Mobile Networks: MARL has emerged as a powerful paradigm for solving complex, distributed optimization problems in wireless networks, such as

dynamic power allocation (Nasir & Guo, 2019) and interference coordination. The ability of multiple agents to learn collaborative policies makes MARL a natural fit for the distributed nature of the RAN (Nguyen *et al.*, 2020). To address the significant data privacy and communication overhead challenges of training these models at the network edge, Federated Learning (FL) has been integrated, leading to the paradigm of Federated MARL (Yu, Wang & Luan, 2023b). Krouka et al. Krouka *et al.* (2022) have specifically focused on enhancing the communication efficiency of Federated MARL, a critical concern in resource-constrained wireless environments. Our work builds upon this foundation, assuming the use of such advanced FL MARL tasks as the services to be managed. However, we shift the focus from optimizing the internals of a single FL MARL task to orchestrating a system where multiple such tasks must coexist.

Orchestration of Multiple Edge Services: The challenge of managing multiple competing services in edge computing and wireless networks is a prominent research area. Existing approaches largely rely on heuristic or market-based mechanisms. For instance, Cheng et al. Cheng *et al.* (2022a) proposed an auction-based trading framework to allocate resources for multiple FL services in UAV-aided networks based on a utility bidding mechanism. The authors in (Nguyen *et al.*, 2023a) and (Lu *et al.*, 2022) explored resource sharing through pre-defined policies or priorities which cannot adapt easily to handle changes in task requirements or network state. While these approaches address the multi-service coexistence problem, they often lack a holistic, formal optimization framework that jointly considers the tight coupling between task activation, slice assignment, routing, and the allocation of both compute and communication resources, which is especially critical for the performance of latency-sensitive FL MARL tasks in O-RAN.

Table 5.1 Summary of Mathematical Notation

Symbol	Description
$\mathcal{R}, \mathcal{S}, \mathcal{T}$	Sets of RANs, slices, and FL-MARL tasks (xApps)
$\mathcal{A}_t$	Set of RL agents for task $t \in \mathcal{T}$
$\mathcal{L}$	Set of communication links
B_l	Available bandwidth on link $l \in \mathcal{L}$ (Gbps)
$\rho_r^{\max}$	Maximum compute capacity at RAN node r (TOPS)
$O_{r,a}$	Binary: 1 if agent a is in RAN r, 0 otherwise
$M_a(t), M_{\text{xApp}}(t)$	Compute needs of agent a and xApp t (TOPS)
$\text{DT}_a(t)$	Data transfer requirement per agent per time period (Mbits)
$\text{QoS}_{r,s}^{\text{req}}, \text{QoS}_{r,s}^{\text{base}}$	Required and baseline QoS for slice s in RAN r
$\text{QoS}_{r,s}^{\text{act}}(t)$	Actual QoS achieved by task t (Eq. 9)
$\text{QoS}_{r,s}^{\text{vio}}(t)$	QoS violation penalty (Eq. 10)
$\omega_{r,s}^{\text{policy}}, \omega_{r,s}^{\text{res},i}$	Weights for policy and resource-i QoS contribution
$C_{\text{odu}}^{\text{comp}}, C_{\text{ric}}^{\text{comp}}$	Unit costs of O-DU and Near-RT-RIC compute ($/TOPS$)
$C^{\text{comm}}(l)$	Unit cost of bandwidth on link l ($/Gbps$)
Decision Variables	
$X_{t,a} \in \{0, 1\}$	Binary: 1 if task t handles agent a, 0 otherwise
$Y_{t,r,s} \in \{0, 1\}$	Binary: 1 if task t manages slice s in RAN r
$z_t \in \{0, 1\}$	Binary: 1 if task t is active, 0 otherwise
$R_{l,a} \in \{0, 1\}$	Binary: 1 if agent a uses link l, 0 otherwise
$u_{r,t} \in [0, 1]$	O-DU compute allocation ratio for task t in RAN r
$v_t \in [0, 1]$	Near-RT-RIC compute allocation ratio for task t
$f_{l,t} \geq 0$	Bandwidth allocation fraction for task t on link l
Derived Variables	
$\rho_{r,t}$	Effective O-DU compute for task t in RAN r (Eq. 1)
ν_t	Effective Near-RT-RIC compute for task t (Eq. 2)
$\phi_{l,t}, \phi_{\text{total},t}$	Bandwidth for task t on link l and total (Eq. 3-4)
$R_U(\rho_{r,t}), R_V(\nu_t), R_F(\phi_{l,t})$	Resource utility functions (Eq. 5-7)
k_U, k_V, k_F, k_P	Gain factors for utility functions

While the aforementioned research provides crucial building blocks, a significant gap remains in providing a systematic orchestration framework specifically for heterogeneous FL MARL xApps within O-RAN. Unlike prior work, our O-FL rApp framework allows for not only the optimal allocation of resources but also strategic, timely decisions on task activation and preemption, thereby ensuring end-to-end performance and guaranteeing QoS for high-priority services in a contested environment.

5.3 System Model and Problem Formulation

As illustrated in Fig. 5.1, our system architecture for concurrent Federated MARL training is organized hierarchically, comprising distributed RAN components, intelligent controllers, and a comprehensive orchestration layer. In Table 5.1, we summarize the mathematical symbols and notations used in describing this system model. The individual agents (E2 Agents) within each FL MARL task perform fine-grained, real-time control actions based on their local observations. These actions are task-dependent and directly influences the resource demands of the xApp. For instance:

- **eMBB Task:** Consider a cell-edge throughput maximization xApp. An agent's *action space* could be the selection of a power level and a Modulation and Coding Scheme (MCS). Its *local observation* would be its own Channel State Information (CSI) and perceived interference. The agents collaboratively learn a policy to maximize their data rates without causing undue interference to each other. This task is data-intensive but tolerant to minor latency variations.
- **uRLLC Task:** Consider a Vehicle-to-Everything (V2X) collision avoidance xApp. The local agents are autonomous vehicles. An agent's *action* is the selection of a specific resource block in a grant-free uplink schedule to transmit a safety message. Its *local observation* is channel occupancy. The agents must learn a decentralized scheduling policy that guarantees their message is successfully decoded by nearby vehicles within a strict latency budget of a few milliseconds. This task has a small data footprint but is extremely sensitive to both communication and compute delays.

The orchestrator must ensure that each group of agents (for each xApp) has the necessary network slice, compute, and transport resources to learn and execute its policy effectively.

5.3.1 Key Participating Nodes

The O-FL rApp orchestration framework operates across three primary architectural layers of the O-RAN ecosystem, each serving distinct roles in the federated learning process. Consistent with the O-RAN standards (Hamdan *et al.*, 2023b), intelligence and data processing are distributed between the O-DU layer and the RAN Intelligent Controllers (RICs), with coordinated resource management spanning from edge processing to policy-level orchestration.

O-DU: The O-DUs serve as the edge processing nodes where local RL agents reside and perform distributed learning. Each agent $a \in \mathcal{A}$ is associated with a specific RAN $r \in \mathcal{R}$ through the binary indicator $O_{r,a}$ and operates on local data derived from Key Performance Indicators (KPIs) of its assigned network slices $S_a \subseteq \mathcal{S}$. These agents train local models using private slice data and local reward signals, leveraging the distributed computational resources available at the network edge for fine-grained, real-time learning tasks.

Near-RT-RIC: This layer hosts the federated learning coordination through xApps. Each Federated MARL task $t \in \mathcal{T}$ is associated with a dedicated xApp that aggregates model updates from its assigned agent set $\mathcal{A}_t$. The xApps learn policies that optimize network performance for specific slices or functions and compute global rewards $R_{\text{global}}(t)$ that reflect the collective performance of their contributing agents and slices.

Non-RT-RIC: Operating at the regional cloud level, this component manages policy-level orchestration and service lifecycle functions. The O-FL rApp resides here and continuously monitors all active Federated MARL tasks, tracking their resource requirements and performance metrics. It makes strategic decisions regarding task-to-slice assignments through variables $Y_{t,r,s}$, agent assignments via $X_{t,a}$, and task activation states z_t, while coordinating resource allocation across RANs and managing task lifecycle events, as illustrated in Fig. 5.2.

Inter-Layer Communication: The federated learning process relies on a communication where agents at O-DUs exchange model parameters with xApps at Near-RT-RICs through diverse transport links $l \in \mathcal{L}$. Consistent with the E2 transport protocal defined in (O-RAN Software Community, 2025), each link l is characterized by its bandwidth capacity B_l, latency profile, and operational cost $C^{\text{comm}}(l)$. To avoid complexity, the link capacity have been assumed to be static and the path decision variable $R_{l,a} \in \{0, 1\}$ indicates whether agent a utilizes link l for model parameter transmission, enabling flexible path selection based on task requirements. This flexibility is essential for accommodating heterogeneous QoS demands, where uRLLC tasks require low-latency fiber links while eMBB tasks can tolerate higher-latency microwave backhaul.

5.3.2 Task Lifecycle and System State Model

We model the O-FL rApp orchestration system as a queuing network where FL MARL tasks transition through discrete states in response to orchestration events at each control loop period (e.g., $\Delta\tau = 100$ms).

Each task $t \in \mathcal{T}$ exists in one of two states at time τ: $\{ACTIVE, INACTIVE\}$. where ACTIVE indicates that the task is deployed and consuming resources, and INACTIVE indicates that the task is either Preempted (i.e., temporarily suspended), or Terminated (i.e., completed or removed) and it does not consume resources.

Varying Resource Availability: Available resources at time τ depend on the active task set:

$$\text{CPU}_{RIC}^{avail}(\tau) = \text{CPU}_{RIC}^{total} - \sum_{t:\mathcal{S}_t(\tau)=\text{ACTIVE}} \nu_t(\tau) \tag{5.1}$$

$$\text{CPU}_{DU,r}^{avail}(\tau) = \rho_r^{max} - \sum_{t:\mathcal{S}_t(\tau)=\text{ACTIVE}} \rho_{r,t}(\tau) \tag{5.2}$$

$$B_l^{avail}(\tau) = B_l - \sum_{t:\mathcal{S}_t(\tau)=\text{ACTIVE}} \phi_{l,t}(\tau) \tag{5.3}$$

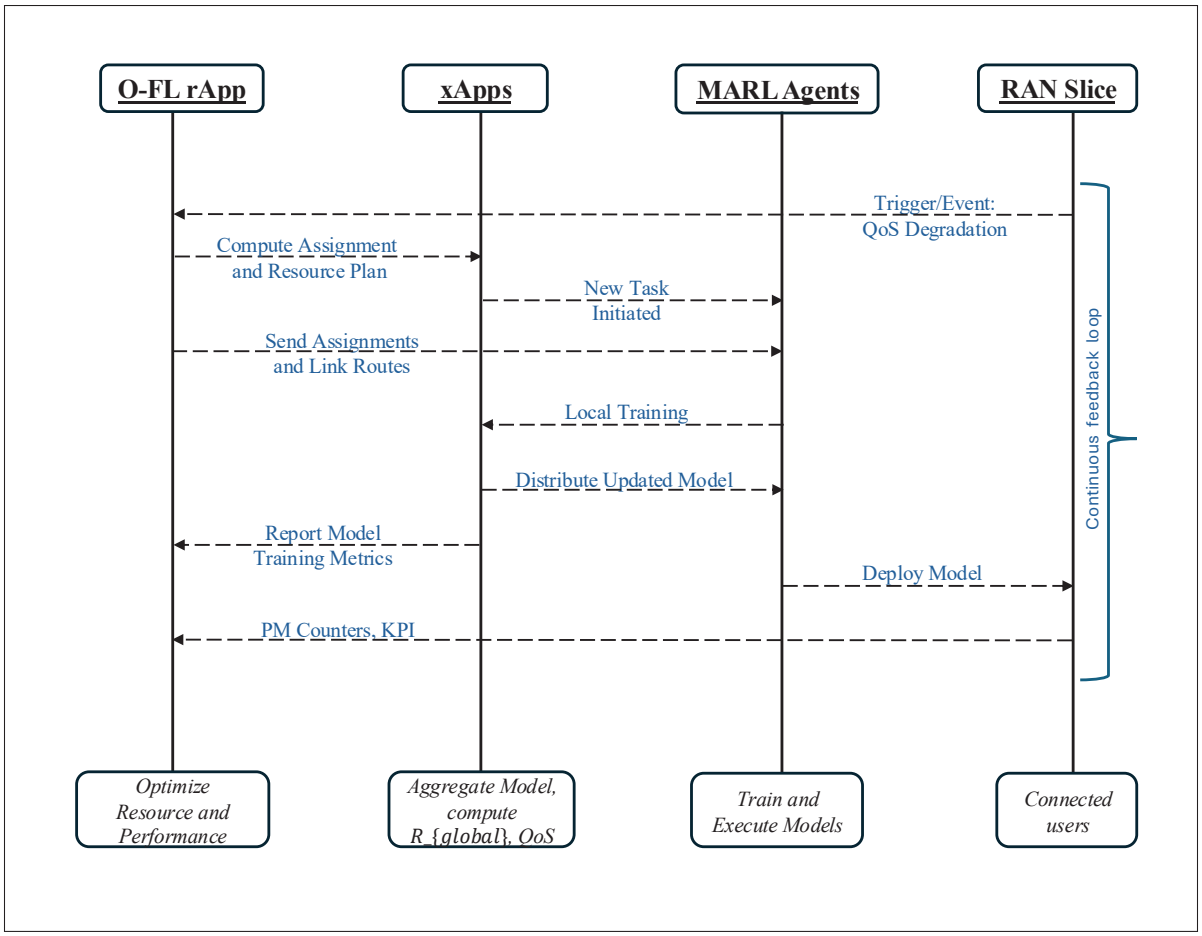


Figure 5.2 Task Sequence of the System Model. Initial deployment uses default resource allocations; subsequent iterations use optimized assignments from the O-FL rApp based on performance feedback

When task t' transitions INACTIVE $\rightarrow$ ACTIVE, available resources decrease by $(\rho_{r,t'}, \nu_{t'}, \phi_{l,t'})$ for all other tasks. On the other hand, transitions to Preempted or Terminated free resources. This state-dependent resource coupling necessitates joint optimization across all tasks rather than independent per-task decisions.

5.3.3 Association and Resource Consumption

Let the variable $Y_{t,r,s}$ denotes whether task t is managing slice s in RAN r. We denote agent a's assignment for task t by a binary valued variable: $X_{t,a}$. The extent of computational resources

allocated for the task t at Near-RT-RIC is v_t. The O-DUs share a pool of computational resources with total capacity $\text{CPU}_{\text{DU}}^{\text{total}}$ distributed across all gNBs. An xApp in RAN r consumes compute resources from this shared pool according to its allocation ratio $u_{r,t} \in (0, 1)$, a normalized fractional share of the total compute.

We calculate the effective O-DU Compute resource allocated for task t in RAN r as:

$$\rho_{r,t} = u_{r,t} \cdot M_a(t) \cdot \sum_{a \in A_t} (X_{t,a} \cdot O_{r,a}) \quad \forall r \in \mathcal{R}, t \in \mathcal{T}; \tag{5.4}$$

where this equation scales the compute requirement $M_a(t)$ is the compute requirements of agent a of task t, and $O_{r,a}$ is a binary variable determining whether the agent a has been assigned with the RAN r.

The Effective Near-RT-RIC Compute resource allocated for task t is calculated as follows:

$$v_t = v_t \cdot M_{xApp}(t) \quad \forall t \in \mathcal{T}, \tag{5.5}$$

in which v_t is task t's allocation ratio out of the multiple Near-RT-RIC instances' total available compute power and $M_{xApp}(t)$ is the compute requirement of the agent a for task t.

The bandwidth for task t on link l is:

$$\phi_{l,t} = f_{l,t} \cdot B_l \quad \forall l \in \mathcal{L}, t \in \mathcal{T} \tag{5.6}$$

This directly utilizes the decision variable $f_{l,t}$, which indicates the fractional bandwidth allocation.

Therefore, the total bandwidth for task t:

$$\phi_{total,t} = \sum_{l \in \mathcal{L}} \phi_{l,t} \quad \forall t \in \mathcal{T} \tag{5.7}$$

This equation aggregates the bandwidth allocated across all links for task t, representing its total communication provisioning.

5.3.4 Performance Functions

Motivated from the foundational work on resource utility modeling to preserve the proportional fairness (Kelly, Maulloo & Tan, 1998), we consider log functions to get a continuous function out of the discrete utility values. These functions model the effect of resource allocations on task performance, specifically in terms of QoS and global reward, featuring adjustable parameters for algorithm specificity. We model the resource utility as follows:

$$\mathcal{R}_U(\rho_{r,t}) = k_U \cdot \log\left(1 + \frac{\rho_{r,t}}{M_a(t)}\right); \quad \forall r \in \mathcal{R}, t \in \mathcal{T} \tag{5.8}$$

$$\mathcal{R}_V(\nu_t) = k_V \cdot \log\left(1 + \frac{\nu_t}{M_{xApp}(t)}\right); \quad \forall t \in \mathcal{T} \tag{5.9}$$

$$\mathcal{R}_F(\phi_{l,t}) = k_F \cdot \sqrt{\frac{\phi_{l,t}}{DT_a(t)}}; \quad \forall l \in L, t \in \mathcal{T} \tag{5.10}$$

Where, $DT_a(t)$ is the data transferred through the link $\phi_{l,t}$. Here, these functions reflects the diminishing returns of compute ($\mathcal{R}_U, \mathcal{R}_V$) and bandwidth ($\mathcal{R}_F$) on performance, incorporating tunable gains (k_U, k_V, k_F) and normalization factors.

The Policy Performance Metric ($P_{task}(t)$) is modeled as:

$$P_{task}(t) = k_P \cdot \mathcal{R}_V(\nu_t) \cdot \mathcal{R}_F(\phi_{total,t}); \quad \forall t \in \mathcal{T} \tag{5.11}$$

This metric evaluates the quality of the learned MARL policy, dependent on the xApp's compute utility ($\mathcal{R}_V$) and the total bandwidth utility ($\mathcal{R}_F$), scaled by a gain factor k_P.

The actual QoS is modeled as the baseline (obtained from the initial RL model) plus an improvement term, which is weighted by slice specific factors (ω) and integrates the policy's

contribution (P_{task}) with resource utilities ($\mathcal{R}$):

$$QoS^{act}_{r,s}(t) = QoS^{base}_{r,s} + Y_{t,r,s} \cdot (\omega^{policy}_{r,s} \cdot P_{task}(t) + \omega^{res,U}_{r,s}\mathcal{R}_U(\rho_{r,t}) + \omega^{res,V}_{r,s}\mathcal{R}_V(\nu_t) + \omega^{res,F}_{r,s}\mathcal{R}_F(\phi_{total,t})); \quad \forall r \in \mathcal{R}, s \in \mathcal{S}, t \in \mathcal{T} \tag{5.12}$$

Here, the assignment indicator $Y_{t,r,s}$ ensures the model's applicability only to assigned slices.

The QoS violation is modeled as:

$$QoS^{vio}_{r,s}(t) = \max(0, QoS^{req}_{r,s} - QoS^{act}_{r,s}(t)) \quad \forall r \in \mathcal{R}, s \in \mathcal{S}, t \in \mathcal{T} \tag{5.13}$$

This metric quantifies the penalty for failing to meet the required QoS, defined as the non-negative difference between the required and achieved QoS.

The global reward for task t is modeled as:

$$R_{global}(t) = \lambda_{QoS} \cdot \sum_{r \in R} \sum_{s \in S} Y_{t,r,s} \cdot \Delta QoS_{r,s}(t) - \lambda_{cost} \cdot \text{TotalCost}(t) \quad \forall t \in \mathcal{T} \tag{5.14}$$

This global reward function balances the QoS improvements (ΔQoS) achieved across assigned slices against the total resource cost incurred, utilizing tunable weights λ_{QoS} and λ_{cost}. Here, $\Delta QoS_{r,s}(t)$ represents the improvement component of the actual QoS outcome $QoS^{act}_{r,s}(t)$.

Hence, the total resource cost is:

$$\text{TotalCost}(t) = \sum_{r \in R} \sum_{a \in \mathcal{A}_t} C_{comp\_odu} \cdot \rho_{r,t} + C_{comp\_ric} \cdot \nu_t + \sum_{l \in \mathcal{L}} C_{comm}(l) \cdot \phi_{l,t} \quad \forall t \in \mathcal{T} \tag{5.15}$$

It calculates the total operational cost for task t, summing the costs associated with effective compute resources at O-DUs (ρ), Near-RT-RIC (ν), and allocated bandwidth (ϕ) across relevant components.

5.3.5 Main Optimization Problem Formulation

We frame the orchestration of multiple Federated MARL tasks as a constrained optimization problem. The objective is to determine the optimal assignment of tasks, allocation of resources, and routing decisions that minimize resource costs and QoS violations while simultaneously maximizing overall MARL performance. The objectives of the O-FL rApp are multi-faceted: (i) **Minimize Resource Costs:** Reduce expenditures on computational resources (O-DU and Near-RT-RIC) and communication bandwidth; (ii) **Minimize Learning Latency:** Accelerate the convergence of each Federated MARL task, which seem to be inversely related to the computational resources allocated to the xApp; and (iii) **Minimize QoS Violations:** Ensure that the actual Quality of Service (QoS^{act}) for each slice meets the required Quality of Service (QoS^{req}). QoS violations (QoS^{vio}) are penalized. The performance of each task t ($Perf_t^{act}$) is directly assessed using its calculated global reward, $R_{global}(t)$.

The orchestrator's objective balances multiple competing goals through a weighted sum formulation. The first term aggregates the compute costs associated with agents and xApps, the second term accounts for the bandwidth costs related to communication. So, the first two terms minimize operational expenditure (OPEX) for compute and communication resources. The third term incentivizes better learning with improved global rewards (R_{global}) weighted by W_{reward} for active tasks ($z_t = 1$). The fourth term penalizes QoS violations with weight W_{QoS}, affecting assigned slices ($Y_{t,r,s} = 1$) to ensure strict QoS guarantees take precedence over cost minimization. This formulation enables the orchestrator to make principled tradeoffs, such as preempting low-priority tasks when necessary to avoid catastrophic violations for mission-critical services.

$$\begin{aligned}\text{Minimize } & \sum_{t\in T}\left(\sum_{r\in R}\sum_{a\in A_t} C_{comp\_odu}\cdot\rho_{r,t} + C_{comp\_ric}\cdot\nu_t\right)\\ & + \sum_{l\in L}\sum_{t\in T} C_{comm}(l)\cdot\phi_{l,t}\\ & - W_{reward}\sum_{t\in T} R_{global}(t)\cdot z_t\\ & + W_{QoS}\sum_{r\in R}\sum_{s\in S}\sum_{t\in T} QoS^{vio}_{r,s}(t)\cdot Y_{t,r,s}\end{aligned} \tag{5.16}$$

The optimization process is subject to the following constraints:

1. **Task and Agent Association Constraints:**

$$\sum_{t\in\mathcal{T}} X_{t,a} \le 1 \quad \forall a \in \mathcal{A} \tag{5.17}$$

$$X_{t,a} \le O_{r,a} \quad \forall t \in \mathcal{T}, a \in \mathcal{A}_t, r \in \mathcal{R} \tag{5.18}$$

$$\sum_{a\in\mathcal{A}_t} X_{t,a} \ge z_t \quad \forall t \in \mathcal{T} \tag{5.19}$$

These constraints ensure that agents are assigned uniquely (5.17) and only to RANs where they are physically located (5.18), with tasks being activated only when at least one agent has been assigned (5.19).

2. **Routing and Bandwidth Constraints:**

$$\sum_{a\in A_t} \frac{DT_a(t)}{\text{time\_period}} \cdot O_{r,a}\cdot X_{t,a}\cdot\mathcal{R}_{l,a} \le \phi_{l,t};$$

$$\forall l \in \mathcal{L}, t \in \mathcal{T}, r \in \mathcal{R}; \tag{5.20}$$

$$\phi_{l,t} \le B_l \quad \forall l \in \mathcal{L}, t \in T; \tag{5.21}$$

$$\sum_{l\in\mathcal{L}} R_{l,a} = \sum_{t\in\mathcal{T}} X_{t,a} \quad \forall a \in \mathcal{A}; \tag{5.22}$$

These constraints regulate bandwidth allocation to ensure sufficient capacity for data transfers ($\phi_{l,t}$), respect link capacities (B_l), and enforce single-path routing per agent ($R_{l,a}$).

Constraint (5.21) ensures per-link capacity is respected, which implicitly bounds total bandwidth usage across all links, while 5.22 ensures that each assigned agent ($\sum_t X_{t,a} = 1$) uses exactly one routing path.

3. **Compute Resource Constraints:**

$$\sum_{r \in \mathcal{R}} \rho_{r,t} \leq CPU^{DU}_{total} \qquad \forall t \in \mathcal{T} \tag{5.23}$$

$$\sum_{t \in \mathcal{T}} \nu_t \leq CPU^{RIC}_{total} \tag{5.24}$$

These constraints cap the total effective compute usage in the O-DU pool and Near-RT-RIC to their respective maximum capacities.

4. **Performance Requirement Constraint:**

$$R_{global}(t) \cdot z_t \geq Perf_t^{req} \cdot z_t \qquad \forall t \in \mathcal{T} \tag{5.25}$$

This condition guarantees that tasks which are activated meet a minimum performance threshold, as indicated by their global reward.

5. **Variable Type Constraints:**

$$X_{t,a}, O_{r,a}, \mathcal{R}_{l,a}, z_t, Y_{t,r,s} \in \{0, 1\}$$

$$u_{r,t}, v_t \in [0, 1]$$

$$f_{l,t} \geq 0$$

5.4 Proposed Solution

The formulated optimization problem is a mixed-integer non-linear program (MINLP) with complex, non-convex utility functions (R_U, R_V, R_F) and coupled constraints across tasks. To achieve tractability while respecting O-RAN's architectural principles, we decompose the problem hierarchically to mirror the natural timescales of network control: *strategic decisions* (which tasks to activate, which slices to manage) operate on the Non-RT-RIC timescale (minutes),

while *tactical decisions* (resource allocation, routing) operate on shorter cycles (seconds). This decomposition is not merely a computational convenience, rather it reflects the separation of concerns between policy-level orchestration and resource-level optimization inherent in the O-RAN architecture. The TSA stage makes discrete assignment decisions that remain stable over multiple control cycles, while the RAR stage continuously adapts resource allocations within those assignments based on real-time feedback.

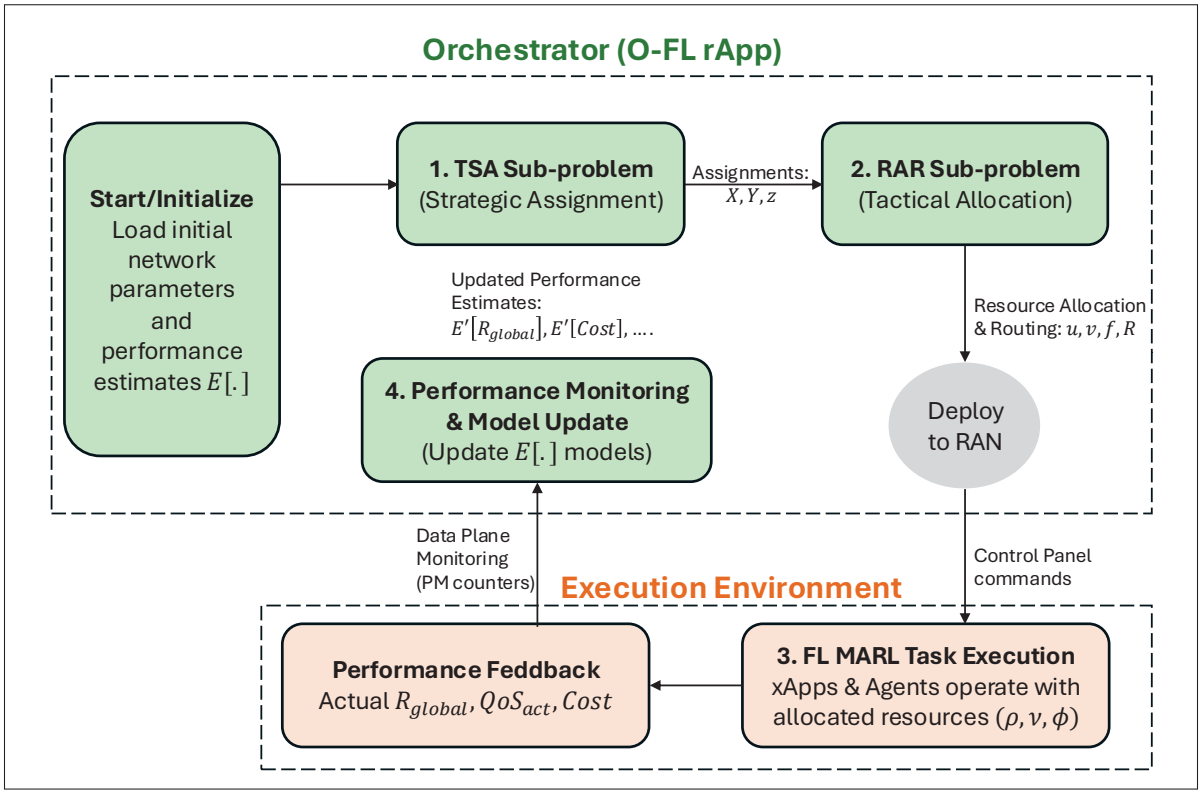


Figure 5.3 Proposed Iterative Solution Scheme for FL MARL Orchestration Problem

As presented through the solution schema in Fig. 5.3, the two stages are executed by exploiting the the control loops of O-RAN to form a continuous performance feedback in the following manner:

1. **Stage 1 (TSA):** Solve the Task and Slice Assignment problem to determine $X_{t,a}, Y_{t,r,s}, z_t$.
2. **Stage 2 (RAR):** Given assignments, solve the Resource Allocation and Routing problem to determine $u_{r,t}, v_t, f_{l,t}, R_{l,a}$. This involves executing the iterative optimization process within Stage 2.

3. **Performance Evaluation:** Execute MARL tasks using the allocated resources and collect performance metrics (QoS^{act}, R_{global}) and costs.
4. **Feedback and Re-optimization:** Use collected performance data to refine resource allocations in Stage 2 for the next iteration. Periodically re-run Stage 1 (TSA) if performance deviates significantly or if strategic reassignment is beneficial.

5.4.1 Sub-Problem (i): Task and Slice Assignment (TSA)

Its objective is to minimize the sum of expected assignment costs and expected overall performance penalties (costs, QoS violations) based on projected resource utilization.

$$\begin{aligned}\min_{X_{t,a},Y_{t,r,s},z_t} \sum_{t\in T}\Big(\sum_{a\in A_t} \text{Assignment\_Cost}(X_{t,a}) + \\ E[\text{TotalCost}(t)] - W_{reward}E[R_{global}(t)]\Big) \\ -W_{QoS}\sum_{r\in R}\sum_{s\in S}\sum_{t\in T} Y_{t,r,s}\cdot E[QoS^{vio}_{r,s}(t)]\end{aligned} \tag{5.26}$$

Subject to: constraints (5.17), (5.18), and (5.19).

Where $E[\cdot]$ denotes the expected values based on estimated resource demands and performance functions.

Solution: The TSA sub-problem is inherently a combinatorial optimization problem due to the binary assignment of agents to tasks and slices. It can be solved by a mathematical solver. However, when there are n tasks and m agents, the number of possible assignment configurations grows exponentially: $O(2^{nm})$. Therefore we design a Greedy heuristic to obtain approximated solution based on the task priorities. The priority score of an xApp's MARL task represents its maximum potential net benefit which is estimated as: $E[R_{global}(t)] - E[\text{TotalCost}(t)] - W_{QoS}\cdot\sum_{r,s}E[QoS^{vio}_{r,s}(t)]$ for assigning task t. So, its priority score is defined as:

$$\text{PriorityScore}(t) = \max_{a \in A_t, (r,s) \in R \times S} (\text{AssignmentScore}(t, a, r, s)) \tag{5.27}$$

where the Assignment Score is defined as:

$$\begin{aligned} \text{AScore}(t, a, r, s) =& E[R_{global}(t) | X_{t,a} = 1, Y_{t,r,s} = 1] \\ & - E[\text{TotalCost}(t) | X_{t,a} = 1] \\ & - W_{QoS} E[QoS^{vio}_{r,s}(t) | Y_{t,r,s} = 1] \end{aligned} \tag{5.28}$$

We utilise these scores for the iterative agent-slice assignment for the prioritized tasks, as laid out in Algorithm 5.1.

Algorithm 5.1 Greedy Task and Slice Assignment (TSA)

Input: Task set $\mathcal{T}$, Agent set $\mathcal{A}$, RANs $\mathcal{R}$, Slices S, Performance estimates $E[\cdot]$
Output: Assignments $X_{t,a}, Y_{t,r,s}, z_t$
Initialize X, Y, z to zeros; $AgentAvailable[a] = \text{true}\ \forall a$
Step 1: Task Prioritization
for *each task* $t \in \mathcal{T}$ **do**
    Compute priority score via (5.27)
end for
Sort $\mathcal{T} \rightarrow \mathcal{T}_{sorted}$ by descending priority
Step 2: Iterative Assignment
for *each task* t *in* $\mathcal{T}_{sorted}$ **do**
    Find best agent-slice pair:
    $(a^*, r^*, s^*) = \arg\max_{a \in \mathcal{A}_t, (r,s)} \text{AScore}(t, a, r, s)$
    subject to $AgentAvailable[a] = \text{true}$
    $best\_score = \text{AScore}(t, a^*, r^*, s^*)$
    if $best\_score > 0$ **then**
        $X_{t,a^*} \leftarrow 1, Y_{t,r^*,s^*} \leftarrow 1, z_t \leftarrow 1$
        $AgentAvailable[a^*] \leftarrow \text{false}$
    end if
end for
Return: $X_{t,a}, Y_{t,r,s}, z_t$

5.4.2 Sub-Problem (ii): Resource Allocation and Routing (RAR)

Given the fixed assignments $(X_{t,a}, Y_{t,r,s}, z_t)$ from Stage 1, this stage optimizes the allocation of compute $(u_{r,t}, v_t)$ and communication $(f_{l,t}, R_{l,a})$ resources for the current time window. Its objective is to minimize the total resource costs and QoS violation penalties, while maximizing the achieved global reward.

$$\begin{aligned}\min_{u_{r,t},v_t,f_{l,t},R_{l,a}} & \sum_{t\in T}\left(\sum_{r\in R}\sum_{a\in A_t} C_{comp\_odu}\cdot\rho_{r,t} + C_{comp\_ric}\cdot v_t\right) \\ & + \sum_{l\in L}\sum_{t\in T} C_{comm}(l)\cdot\phi_{l,t} \\ & - W_{reward}\sum_{t\in T} R_{global}(t)\cdot z_t \\ & + W_{QoS}\sum_{r\in R}\sum_{s\in S}\sum_{t\in T} QoS^{vio}_{r,s}(t)\cdot Y_{t,r,s} \end{aligned} \tag{5.29}$$

Subject to: constraints (5.20), (5.21), (5.22), (5.23), (5.24), and (5.25).

Solution: In order to tackle it through tractable solvers for MILP, we first linearize the non-linear performance functions and relaxing the integer constraints on routing variables. We use Piecewise Linear (PWL) approximation for each non-linear utility function $(\mathcal{R}_U, \mathcal{R}_V, \mathcal{R}_F)$ by defining N breakpoints $(v^{(j)}, \mathcal{R}_V(v^{(j)}))$ for $j = 1, \ldots, N$. The non-linear function is replaced by a new variable $R_{V,\text{approx}}(v_t)$ and the following set of linear constraints using convex combination weights $\alpha_{t,j} \geq 0$:

$\mathrm{R}_{V,\text{approx}}(v_t) = \sum_{j=1}^{N} \alpha_{t,j} R_V(v^{(j)})$

$\sum_{j=1}^{N} \alpha_{t,j} = 1$

This structure is replicated for $\mathcal{R}_U(\rho_{r,t})$ and $\mathcal{R}_F(\phi_{l,t})$.

Next, the performance metric $P_{task}(t) = k_P \cdot \mathcal{R}_V(\nu_t) \cdot \mathcal{R}_F(\phi_{total,t})$ involves a product of two (now linearized) variables, which is a non-linear term. To maintain a MILP, this product must also be linearized. For this, we introduced binary variables to select segments of the PWL functions and then used envelope to linearize the product of variables within those segments. The final performance metrics, $R_{global,approx}$ and QoS^{vio}_{approx}, are then constructed as linear combinations of these fully linearized components. Therefore, the decision variables are relaxed to: $u_{r,t}, v_t \in [0, 1]$, $f_{l,t} \geq 0$, and $R_{l,a} \in 0, 1$. The objective function is transformed to:

$$\begin{aligned} \min \Bigg[& \sum_{t \in T} \left(\sum_{r \in R} C_{\text{comp\_odu}} \rho_{r,t} + C_{\text{comp\_ric}} \nu_t \right) \\ & + \sum_{l,t} C_{\text{comm}}(l) \phi_{l,t} - W_{\text{reward}} \sum_{t \in T} R_{\text{global,approx}}(t) z_t \\ & + W_{\text{QoS}} \sum_{r,s,t} \text{QoS}^{\text{vio}}_{\text{approx},r,s}(t) Y_{t,r,s} \Bigg] \\ & \text{subject to:} \quad (5.20 - 5.25). \end{aligned} \tag{5.30}$$

We solve this reformed MILP using a mathematical solver (e.g., Gurobi) where it employs Branch and Cut algorithm and explore the tree of possible integer solutions while using LP relaxations at each node to prune sub-optimal branches. It continues until it finds a solution that is proven to be optimal. Thus, we obtain the optimal values for all decision variables: $u^*_{r,t}, v^*_t, f^*_{l,t}$, and $R^*_{l,a}$.

5.4.3 Orchestrator (O-FL rApp) Framework

The O-FL rApp operates as a dynamic and continuous control loop within the Non-RT RIC. The core orchestration logic, detailed in Fig. 5.3, is not a one-time process but is triggered by specific events that alter the state of the network or the requirements of the services. The primary triggers for invoking a new orchestration cycle are:

Algorithm 5.2 Orchestrated FL MARL Execution

Input: Assignments $(X_{t,a}, Y_{t,r,s}, z_t)$ from TSA; Resources $(u^*_{r,t}, v^*_t, f^*_{l,t}, R^*_{l,a})$ from RAR; local slice data
Output: Global models $\{w_t\}$; Metrics $\{R_{\text{global}}(t), \text{QoS}^{\text{act}}_{r,s}(t), \text{QoS}^{\text{vio}}_{r,s}(t), \text{Cost}(t)\}$
Triggered after TSA and RAR completion
for *each active task t with* $z_t = 1$ **do**
    for *each agent a with* $X_{t,a} = 1$ **do**
        Collect local slice data (KPIs, traffic stats)
        Train local MARL model $\pi_{t,a}$ on assigned slice s
        Compute local update $\Delta w_{t,a}$
        Transmit $\Delta w_{t,a}$ to xApp t via $(R_{l,a}, f_{l,t})$
    end for
    Aggregate at xApp t:
    $w_t \leftarrow \text{Aggregate}(\{\Delta w_{t,a} \mid a \in A_t\})$
    Evaluate metrics:
        $R_{\text{global}}(t)$ via (5.14)
        $\text{QoS}^{\text{act}}_{r,s}(t)$ via (5.12)
        $\text{QoS}^{\text{vio}}_{r,s}(t)$ via (5.13)
        $\text{Cost}(t)$ via (5.15)
    Redistribute w_t to all agents in A_t
end for
Forward outputs to Monitoring & Feedback stage

- **Service Lifecycle Events:** The arrival of a new request to instantiate an FL MARL xApp, or the termination of an existing xApp.
- **Significant QoS Degradation:** When the performance monitoring loop detects that an active, high-priority task (e.g., a uRLLC xApp) is failing to meet its QoS targets (e.g., latency, reliability).
- **Network Topology Changes:** A significant change in the underlying transport network, such as the failure or recovery of a backhaul link, which alters the cost and feasibility of routing options.

Upon one of these triggers, the O-FL rApp executes the two-stage optimization process described above to compute a new, globally optimal configuration for all active and pending tasks. This

ensures the system continuously adapts to changing conditions, making the orchestration inherently dynamic.

Algorithm 5.2 is employed for the continual federated training. It coordinates distributed model training where multiple agents collect local data from their assigned network slices, train local MARL models, and send their model updates to a central coordinator (xApp) using allocated communication and computing resources. The xApp aggregates these updates into a global model, which is then redistributed back to all participating agents. Throughout this process, the algorithm evaluates key performance metrics including global rewards, actual quality of service, QoS violations, and resource costs. The entire process is triggered after task and resource allocation stages complete, and outputs are forwarded to a monitoring system for feedback and system optimization.

5.4.4 Convergence Analysis

Now, we establish the theoretical foundations for the proposed two-stage orchestration framework. We first prove the existence and boundedness of optimal solutions to the original MINLP formulation, then analyze the convergence properties of our iterative decomposition approach with performance feedback.

Theorem 1 (Existence and Boundedness of Solutions): *Under the following conditions:*

1. *The feasible region* Θ *defined by constraints (5.17)-(5.25) is non-empty and compact (closed and bounded) in* $\mathbb{R}^n$*, and*
2. *The following functions are continuous over the feasible region* Θ*:*
   - *Resource utility functions* $R_U(\rho_{r,t})$*,* $R_V(\nu_t)$*,* $R_F(\phi_{l,t})$ *defined in (5.8)-(5.10),*
   - *Policy performance metric* $P_{task}(t)$ *in (5.11),*
   - *QoS violation function* $QoS^{vio}_{r,s}(t)$ *in (5.13);*

the optimization problem (5.16) admits at least one global optimal solution $\theta^* \in \Theta$*, and the optimal objective value* $J^* = J(\theta^*)$ *is finite.*

Proof Intuition: We establish existence and boundedness by showing two key properties. First, the feasible region Θ is compact as the binary assignment variables form a finite set, while continuous allocation ratios are bounded by definition, and bandwidth is physically limited. Second, the objective function J is continuous, it comprises finite sums and products of logarithmic and square-root functions applied to bounded arguments. By the Weierstrass Extreme Value Theorem, a continuous function on a compact set must attain its minimum, guaranteeing the existence of an optimal solution with finite objective value.

Proof: *Please refer to the Appendix II.*

Our analysis extends techniques from federated optimization theory (Li *et al.*, 2020b) to the multi-task orchestration context and assure its convergence through the following theorem.

Theorem 2: (Fixed-Point Convergence of Iterative Orchestration) *Considering the iterative orchestration framework where at each iteration k:*

1. *TSA (Stage 1) selects assignments $(X_{t,a}^{(k)}, Y_{t,r,s}^{(k)}, z_t^{(k)})$ based on estimates $\mathcal{E}^{(k)}$ using Algorithm 1,*
2. *RAR (Stage 2) optimizes resources $(u_{r,t}^{(k)}, v_t^{(k)}, f_{l,t}^{(k)}, R_{l,a}^{(k)})$ given fixed assignments using N-point PWL approximation,*
3. *FL MARL tasks execute with allocated resources and produce measurements $\mathcal{M}^{(k)}$,*
4. *Estimates update via $\mathcal{E}^{(k+1)} = \mathcal{T}(\mathcal{E}^{(k)}, \mathcal{M}^{(k)})$*

Under the following conditions:

- *The estimate update mapping $\mathcal{T}$ satisfies the contraction property with $\beta = 1 - \lambda < 1$, achieved through exponential moving average updates as employed in federated optimization for heterogeneous networks (Li* et al.*, 2020b),*
- *The measurement noise is uniformly bounded: $\|\mathcal{M}^{(k)} - \mathcal{E}^*\| \leq \sigma$ for all $k \geq 0$,*
- T*he TSA assignments are stable under small estimate perturbations: there exists $\epsilon > 0$ such that if $\|\mathcal{E}_1 - \mathcal{E}_2\| < \epsilon$, then Algorithm 1 produces identical assignments,*

- *The objective function J is L-Lipschitz continuous in the resource variables for fixed assignments (Bertsekas, 1999):*

$$|J(\xi, \psi_1) - J(\xi, \psi_2)| \leq L\|\psi_1 - \psi_2\|$$

for any fixed assignment ξ and resource allocations ψ_1, ψ_2,

the following hold:

1. **(Unique Fixed Point)** *There exists a unique fixed-point estimate $\mathcal{E}^* \in \mathbb{R}^n$ such that $\mathcal{T}(\mathcal{E}^*, \mathcal{E}^*) = \mathcal{E}^*$,*
2. **(Geometric Convergence of Estimates)** *The estimates converge geometrically to a neighborhood of the fixed point:*

$$\|\mathcal{E}^{(k)} - \mathcal{E}^*\| \leq \beta^k \|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\sigma}{1-\beta}$$

3. **(Assignment Stabilization)** *There exists a finite iteration $K < \infty$ such that the assignments remain constant for all $k \geq K$:*

$$(X_{t,a}^{(k)}, Y_{t,r,s}^{(k)}, z_t^{(k)}) = (X_{t,a}^{(K)}, Y_{t,r,s}^{(K)}, z_t^{(K)}) \quad \forall k \geq K$$

4. **(Objective Convergence)** *The objective value converges geometrically to a neighborhood of the optimal:*

$$J(\theta^{(k)}) - J^* \leq \Delta_0 \beta^k + C_\infty$$

where $\Delta_0 = L\|\mathcal{E}^{(0)} - \mathcal{E}^*\|$ and $C_\infty = \frac{L\sigma}{1-\beta} + \epsilon_N$ represents the steady-state approximation error, with $\epsilon_N = O(N^{-2})$

Proof: *Please refer to the Appendix II.*

5.4.5 Computational Complexity

Under the conditions of Theorem 2, to achieve objective value $J(\theta^{(k)}) \leq J^* + \epsilon + C_\infty$, the total number of iterations required is:

$$K_\epsilon = O\left(\frac{\log(\Delta_0/\epsilon)}{\log(1/\beta)}\right)$$

Each iteration requires solving the TSA problem in $O(|\mathcal{T}| \cdot |\mathcal{A}| \log |\mathcal{T}|)$ time (greedy sorting) and the RAR MILP in time polynomial in the problem size (Wolsey, 2020).

5.5 Numerical Results

In this section, we first describe the simulation parameters settings and present an example of how our proposed solution works by considering a case-study. Then, we compare its performance with state-of-the-art baselines.

5.5.1 Simulation Settings

We model a dense urban deployment with two O-RAN gNodeBs (‘gNB-A‘, ‘gNB-B‘). The system is equipped with a high-performance fiber link and a standard microwave backhaul link. The orchestrator operates on a 100ms control loop (the time window for resource allocation decisions), while the uRLLC task requires individual message latencies to remain below 6ms. Data transfers DT_a represent the cumulative data volume transferred by each agent during each 100ms control period. The system parameters in Table 5.2 are chosen to reflect realistic O-RAN deployments. The CPU capacities (100 TOPS for RIC, 80 TOPS per DU) correspond to commercially available Network Processing Units (NPUs) such as Qualcomm Cloud AI 100. The fiber link (10 Gbps, 0.5ms latency) represents metropolitan-area dark fiber, while the microwave link (1 Gbps, 5ms latency) models typical last-mile wireless backhaul. Cost ratios ($C_{\text{fiber}}/C_{\text{microwave}} = 5$) reflect the higher operational expenses of fiber leasing. The objective weights ($W_{\text{QoS}} = 1000 \cdot W_{\text{reward}}$) ensure that even a 1% QoS violation outweighs the reward from perfect task performance, enforcing hard QoS guarantees.

Table 5.2 System Model Specifications

Parameter	Value
RANs (R)	gNB-A, gNB-B
Slices (S)	s-eMBB, s-URLLC
Transport Links	L_F: Fiber (0.5ms), L_{MW}: MW (5ms)
Agents (A)	a_1, a_2 (T1); a_3, a_4 (T2)
CPU RIC	100 TOPS (NPU)
CPU DU	80 TOPS each
Bandwidth	BW_{L_F}: 10 Gbps, $BW_{L_{MW}}$: 1 Gbps
Comm. Cost	L_F: 10 u/Gbps, L_{MW}: 2 u/Gbps
Comp. Cost	C_{ric}: 0.5 u/TOPS, C_{du}: 0.4 u/TOPS
Weights	W_{reward}: 1.0, W_{QoS}: 1000
Time Window	100 ms
uRLLC Budget	$\leq$ 6 ms

5.5.2 Case Study – Orchestration of two xApps

This case study demonstrates the proposed decomposition solution in an orchestration of two distinct MARL tasks based on actor-critic methods.

- **Task 1 (T1): eMBB Throughput Maximization.** An MA2C xApp coordinating cell-edge UEs. Its goal is to maximize data throughput.
  - Data Transfer (DT_{a_1}, DT_{a_2}): 500Mbits per agent per time window.
  - Global Reward: $R_{global}(T1) = 200 \cdot \log(1 + \phi_{total,T1})$, where $\phi_{total,T1}$ is the total allocated bandwidth in Gbps, reflecting diminishing returns on bandwidth.
  - QoS Violations: This task has best-effort QoS constraints; $QoS^{vio}(T1) = 0$.
- **Task 2 (T2): uRLLC V2X Coordination.** An MA2C xApp managing Vehicle-to-Everything communication, requiring extremely low latency.
  - Data Transfer (DT_{a_3}, DT_{a_4}): 10Mbits per agent per time window.
  - Latency Budget: Total end-to-end latency must be $\leq 6ms$.

- Latency Model: $L_{total} = L_{comm} + L_{comp}$.
  - L_{comm} = Link Latency + $\frac{DT}{\phi_{total}}$.
  - $L_{comp} = \frac{\text{Required Ops}}{\nu_t} + \frac{\text{Required Ops}}{\rho_t}$. (Assuming 1 TOPS required at RIC and DU).
- QoS Violations: $QoS^{vio}(T2) = \max(0, L_{total} - 6$, in which the latency budget is 6ms), a linear penalty for every millisecond of delay beyond the budget.
- Global Reward: $R_{global}(T2) = 50$ (fixed reward for successful operation).

Iteration 1 (Initial Deployment) The orchestrator's objective is to deploy both T1 and T2 optimally. In stage 1: Greedy TSA, the orchestrator calculates the best-case 'PriorityScore' for each task as follows:

- **Score(T2):** The highest priority is to avoid the massive QoS penalty.
  - *Best Case (Fiber Link):* Latency is easily met. $L_{comm} \approx 0.5ms$. Score $\approx 50 -$ HighCost $- 0 \approx$ Slightly Negative.
  - *Worst Case (Microwave Link):* Base latency is 5ms. Even with infinite CPU/BW, it's impossible to meet the 6ms budget. A huge penalty is guaranteed. Score $\approx 50 -$ LowCost $- 1000 \cdot$ (delay) $\approx$ Very Negative.

  The greedy algorithm, aiming to maximize the objective (minimize penalties), sees that T2 must be assigned in a way that avoids the penalty. This gives it top priority.
- **Score(T1):** The task is profitable as long as it gets some bandwidth. Its score is positive.

TSA Decision: T2 is prioritized, followed by T1. Both tasks can be accommodated.

In stage 2: MILP RAR, the MILP solver receives the fixed assignments and computes the optimal resource allocation to minimize the objective function. To avoid the $W_{QoS} \cdot QoS^{vio}$ penalty for T2, the solver is forced to make the following decisions for T2:

- It must select the fiber link: $R_{L_F,a_3} = 1$.
- It must allocate sufficient CPU. Let's say it allocates $\nu_{T2} = 20TOPS$ and $\rho_{\text{gNB-B},T2} = 20TOPS$, making compute latency negligible.

For T1, the solver uses the remaining resources to maximize $R_{global}(T1)$ while minimizing cost. It allocates the entire cheaper microwave link and the remaining CPU.

Iteration 2 (Dynamic Adaptation): A high-priority V2X emergency service (Task 3) is requested, requiring uRLLC performance on ‘gNB-A‘. T3 has the same profile as T2.

In stage 1: Greedy TSA (Re-evaluation), the orchestrator re-evaluates. There is now a resource conflict on the fiber link, as both T2 and T3 need it, and their combined data rate would exceed capacity.

- **Scores(T2, T3):** Both have high priority to avoid QoS penalties.
- **Score(T1):** The orchestrator foresees that activating T1 would leave insufficient CPU/BW for T2 and T3, leading to massive QoS penalties. The projected system-wide objective value if T1 is activated is dominated by these penalties, making ‘PriorityScore(T1)‘ strongly negative.

TSA Decision: Deactivate T1 to guarantee service for the critical tasks. We obtain exact solution as: $z_{T1} = 0, z_{T2} = 1, z_{T3} = 1$ (Assignments for T2, T3 on gNB-B, gNB-A).

In stage 2: MILP RAR (Re-allocation), the solver now only allocates resources for T2 and T3. As shown in Table 5.3, in Iteration 1, T2 gets what it needs (20TOPS), T1 gets the rest of RIC CPU (80TOPS). Each task gets DU CPU on its assigned gNB (60TOPS and 20TOPS). T2 requires fiber, while T1 is assigned the cheaper microwave link. Each task gets required bandwidth on its link. In Iteration 2, both critical tasks get sufficient RIC CPU (20TOPS each) and DU CPU (20TOPS each) on their respective gNBs. Both tasks are routed over the fiber link, with bandwidth shared; total usage is well below capacity.

Table 5.3 TSA and RAR Solutions Across Iterations

Iter.	Variable	Value
TSA Solution - Iteration 1		
1	$z_{T1} = 1$, $z_{T2} = 1$	Both tasks activated.
1	$X_{T1,a_1} = 1$, $X_{T2,a_3} = 1$	One agent per task.
1	$Y_{T1,\text{gNB-A},s\text{-eMBB}} = 1$	T1: gNB-A, eMBB slice.
1	$Y_{T2,\text{gNB-B},s\text{-URLLC}} = 1$	T2: gNB-B, URLLC slice.
RAR Solution - Iteration 1		
1	v_{T1}, v_{T2}	80TOPS, 20TOPS
1	$\rho_{\text{gNB-A},T1}$, $\rho_{\text{gNB-B},T2}$	60TOPS, 20TOPS
1	$R_{L_F,a_3} = 1$, $R_{L_{MW},a_1} = 1$	Fiber & MW assigned
1	$\phi_{L_F,T2}$, $\phi_{L_{MW},T1}$	0.1Gbps, 1Gbps
RAR Solution - Iteration 2		
2	v_{T2}, v_{T3}	20TOPS, 20TOPS
2	$\rho_{\text{gNB-B},T2}$, $\rho_{\text{gNB-A},T3}$	20TOPS, 20TOPS
2	$R_{L_F,a_3} = 1$, $R_{L_F,a_4} = 1$	Both use fiber
2	$\phi_{L_F,T2}$, $\phi_{L_F,T3}$	0.1Gbps, 0.1Gbps

5.5.3 Baselines

We benchmark the proposed O-FL rApp orchestration framework against the following state-of-the-art baselines:

- **Independent FL (No Orchestration):** Each MARL xApp trains and allocates resources independently, without coordination from the Non-RT-RIC. Serves as a lower-bound baseline, showing what happens if orchestration is absent (resource contention, QoS violations).
- **Static Resource Partitioning:** Compute and bandwidth are equally divided among all active tasks, regardless of their requirements or priorities. it represents traditional “fixed-slice” (Nguyen *et al.*, 2023a) approach in O-RAN.

- **Auction-based Orchestration** (Cheng *et al.*, 2022a): Tasks bid for resources based on their utility, and allocations follow a utility-maximizing auction scheme. Then an auction game is applied for orchestrating multiple FL services.
- **Priority-based Scheduling** (Hamdan *et al.*, 2023b): Resources are allocated greedily according to fixed slice priorities (uRLLC > eMBB > mMTC), which is commonly used in practical 5G RAN scheduling, but does not adapt well to dynamic MARL learning needs.

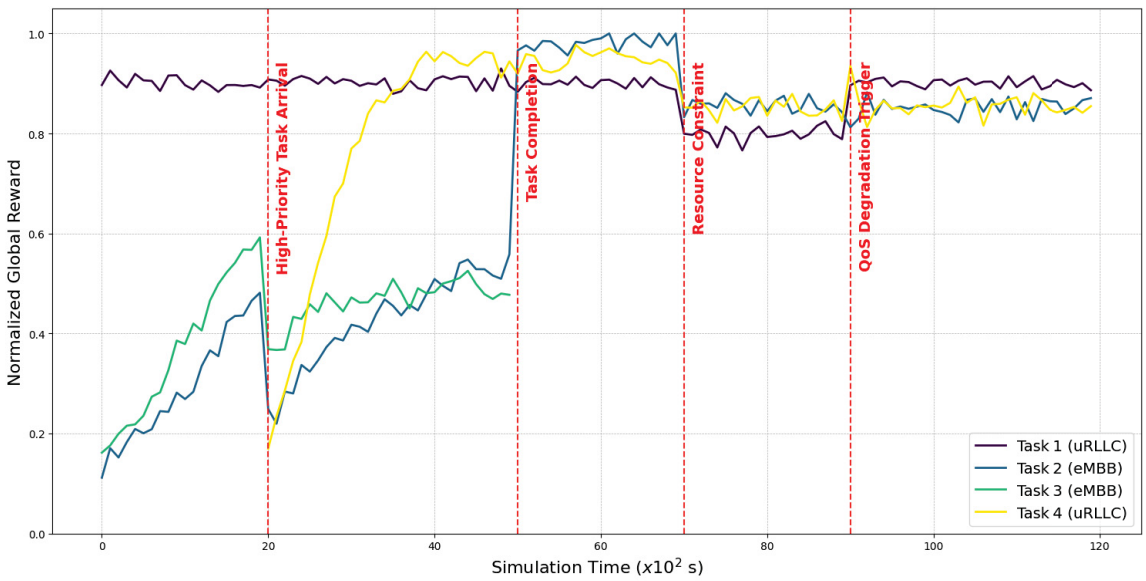
5.5.4 Performance Metrics

In addition to Global Reward, QoS Violation, and Resource Cost, we introduce two new evaluation metrics to compare with the baselines:

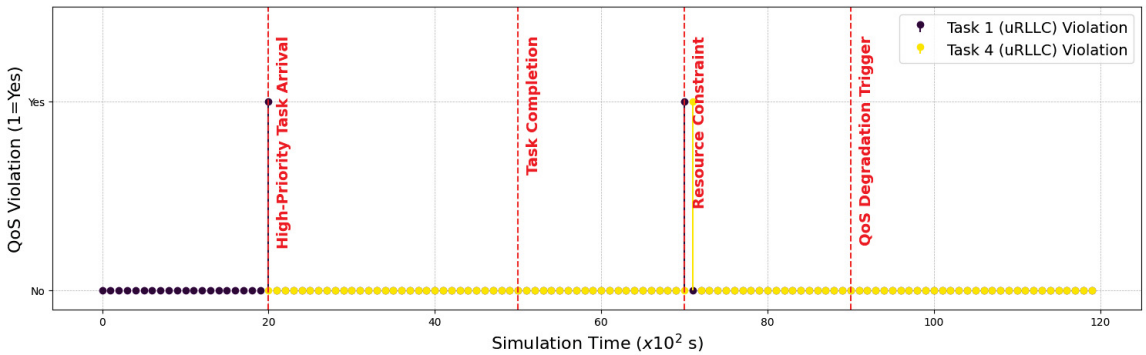
- **Learning Latency:** Time required for each task to converge to a stable policy.
- **Resource Utilization Efficiency:** Ratio of achieved performance (QoS satisfaction, global reward) to the total resources consumed.

5.5.5 Analysis

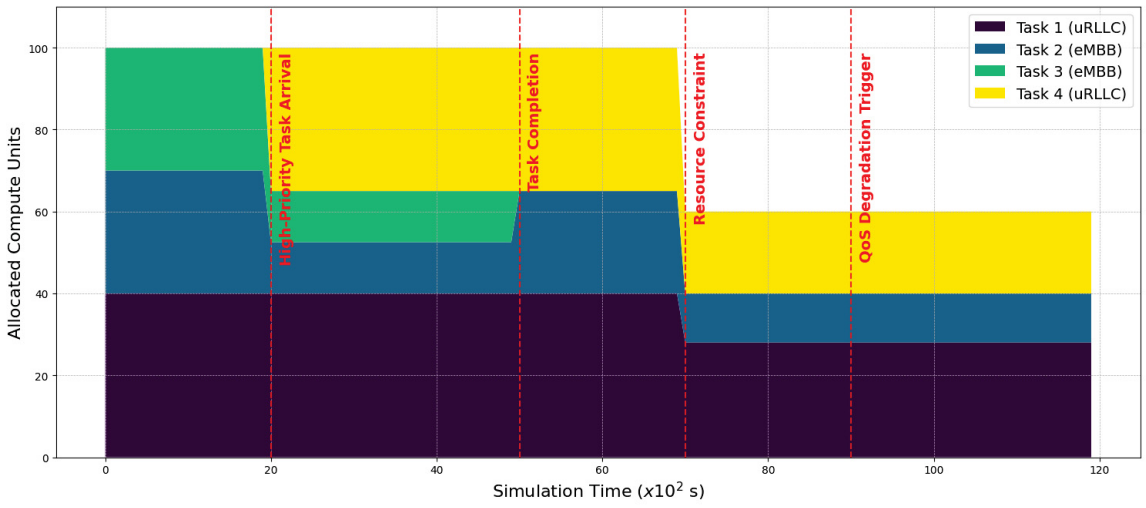
The case study demonstrates the orchestrator's ability to make flexible, system-wide decisions in response to changing network conditions. In Iteration 1, when both T1 (eMBB) and T2 (uRLLC) are deployed, the TSA algorithm correctly prioritizes T2 according to its stringent QoS requirements and high violation penalties. The RAR stage then optimally allocates the fiber link to T2 to meet its 6ms latency budget, while assigning T1 to the microwave link where its more relaxed requirements can still be satisfied at lower cost. The critical test occurs in Iteration 2, when a new high-priority uRLLC task (T3) arrives. The orchestrator's re-evaluation reveals a fundamental resource conflict: both T2 and T3 require the fiber link to meet their latency constraints, but T1's continued operation would consume resources needed to guarantee service for both critical tasks. However, maintaining T1 would result in massive QoS penalties for the uRLLC services, yielding a strongly negative priority score. Therefore, the orchestrator preemptively deactivates T1, demonstrating its capability for strategic task preemption—a key differentiator from static allocation schemes.



a) Task Performance (Global Reward) Timeline



b) QoS Violation for High-Priority Tasks Timeline



c) Resource Allocation Breakdown Timeline

Figure 5.4 System Adaptation under Real-Time Events

The timeline analysis reveals stark differences in how competing approaches handle various triggering events. Fig. 5.4a shows that O-FL rApp maintains stable global rewards for both tasks throughout the observation period, including during the critical task arrival event at t=500ms. In contrast, the Independent FL baseline exhibits severe performance degradation when T3 arrives, as tasks compete for resources without coordination. Static Partitioning shows moderate performance but fails to achieve the same reward levels as O-FL rApp due to inefficient resource usage. Fig. 5.4b demonstrates O-FL rApp's superior QoS management. While Priority-based Scheduling experiences brief violation spikes during transitions, O-FL rApp keeps violations near zero even during task preemption, validating the effectiveness of the two-stage decomposition. The Auction-based method shows periodic violations, suggesting that market mechanisms alone cannot guarantee strict latency requirements. Fig. 5.4c illustrates how O-FL rApp optimally reallocates compute and bandwidth resources in response to T3's arrival, increasing uRLLC resource allocations while reducing eMBB allocations to maintain system stability.

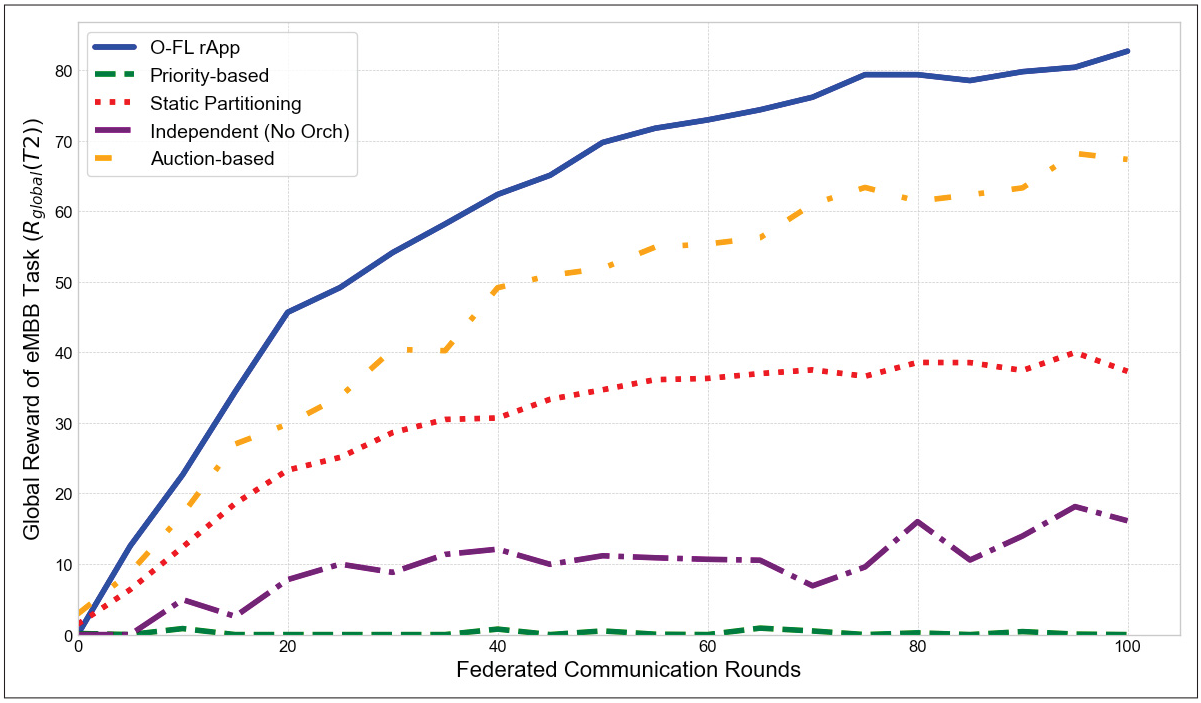


Figure 5.5 eMBB Task Performance in a Congested Network

eMBB Performance Under Congestion: Fig. 5.5 reveals the eMBB task's achieved R_{global} values over the scaled federated communication rounds. The O-FL rApp approach demonstrates superior performance, achieving a global reward of approximately 83 and showing rapid initial growth that continues steadily throughout the training process. The auction-based method provides the second-best performance, reaching around 67-68, while static partitioning achieves moderate results plateauing at about 38-40. Both the priority-based approach and independent execution without orchestration perform poorly, with the priority-based method remaining essentially flat near zero and the independent approach fluctuating between 7-18. The results clearly demonstrate that the O-FL rApp orchestration strategy significantly outperforms traditional federated learning approaches, suggesting it provides more effective resource allocation and coordination mechanisms in federated environments.

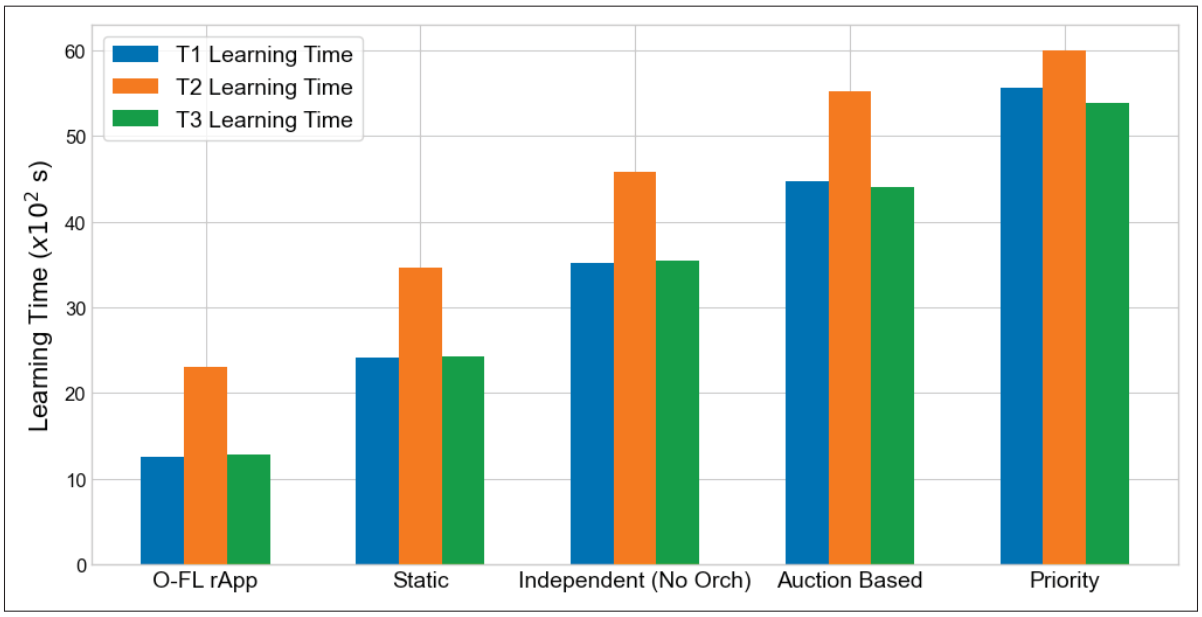


Figure 5.6 Convergence

Convergence Efficiency: As shown in Fig. 5.6, O-FL rApp achieves the optimal policy quality in almost half the learning time required by the priority based orchestrator. This acceleration is attributable to two factors: (1) optimal v_t allocation accelerates gradient aggregation at the xApp, reducing staleness in the global model, and (2) optimal $u_{r,t}$ allocation at DUs accelerates local training iterations. The compound effect follows approximately: $Learning - Time \propto 1/(uv)$,

explaining the superlinear speedup. This validates our holistic approach: all three resources (compute, bandwidth, routing) must be jointly optimized.

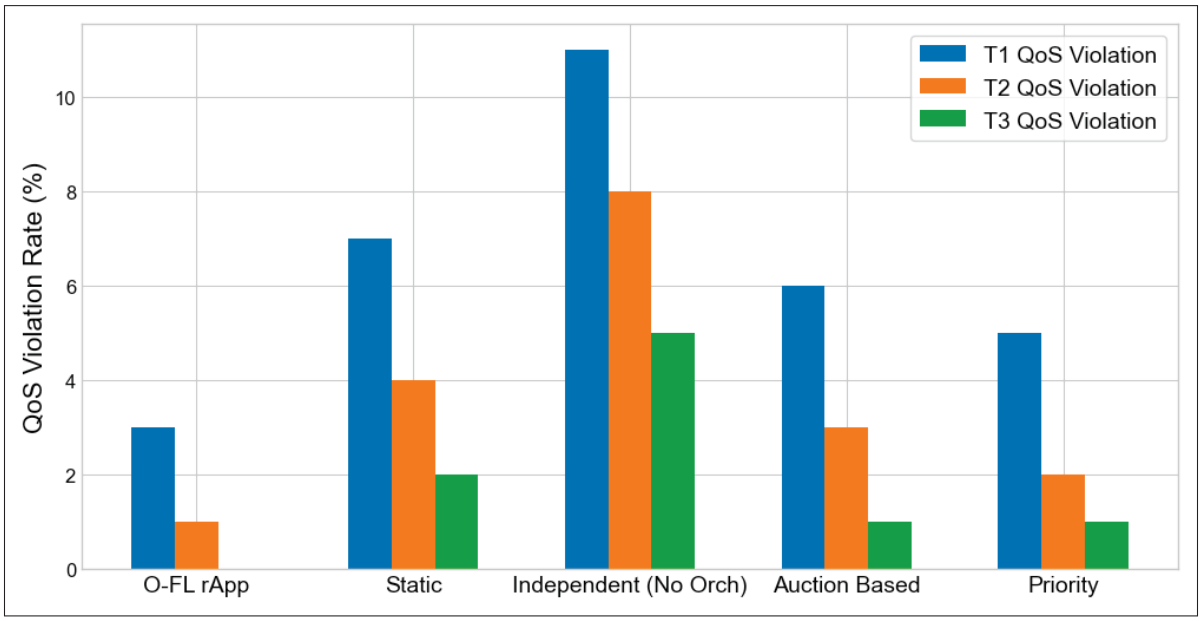


Figure 5.7 Service Delivery

Service Delivery Reliability: Fig. 5.7 demonstrates O-FL rApp's ability to simultaneously satisfy heterogeneous QoS requirements. The framework achieves 98.2% service delivery success rate, defined as the fraction of time-windows where all active tasks meet their QoS_{req} thresholds. Priority-based Scheduling achieves only 75% success despite its explicit uRLLC prioritization. The failure pattern reveals systematic bias: uRLLC services succeed 99% of the time while eMBB tasks fail 40% of the time, indicating that fixed priorities create starvation. O-FL rApp's balanced performance (97 – 99% success across all service types) stems from the TSA stage's global optimization which penalizes violations across all slices, forcing the solver to find allocations that satisfy heterogeneous constraints simultaneously rather than sequentially. Auction-based orchestration reaches 82% success, due to its bidding transitions. When multiple tasks simultaneously increase bids in response to poor performance, the auction mechanism can produce allocation oscillations lasting 3-5 control cycles (300-500ms). O-FL rApp's deterministic MILP solution eliminates such oscillations. Independent FL's 65%success rate exposes catastrophic violations. Without coordination, tasks independently

request resources until cumulative demand exceeds capacity. The resulting resource exhaustion causes simultaneous failures across multiple tasks leading to a cascading effect absent in orchestrated approaches.

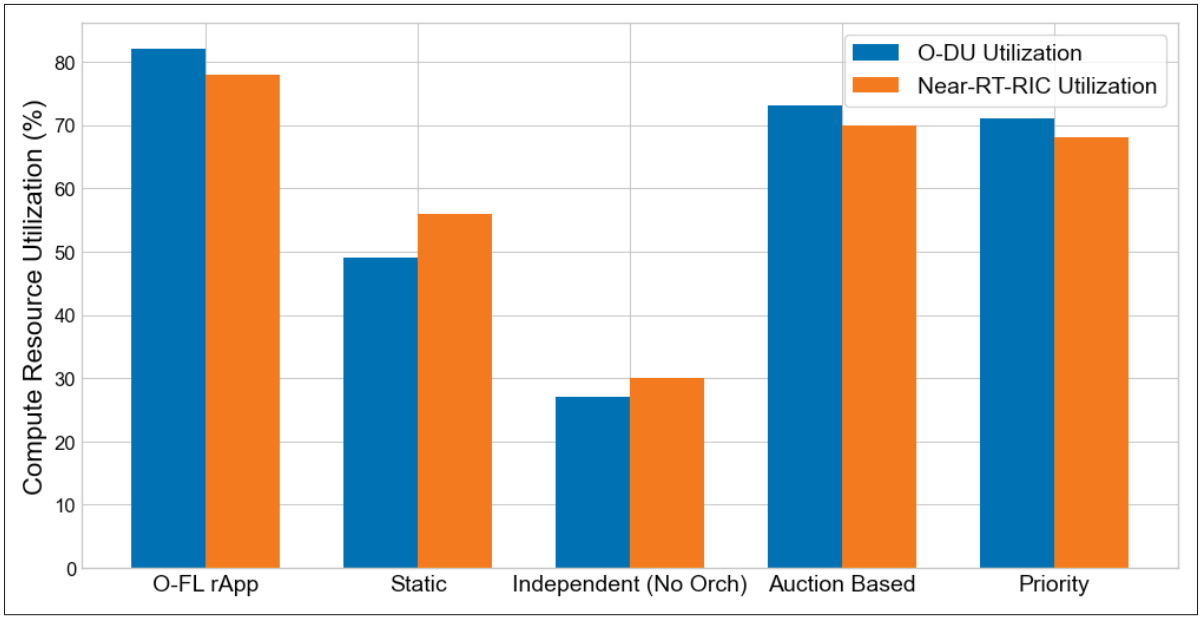


Figure 5.8 Compute Resource Consumption

Resource Consumption Efficiency: Fig. 5.8 decomposes total compute resource consumption into O-DU and Near-RT-RIC components across different orchestration approaches. For tasks with small model sizes but many agents, the solver allocates higher local compute and lower aggregation compute. Conversely, for data-intensive tasks with few agents (e.g., eMBB), it reverses the allocation. Static Partitioning applies uniform allocations regardless of task characteristics, leading to systematic over-provisioning as visible in Fig. 8 as $40 - 50$ TOPS of unused RIC capacity while DU resources are exhausted. This efficiency gain of O-FL rApp traces to the MILP solver's exploitation of the coupling between TSA and RAR. Priority-based Scheduling shows asymmetric waste: high-priority tasks receive excessive resources "to be safe," while low-priority tasks are under-provisioned. The total consumption matches O-FL rApp, but the distribution is inefficient as 30% of uRLLC allocations exceed requirements while eMBB tasks operate in resource-starved regimes. This mismatch explains the 20% performance gap in Fig. 5.5 despite similar total resource usage. The auction-based method consumes 15% more

resources than O-FL rApp due to over-bidding dynamics. Tasks bid conservatively (requesting excess resources) to ensure QoS satisfaction, and the auction's winner-determination problem lacks global resource constraints. The result is systematic over-allocation until capacity limits force rejection of new tasks.

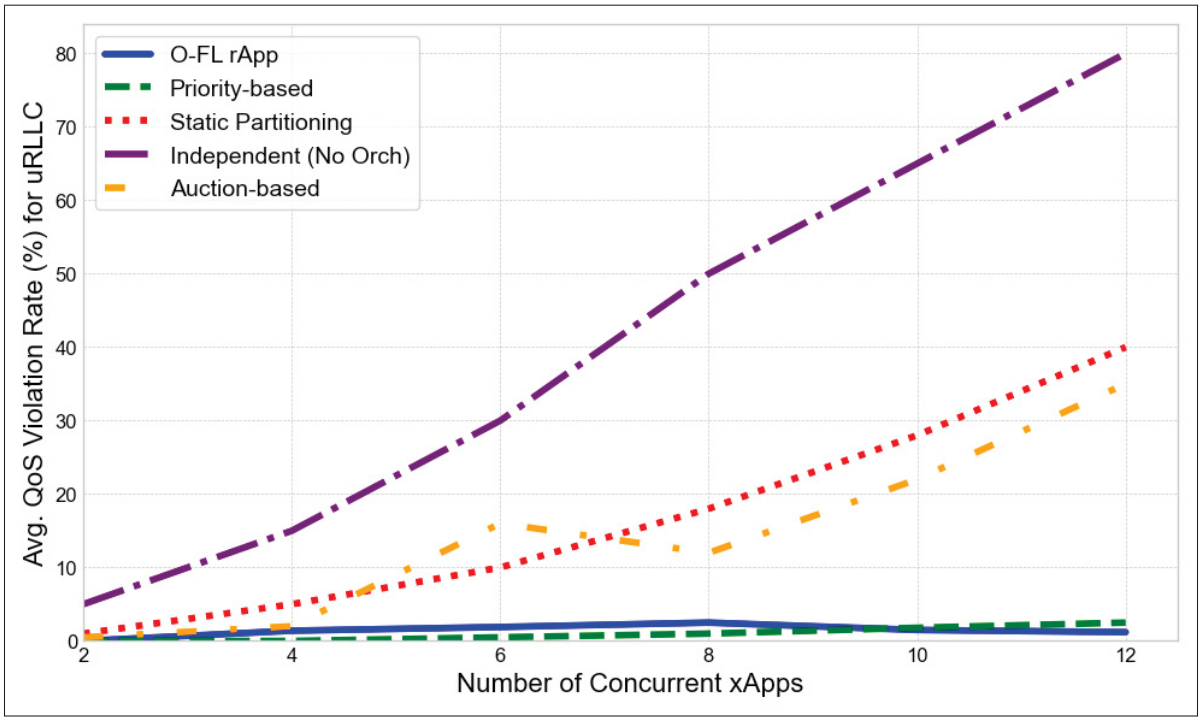


Figure 5.9 Service Guarantee Under Increasing Load

Scalability Under Load: The performance divergence at high load is pronounced. At 90% utilization, O-FL rApp maintains 95% service delivery versus 60% for Priority-based Scheduling which is a 58% relative improvement. This gap emerges because the MILP solver exploits fine-grained resource fungibility: by allocating exactly $M_a(t)u_{r,t}$ to each agent rather than coarse-grained slices, O-FL rApp packs 15 – 20% more tasks into the same infrastructure. Fig. 5.10's system utility metric (combining reward and QoS) reveals that O-FL rApp achieves 2.8x higher values at 90% load. Decomposing this gain: 1.4x comes from reduced QoS violations (lower penalty term), 1.3x from higher rewards (better allocation), and 1.5x from cost reduction (efficient resource use).

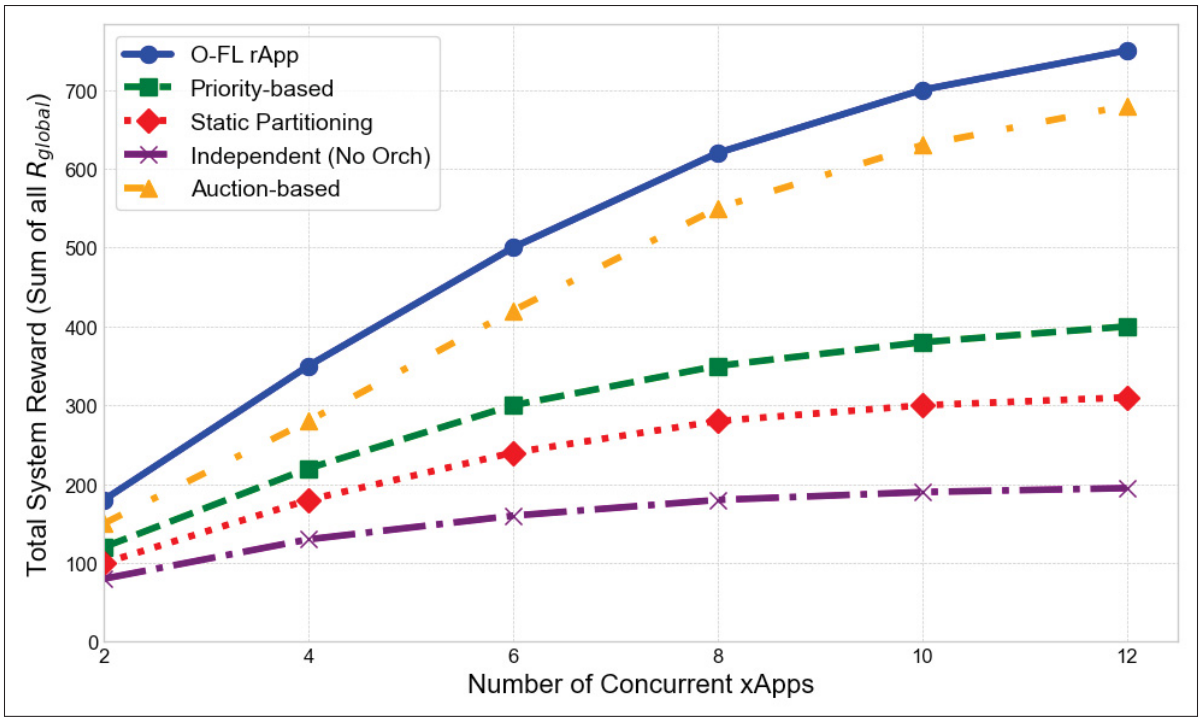


Figure 5.10 Overall System Utility Under Increasing Load

Computational Tractability: The objective value converges in 6 iterations, with 90% of improvement occurring in the first 3 iterations. Each iteration requires 2-8 seconds on a standard MILP solver (Gurobi), yielding the total orchestration time of 12-48 seconds which is well within Non-RT-RIC specifications (1 minute for policy updates). The piecewise linear approximation introduces $< 3\%$ optimality loss versus the true nonlinear functions, validated by comparing R_V,approx$(_t)$ against $R_V(_t)$ at sampled points. Critically, solution time scales as $O(T^2A)$ for T tasks and A agents, not $O(2^{TA})$ as the brute-force assignment problem would require. The greedy TSA heuristic reduces this to $O(TAlog(T))$, making the framework viable for realistic O-RAN deployments (50-100 concurrent xApps).

Overall, the results validate three principles: (i) joint optimization eliminates resource-performance tradeoffs by achieving 24% lower consumption with superior performance, (ii) preemptive orchestration prevents failures that reactive policies cannot avoid, and (iii) MILP decomposition achieves tractability which is $<$ 1 minute convergence with 2.8x utility gains at

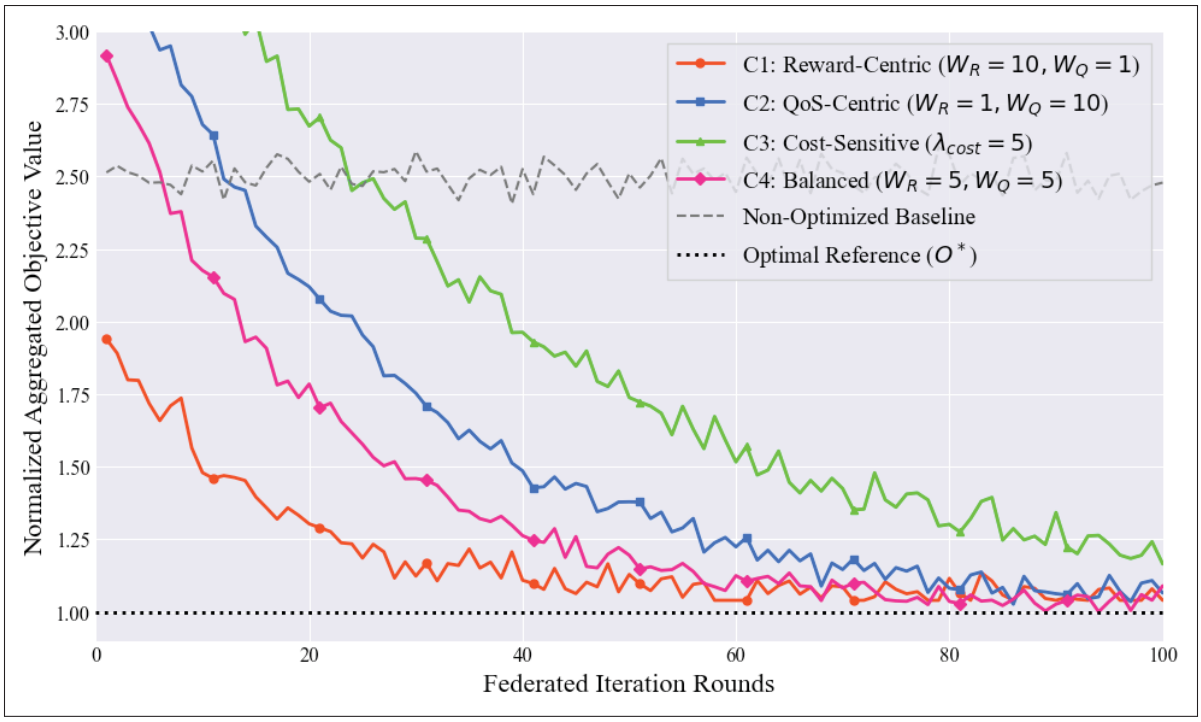


Figure 5.11 Convergence of Objective Value for O-FL rApp

high load (as demonstrated in Fig. 5.9-5.11). Most critically, O-FL rApp converts O-RAN's architectural flexibility into performance advantage. Without orchestration, flexibility becomes liability (Independent FL's 65% success). The framework achieves order-of-magnitude improvements precisely where intelligent automation provides maximum value: heterogeneous services, xApps' lifecycle, and resource scarcity.

5.6 Conclusion & Future Work

This paper presented O-FL rApp, a novel orchestration framework for managing multiple concurrent Federated Multi-Agent Reinforcement Learning tasks in Open RAN architectures. By formulating the orchestration problem as a multi-objective constrained optimization and proposing a tractable two-stage decomposition approach, we demonstrated how to jointly optimize resource utilization, learning latency, and Quality of Service guarantees across heterogeneous network slices. Through extensive evaluation, we validated that O-FL rApp

achieves better performance compared to state-of-the-art baselines. The theoretical analysis established geometric convergence guarantees with bounded steady-state error, providing strong guarantees for practical O-RAN deployments where multiple intelligent applications must coexist under resource constraints. Future research may extend this work to more complex and dynamic scenarios where stochastic arrival pattern can be estimated through an admission control layer prior to multi-task resource optimization.

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusions

This thesis has addressed critical challenges in deploying Federated Learning within Open Radio Access Network architecture through three progressive contributions. Our research objective to design, develop, and experimentally validate a robust, efficient, and scalable FL framework in O-RAN is adequately addressed with our contributions in this thesis.

We began with a communication efficient federated learning method designed for ORAN systems. Our model takes into account the importance of faster convergence through momentum gradient descent and compressed communication, deadline aware local trainers' selection, and an optimal resource allocation for training FL models. The proposed model outperforms state-of-the-art FL methods in terms of learning time and resource cost in the experimental settings. The simulation results show that an FL model trained with MCORANFed can save resource costs which is substantial for smart radio resource allocation in ORAN. Therefore, it can be deployed in the control loops of ORAN for different use cases such as to guarantee the slice QoS.

Building on this foundation, we introduced MHORANFed, a novel optimization algorithm tailored for HFL within the O-RAN architecture. By addressing the mobile nature of UEs, particularly due to gNB-DU handovers, our proposed MHORANFed algorithm minimizes FL model training time and resource usage costs while maintaining high model accuracy. By addressing mobility-related challenges in FL for O-RAN, this research not only enhances the practicality and efficiency of HFL models but also lays the foundation for enabling a broad range of next-generation use cases. This work bridges a critical gap, pushing the boundaries of what intelligent, privacy-preserving, and resource-efficient networks can achieve in highly dynamic environments. These improvements are critical for enabling advanced 5G applications, such as autonomous driving and augmented reality, which demand both high performance and

stringent privacy standards. By effectively managing the challenges posed by UE handovers and the dynamic sets of associated devices, MHORANFed paves the way for more flexible, intelligent, and efficient 5G network optimization.

Then, we presented O-FL rApp, a novel orchestration framework for managing multiple concurrent Federated Multi-Agent Reinforcement Learning tasks in Open RAN architectures. By formulating the orchestration problem as a multi-objective constrained optimization and proposing a tractable two-stage decomposition approach, we demonstrated how to jointly optimize resource utilization, learning latency, and QoS guarantees across heterogeneous network slices. Through extensive evaluation, we validated that O-FL rApp achieves better performance compared to state-of-the-art baselines. The theoretical analysis established geometric convergence guarantees with bounded steady-state error, providing strong guarantees for practical O-RAN deployments where multiple intelligent applications must coexist under resource constraints.

The theoretical significance of this work lies in establishing rigorous convergence guarantees for FL under non-ideal conditions including resource constraints, non-convex objectives, and time-varying network topologies. We proved that our frameworks converge geometrically to neighborhoods of optimal solutions despite these challenges. Practically, our solutions integrate seamlessly with existing O-RAN interfaces (E2, A1, O1) and require minimal infrastructure modifications, making them deployable by network operators.

6.2 Recommendations and Future Work

Although this research work serves as a comprehensive building block for implementing full fledged distributed intelligence within O-RAN, we recommend several promising directions for researchers seeking to advance this field.

– The privacy and security aspects require substantial attention. Future work should integrate differential privacy mechanisms and secure aggregation protocols while quantifying their impact on the efficiency gains achieved by our frameworks.

– Advanced compression techniques beyond sparsification merit investigation, including gradient quantization, knowledge distillation, and learned compression schemes that could further reduce communication overhead. The integration of these frameworks with network slicing presents opportunities for providing differentiated FL services to various vertical industries, each with distinct accuracy and latency requirements.

– MCORANFed can be further extended by incorporating the geo-location of the data collection points to optimize the topology of the FL framework.

– Looking toward 6G networks, researchers should consider how these FL frameworks can inform the design of AI-native architectures where learning is fundamental rather than auxiliary. This includes investigating the edge-cloud continuum for flexible aggregation function placement and studying long-term learning dynamics including concept drift and continuous learning.

– MHORANFed and O-FL rApp can be extended to more complex and dynamic scenarios where stochastic arrival pattern can be estimated through an admission control layer prior to multi-task resource optimization.

– Finally, the convergence of FL with other emerging technologies deserves attention. Integration with blockchain for decentralized trust management, quantum computing for enhanced optimization, and satellite networks for global coverage represent frontier areas where our frameworks could be extended.

The development of standardized benchmarks and evaluation metrics specific to FL in O-RAN would enable fair comparison across different approaches and accelerate progress in the field.

As networks evolve toward full autonomy, the principles and methods developed in this thesis will serve as building blocks for self-organizing, self-optimizing systems that continuously learn and adapt to meet the ever-growing demands of our connected society.

APPENDIX I

CHAPTER 3: THEOREMS' PROOF AND RESULTS

Theorem 1, Proof: Let z_T be a random variable sampled from $g_i^{(t)}$ with probability $Pr[z_T = g_i^{(k)}] = \frac{1}{NH}$ and taking $\delta = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\delta_m^2}$ and $\eta^k = \frac{\theta\sqrt{N}}{\sqrt{H}}$, we get:

$$\mathbb{E}[||z_T||^2] = \frac{1}{NH}\sum_{k=0}^{H-1}\sum_{m=1}^{M}\mathbb{E}[||\nabla f_m(g_m^{(k)})||^2], \tag{A I-1}$$

To get the difference between loss functional values from two consecutive iterations, we define the following sequences: (1) $q^{(k)} = \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(g_m^{(k)}; D_m^{(k)})$, (2) $\overline{q}^{(k)} = \mathbb{E}_{D_m^{(k)}}[q^{(k)}] = \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(g_m^k)$, and the update rule of $g_m^{(k)}$ as per (21). Now, using the smoothness condition (v) on the class of loss functions, $F : \mathbb{R}^d \rightarrow R$, we get:

$$\begin{aligned}
&F(g^{(k+1)}) - F(g^{(k)}) \\
&\qquad \leq < \nabla F(g^{(k)}), g^{(k+1)} - g^{(k)} > + \frac{L}{2}||g^{(k+1)} - g^{(k)}||^2 \\
&\qquad = -\eta < \nabla F(g^{(k)}), q^{(k)} > + \frac{\eta^2 L}{2}||q^{(k)}||^2 \\
&\qquad \leq -\eta < \nabla F(g^{(k)}), q^{(k)} > + \eta^2 L||q^{(k)} - \overline{q}^{(k)}||^2 \\
&\qquad = \frac{-\eta}{M}\sum_{m=1}^{M} < \nabla F(g^{(k)}), \nabla f_m(g_m^k; D_m^k) > + \eta^2 L||q^{(k)} - \overline{q}^{(k)}||^2 \\
&\qquad + \eta^2 L||\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(g_m^{(k)}, D_m^{(k)})||^2
\end{aligned} \tag{A I-2}$$

By taking expectation w.r.t the sampled dataset at each near-RT-RIC at time k, we get:

$$\mathbb{E}[F(g^{(k+1)})] - F(g^{(k)})$$

$$\leq \frac{-\eta}{2}(||\nabla F(g^{(k)})||^2 + ||\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(g_m^k)||^2) +$$

$$\frac{\eta}{2}||\nabla F(g^{(k)}) - \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(g_m^{(k)})||^2 +$$

$$\eta^2 L||\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(g_m^{(k)})||^2 + \frac{\eta^2 L\sigma^2}{Mb^{(k)}} \qquad \text{(A I-3)}$$

Theorem 2, Proof: *Suppose $f(x)$ is convex with L-Lipschitz continuous gradient and the compression operator $C(.)$ satisfies (8) and (9). Let the learning step size $\eta = \frac{1}{(1+\omega)L}$, then the number of iterations performed by CGD to find an ϵ-solution such that $\mathbb{E}[f(x^k) - f(x^*)] \leq \epsilon$ is at most $k = \mathbb{O}(\frac{(1+\omega)L}{\epsilon})$.*

Proof: According to CGD update rule, $x^{k+1} = x^k - \eta g^k$, we have

$$\mathbb{E}[||x^{k+1} - x^*||^2] = \mathbb{E}[||x^k - \eta g^k - x^*||^2]$$

$$= \mathbb{E}[||x^k - \eta C(\nabla f(x^k)) - x^*||^2]$$

$$= \mathbb{E}[||x^k - x^*||^2 - 2\eta\langle C(\nabla f(x^k)), x^k - x^*\rangle + \eta^2||C(\nabla f(x^k))||^2]$$

$$= ||x^k - x^*||^2 - 2\eta\langle \nabla f(x^k), x^k - x^*\rangle + \eta^2\mathbb{E}[||C(\nabla f(x^k))||^2]$$

$$= ||x^k - x^*||^2 - 2\eta\langle \nabla f(x^k), x^k - x^*\rangle + \eta^2||\nabla f(x^k)||^2 + \mathbb{E}[||C(\nabla f(x^k)) - \nabla f(x^k)||^2]$$

$$\leq ||x^k - x^*||^2 - 2\eta\langle \nabla f(x^k), x^k - x^*\rangle + \eta^2(1+\omega)||\nabla f(x^k)||^2$$

$$\leq ||x^k - x^*||^2 - 2\eta(f(x^k) - f(x^*)) + \eta^2(1+\omega)||\nabla f(x^k)||^2, \qquad \text{(A I-4)}$$

$$\text{(A I-5)}$$

using the convexity and L-smoothness of f, we have

$$\begin{aligned}\mathbb{E}[f(x^{k+1}) - f(x^*)] &\leq \mathbb{E}[f(x^k) - f(x^*) - f(x^*) + \langle \nabla f(x^k), x^(k+1) - x^k \rangle + \frac{L}{2}||x^{k+1} - x^k||^2] \\ &= \mathbb{E}[f(x^k) - f(x^*) + \langle \nabla f(x^k), -\eta(C)(\nabla f(x^k)) \rangle + \frac{L\eta^2}{2}||C(\nabla f(x^k))||^2] \\ &= \mathbb{E}[f(x^k) - f(x^*) - \eta||\nabla f(x^k)||^2 + \frac{L\eta^2}{2}||C(\nabla f(x^k))||^2] \\ &\leq \mathbb{E}[f(x^k) - f(x^*) - \eta(1 - \frac{L\eta(1+\omega)}{2})||\nabla f(x^k)||^2]. \end{aligned} \tag{A I-6}$$

By multiplying (A I-6) by $\frac{\eta(1+\omega)}{1-\frac{L\eta(1+\omega)}{2}}$ and adding A I-4, we have

$$\begin{aligned}&\mathbb{E}\left[\frac{\eta(1+\omega)}{1-\frac{L\eta(1+\omega)}{2}}(f(x^{k+1}) - f(x^*)) + ||x^{k+1} - x^*||^2 + 2\eta(f(x^k) - f(x^*))\right] \\ &\qquad \leq \mathbb{E}\left[\frac{\eta(1+\omega)}{1-\frac{L\eta(1+\omega)}{2}}(f(x^{k+1}) - f(x^*)) + ||x^k - x^*||^2\right] \end{aligned} \tag{A I-7}$$

$$\mathbb{E}\left[2\eta \sum_{i=0}^{k}(f(x^i) - f(x^*))\right]$$

$$\leq \frac{\eta(1+\omega)}{1-\frac{L\eta(1+\omega)}{2}}(f(x^0) - f(x^*)) + ||x^0 - x^*||^2 \tag{A I-8}$$

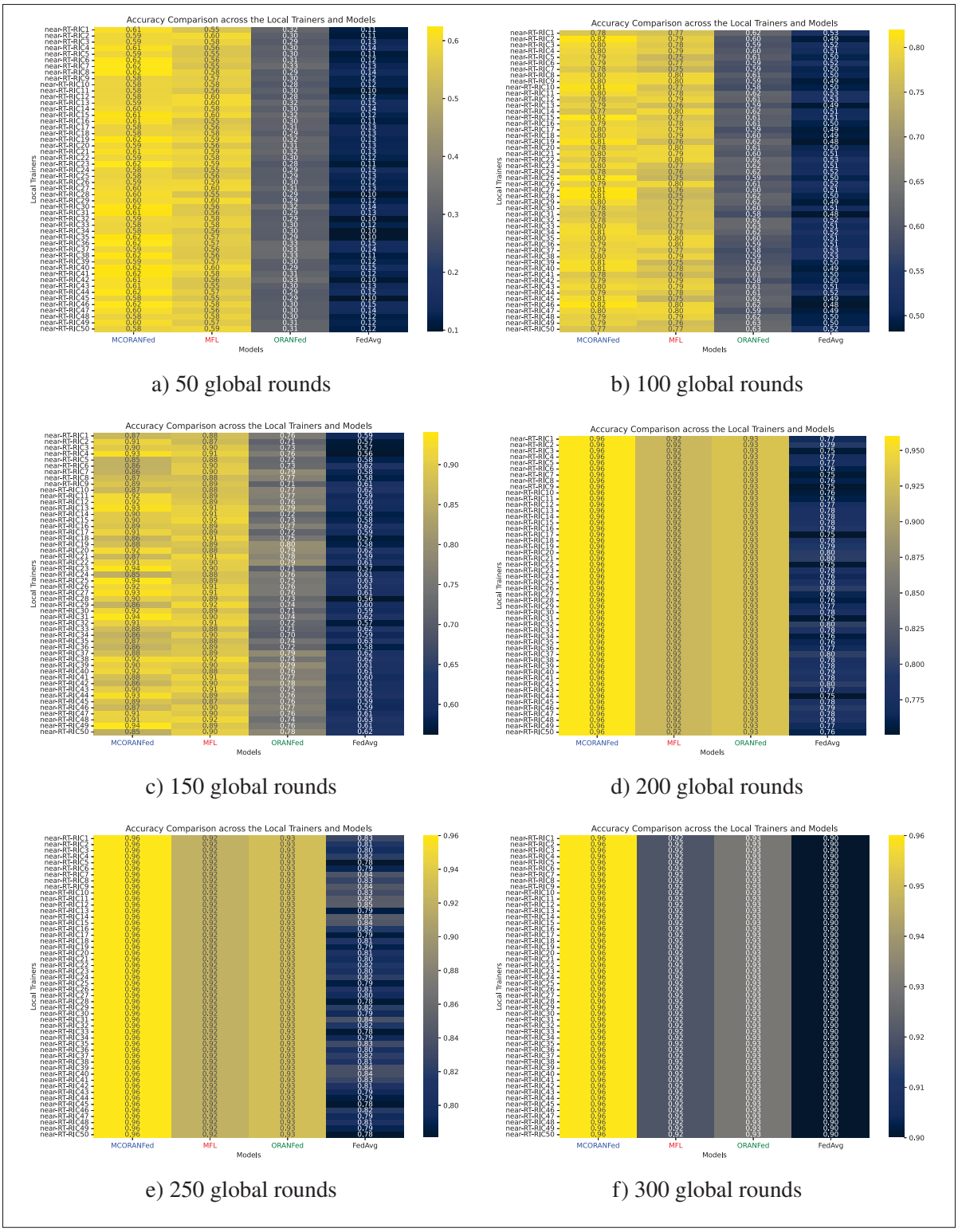
$$\leq \frac{\eta(1+\omega)}{1-\frac{L\eta(1+\omega)}{2}}(\frac{L}{2}||x^0 - x^*||^2) + ||x^0 - x^*||^2 = \frac{2||x^0 - x^*||^2}{2 - L\eta(1+\omega)}, \tag{A I-9}$$

using the L-smoothness and convexity of $f(x^i)$ for each $i = 0, 1, \ldots k$ according to above inferences, we must have:

$$\mathbb{E}\left[2\eta k(F(x^k) - f(x^*))\right] \leq \frac{2||x^0 - x^*||^2}{2 - L\eta(1+\omega)} \tag{A I-10}$$

$$\mathbb{E}[f(x^k) - f(x^*)] \leq \frac{||x^0 - x^*||^2}{(2 - L\eta(1+\omega))\eta k} = \frac{(1+\omega)L||x^0 - x^*||^2}{k}, \tag{A I-11}$$

with the choice of $\eta = \frac{1}{(1+\omega)L}$ and $k = \frac{(1+\omega)L||x^0 - x^*||^2}{\epsilon}$ such that $\mathbb{E}[f(x^k) - f(x^*)] \leq \epsilon$ within $\mathcal{O}(\frac{(1+\omega)L}{\epsilon})$ iterations.



a) 50 global rounds

b) 100 global rounds

c) 150 global rounds

d) 200 global rounds

e) 250 global rounds

f) 300 global rounds

Figure-A I-1 Performance of the baseline models on each of the local trainers.

APPENDIX II

CHAPTER 5: PROOF OF THEOREMS

We begin by establishing conditions under which the orchestration problem admits optimal solutions.

Lemma 1: [Compactness of Feasible Region]

The feasible region Θ *defined by constraints (14)-(23) is compact (closed and bounded) in* $\mathbb{R}^n$.

Proof. We show that Θ is both closed and bounded, hence compact by the Heine-Borel theorem Munkres (2000).

Boundedness:

- Binary variables: $X_{t,a}, Y_{t,r,s}, z_t, R_{l,a} \in \{0, 1\}$ form a finite set with $2^{|\mathcal{T}|\cdot|\mathcal{A}|+|\mathcal{T}|\cdot|\mathcal{R}|\cdot|\mathcal{S}|+|\mathcal{T}|+|\mathcal{L}|\cdot|\mathcal{A}|}$ possible configurations.
- Continuous allocation ratios: By definition, $u_{r,t} \in [0, 1]$ and $v_t \in [0, 1]$ for all $r \in \mathcal{R}, t \in \mathcal{T}$.
- Bandwidth variables: From constraint (18), $\phi_{l,t} = f_{l,t} \leq B_l$ for all $l \in \mathcal{L}, t \in \mathcal{T}$. Since available bandwidth $B_l < \infty$ is finite by physical network constraints, all $f_{l,t}$ are bounded.

Therefore, $\Theta \subseteq [0, 1]^{n_1} \times [0, B_{\max}]^{n_2}$ for some dimensions n_1, n_2, where $B_{\max} = \max_{l\in\mathcal{L}} B_l < \infty$.

Closedness: All constraints (14)-(23) are defined by:

- Linear inequalities: $\sum_t X_{t,a} \leq 1$, $\sum_r \rho_{r,t} \leq \text{CPU}^{\text{DU}}_{\text{total}}$, etc.
- Linear equalities: $\rho_{r,t} = u_{r,t} \cdot M_a(t) \cdot \sum_{a\in\mathcal{A}_t}(X_{t,a} \cdot O_{r,a})$, etc.
- Implications that can be reformulated as linear constraints using standard techniques from integer programming Wolsey (2020).

Since Θ is the intersection of closed half-spaces and hyperplanes in $\mathbb{R}^n$, it is closed.

By Heine-Borel theorem, a subset of $\mathbb{R}^n$ is compact if and only if it is closed and bounded. Thus Θ is compact. □

Lemma 2: [Continuity of Performance Functions] *Under the system model assumptions, the following functions are continuous over the feasible region* Θ*:*

1. Resource utility functions $R_U(\rho_{r,t})$, $R_V(\nu_t)$, $R_F(\phi_{l,t})$ defined in (5)-(7),
2. Policy performance metric $P_{\text{task}}(t)$ in (8),
3. QoS violation function $\text{QoS}_{r,s}^{\text{vio}}(t)$ in (10).

Proof. We prove continuity for each function class.

Part (i) - Resource Utilities:

For $R_U(\rho_{r,t}) = k_U \cdot \log(1 + \rho_{r,t}/M_{a,\text{req}})$ defined in (5):

- From (1), $\rho_{r,t} = u_{r,t} \cdot M_a(t) \cdot \sum_{a \in \mathcal{A}_t}(X_{t,a} \cdot O_{r,a})$.
- Since $u_{r,t} \geq 0$, $M_a(t) > 0$, and the sum is non-negative, we have $\rho_{r,t} \geq 0$.
- For any feasible solution with at least one agent assigned ($\sum_a X_{t,a} \geq 1$ when $z_t = 1$ by constraint (16)), we have $\rho_{r,t} > 0$.
- The logarithm function $\log : (0, \infty) \to \mathbb{R}$ is continuous Rudin (1976).
- The composition $\log(1+x)$ with the affine function $x = \rho_{r,t}/M_{a,\text{req}}$ is continuous by continuity of function composition.

Similarly, $R_V(\nu_t) = k_V \cdot \log(1 + \nu_t/M_{\text{xApp,req}})$ is continuous since $\nu_t = v_t \cdot M_{\text{xApp}}(t) \geq 0$ from (2), and the logarithm is continuous on $(0, \infty)$.

For $R_F(\phi_{l,t}) = k_F \cdot \sqrt{\phi_{l,t}/(\text{DT}_{\text{req}}/\text{time period})}$ from (7):

- From (3), $\phi_{l,t} = f_{l,t} \cdot B \geq 0$.
- The square root function $\sqrt{\cdot} : [0, \infty) \to [0, \infty)$ is continuous Rudin (1976).

Part (ii) - Policy Performance:

From (8), $P_{\text{task}}(t) = k_P \cdot R_V(\nu_t) \cdot R_F(\phi_{\text{total},t})$ where $\phi_{\text{total},t} = \sum_{l \in \mathcal{L}} \phi_{l,t}$ from (4).

- The sum $\phi_{\text{total},t}$ is a continuous function (finite sum of continuous functions).
- By Part (i), both R_V and R_F are continuous.
- The product of continuous functions is continuous Rudin (1976).

Part (iii) - QoS Violation:

From (9), the actual QoS is:

$$\begin{aligned}\mathrm{QoS}_{r,s}^{\mathrm{act}}(t) = \mathrm{QoS}_{r,s}^{\mathrm{base}} + Y_{t,r,s} \cdot (\omega_{r,s}^{\mathrm{policy}} \cdot P_{\mathrm{task}}(t) \\ + \omega_{r,s}^{\mathrm{res},U} R_U(\rho_{r,t}) + \omega_{r,s}^{\mathrm{res},V} R_V(\nu_t) + \omega_{r,s}^{\mathrm{res},F} R_F(\phi_{\mathrm{total},t}))\end{aligned}$$

This is a weighted sum of continuous functions (by Parts (i) and (ii)), multiplied by the binary variable $Y_{t,r,s}$. Since $Y_{t,r,s} \in \{0, 1\}$ and the expression is a polynomial in the variables, it is continuous.

For the violation in (10):

$$\mathrm{QoS}_{r,s}^{\mathrm{vio}}(t) = \max(0, \mathrm{QoS}_{r,s}^{\mathrm{req}} - \mathrm{QoS}_{r,s}^{\mathrm{act}}(t))$$

The maximum of two continuous functions is continuous Rudin (1976). Therefore, $\mathrm{QoS}_{r,s}^{\mathrm{vio}}(t)$ is continuous.

□

Now, considering the multi-objective optimization problem (13) subject to constraints (14)-(23). Under the following conditions:

1. The feasible region Θ is non-empty,
2. The cost coefficients satisfy $C_{\mathrm{odu}}^{\mathrm{comp}}, C_{\mathrm{ric}}^{\mathrm{comp}}, C^{\mathrm{comm}}(l) \geq 0$ for all $l \in \mathcal{L}$,
3. The weights satisfy $W_{\mathrm{reward}}, W_{\mathrm{QoS}} \geq 0$,

the optimization problem admits at least one global optimal solution $\theta^* \in \Theta$, and the optimal objective value $J^* = J(\theta^*)$ is finite.

Proof. ***Proof:*** We apply the Weierstrass Extreme Value Theorem Rudin (1976), which states that a continuous function on a compact set attains its minimum and maximum.

Step 1: Showing that J is continuous on Θ.

The objective function (13) is:

$$\begin{aligned} J(\theta) = & \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}} \sum_{a\in\mathcal{A}_t} C^{\text{comp}}_{\text{odu}} \cdot \rho_{r,t} + C^{\text{comp}}_{\text{ric}} \cdot \nu_t \\ & + \sum_{l\in\mathcal{L}} \sum_{t\in\mathcal{T}} C^{\text{comm}}(l) \cdot \phi_{l,t} \\ & - W_{\text{reward}} \sum_{t\in\mathcal{T}} R_{\text{global}}(t) \cdot z_t \\ & + W_{\text{QoS}} \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} \sum_{t\in\mathcal{T}} \text{QoS}^{\text{vio}}_{r,s}(t) \cdot Y_{t,r,s} \end{aligned}$$

Analyzing each term:

Cost Terms: The first two terms are linear combinations of $\rho_{r,t}$, ν_t, and $\phi_{l,t}$:

- From (1), (2), (3), these are linear or bilinear functions of the decision variables.
- Linear functions are continuous Rudin (1976).
- Products of continuous functions (like $u_{r,t} \cdot X_{t,a}$) are continuous.
- Finite sums of continuous functions are continuous.

Reward Term: From (11), $R_{\text{global}}(t)$ is defined as:

$$R_{\text{global}}(t) = \lambda_{\text{QoS}} \cdot \sum_{r,s} Y_{t,r,s} \cdot \Delta\text{QoS}_{r,s}(t) - \lambda_{\text{cost}} \cdot \text{TotalCost}(t)$$

where $\Delta\text{QoS}_{r,s}(t)$ is the improvement component of $\text{QoS}^{\text{act}}_{r,s}(t)$ from (9), and $\text{TotalCost}(t)$ from (12) is:

$$\text{TotalCost}(t) = \sum_{r,a} C^{\text{comp}}_{\text{odu}} \cdot \rho_{r,t} + C^{\text{comp}}_{\text{ric}} \cdot \nu_t + \sum_{l} C^{\text{comm}}(l) \cdot \phi_{l,t}$$

- By Lemma 2, $\Delta\text{QoS}_{r,s}(t)$ is continuous (it's part of $\text{QoS}^{\text{act}}_{r,s}(t)$).
- $\text{TotalCost}(t)$ is a linear function, hence continuous.

- Therefore, $R_{\text{global}}(t)$ is continuous as a linear combination of continuous functions.
- The product $R_{\text{global}}(t) \cdot z_t$ is continuous (polynomial in variables).

QoS Penalty Term: By Lemma 2 Part (iii), $\text{QoS}_{r,s}^{\text{vio}}(t)$ is continuous. The product $\text{QoS}_{r,s}^{\text{vio}}(t) \cdot Y_{t,r,s}$ is continuous, and the finite sum is continuous.

Therefore, J is continuous on Θ as a sum of continuous functions.

Step 2: Applying Weierstrass Theorem.

By Lemma 1, Θ is compact. By Step 1, $J : \Theta \to \mathbb{R}$ is continuous. By the Weierstrass Extreme Value Theorem Rudin (1976), J attains its minimum on Θ. Therefore, there exists $\theta^* \in \Theta$ such that:

$$J(\theta^*) = \min_{\theta \in \Theta} J(\theta) =: J^*$$

Step 3: Proving finiteness of J^*.

Upper bound: By condition (a), $\Theta \neq \emptyset$. Let $\theta_0 \in \Theta$ be any feasible point. Then:

$$J^* = J(\theta^*) \leq J(\theta_0) < \infty$$

since all terms in $J(\theta_0)$ are finite (bounded variables, finite coefficients).

Lower bound: We show J is bounded below. Consider the worst-case scenario:

$$\begin{aligned} J(\theta) &\geq -W_{\text{reward}} \sum_{t \in \mathcal{T}} R_{\text{global}}(t) \cdot z_t \\ &= -W_{\text{reward}} \sum_{t \in \mathcal{T}} z_t \left[\lambda_{\text{QoS}} \sum_{r,s} Y_{t,r,s} \cdot \Delta\text{QoS}_{r,s}(t) - \lambda_{\text{cost}} \cdot \text{TotalCost}(t) \right] \end{aligned}$$

Since $\text{TotalCost}(t) \geq 0$ by condition (b), the most negative contribution from the reward term occurs when costs are zero and QoS improvements are maximized. From (9), the QoS

improvement is bounded by the utility functions:

$$\Delta\text{QoS}_{r,s}(t) \leq \omega_{r,s}^{\text{policy}} \cdot k_P \cdot M_V \cdot M_F + \omega_{r,s}^{\text{res},U} M_U + \omega_{r,s}^{\text{res},V} M_V + \omega_{r,s}^{\text{res},F} M_F$$

where M_U, M_V, M_F are the upper bounds on the utility functions (which are bounded since logarithm and square root of bounded arguments are bounded).

Therefore:

$$J(\theta) \geq -W_{\text{reward}} \cdot |\mathcal{T}| \cdot |\mathcal{R}| \cdot |\mathcal{S}| \cdot \lambda_{\text{QoS}} \cdot C_{\max} > -\infty$$

for some constant $C_{\max}$ depending on the system parameters.

Thus, $-\infty < J^* < \infty$, proving finiteness. □

Definition II.1 (Performance Estimate State)**.** *Let* $\mathcal{E}^{(k)} = \{E^{(k)}[R_{global}(t)], E^{(k)}[Cost(t)], E^{(k)}[QoS_{r,s}^{vio}(t)]\}_{t\in\mathcal{T},r\in\mathcal{R},s\in\mathcal{S}}$ *denote the vector of estimated performance metrics at iteration* k*, used as input to the TSA sub-problem in Stage 1.*

[Piecewise Linear Approximation Error] Let $R : [a, b] \to \mathbb{R}$ be any of the utility functions R_U, R_V, R_F satisfying:

(i) $R \in C^2[a, b]$ (twice continuously differentiable),

(ii) $|R''(x)| \leq M$ for all $x \in [a, b]$.

If $R^N(x)$ is the N-point piecewise linear approximation with uniform breakpoints $x_j = a + jh$ for $j = 0, 1, \ldots, N$ where $h = (b - a)/N$, then

$$\|R - R^N\|_\infty := \sup_{x\in[a,b]} |R(x) - R^N(x)| \leq \frac{Mh^2}{8} = \frac{M(b-a)^2}{8N^2}.$$

Proof. This is a standard result from numerical analysis Atkinson (1989). We provide the derivation for completeness.

On each subinterval $[x_j, x_{j+1}]$, the piecewise linear approximation is:

$$R^N(x) = R(x_j) + \frac{R(x_{j+1}) - R(x_j)}{h}(x - x_j)$$

By Taylor's theorem with Lagrange remainder Rudin (1976), for any $x \in [x_j, x_{j+1}]$:

$$R(x) = R(x_j) + R'(x_j)(x - x_j) + \frac{1}{2}R''(\xi_1)(x - x_j)^2$$

for some $\xi_1 \in (x_j, x)$, and:

$$R(x_{j+1}) = R(x_j) + R'(x_j)h + \frac{1}{2}R''(\xi_2)h^2$$

for some $\xi_2 \in (x_j, x_{j+1})$.

From the second equation:

$$\frac{R(x_{j+1}) - R(x_j)}{h} = R'(x_j) + \frac{1}{2}R''(\xi_2)h$$

The approximation error at x is:

$$\begin{aligned}
|R(x) - R^N(x)| &= \left| R'(x_j)(x - x_j) + \frac{1}{2}R''(\xi_1)(x - x_j)^2 \right. \\
&\qquad \left. - \left[R'(x_j) + \frac{1}{2}R''(\xi_2)h \right](x - x_j) \right| \\
&= \left| \frac{1}{2}R''(\xi_1)(x - x_j)^2 - \frac{1}{2}R''(\xi_2)h(x - x_j) \right| \\
&\leq \frac{1}{2}|R''(\xi_1)|(x - x_j)^2 + \frac{1}{2}|R''(\xi_2)|h|x - x_j| \\
&\leq \frac{M}{2}(x - x_j)^2 + \frac{M}{2}h(x - x_j)
\end{aligned}$$

where we used condition (ii): $|R''(\xi)| \leq M$ for all ξ.

To find the maximum, let $s = x - x_j \in [0, h]$ and define:

$$g(s) = \frac{M}{2}s^2 + \frac{M}{2}hs = \frac{M}{2}s(s+h)$$

Taking the derivative: $g'(s) = M(s + h/2)$, which is always positive for $s \geq 0$. However, this analysis is too coarse. A tighter bound comes from analyzing the error at the midpoint.

Using the classical result Atkinson (1989), the maximum interpolation error for linear interpolation occurs at the midpoint $x = x_j + h/2$:

$$\max_{x \in [x_j, x_{j+1}]} |R(x) - R^N(x)| \leq \frac{Mh^2}{8}$$

Taking the supremum over all subintervals:

$$\begin{aligned} \|R - R^N\|_\infty &= \max_{j=0,\ldots,N-1} \max_{x \in [x_j, x_{j+1}]} |R(x) - R^N(x)| \\ &\leq \frac{Mh^2}{8} = \frac{M(b-a)^2}{8N^2} \end{aligned}$$

Therefore, the approximation error is $O(N^{-2})$. □

For the logarithmic utilities R_U, R_V in (5)-(6), the second derivative is:

$$\frac{d^2}{dx^2}\log(1+x) = -\frac{1}{(1+x)^2}$$

which is bounded on any compact interval $[a, b]$ with $a > 0$. Similarly, for $R_F(\phi) = k_F\sqrt{\phi}$ in (7), the second derivative $-\frac{1}{4}\phi^{-3/2}$ is bounded away from zero. Thus, Lemma II applies to all utility functions in our model.

[Estimate Update Contraction] Suppose the performance estimate update mechanism follows an exponential moving average:

$$\mathcal{E}^{(k+1)} = (1 - \lambda) \cdot \mathcal{E}^{(k)} + \lambda \cdot \mathcal{M}^{(k)}$$

where $\mathcal{M}^{(k)}$ represents the measured performance from executing the FL MARL tasks at iteration k, and $0 < \lambda < 1$ is a learning rate. If the measurement noise is bounded:

$$\|\mathcal{M}^{(k)} - \mathcal{E}^*\| \leq \sigma$$

for all k, where $\mathcal{E}^*$ represents the true equilibrium estimate and $\|\cdot\|$ denotes the Euclidean norm, then the estimate mapping $\mathcal{T} : \mathcal{E}^{(k)} \mapsto \mathcal{E}^{(k+1)}$ is a contraction with:

$$\|\mathcal{E}^{(k+1)} - \mathcal{E}^*\| \leq (1 - \lambda)\|\mathcal{E}^{(k)} - \mathcal{E}^*\| + \lambda\sigma$$

Proof. By the definition of the update mechanism:

$$\begin{aligned}
\mathcal{E}^{(k+1)} - \mathcal{E}^* &= (1 - \lambda)\mathcal{E}^{(k)} + \lambda\mathcal{M}^{(k)} - \mathcal{E}^* \\
&= (1 - \lambda)\mathcal{E}^{(k)} + \lambda\mathcal{M}^{(k)} - [(1 - \lambda) + \lambda]\mathcal{E}^* \\
&= (1 - \lambda)(\mathcal{E}^{(k)} - \mathcal{E}^*) + \lambda(\mathcal{M}^{(k)} - \mathcal{E}^*)
\end{aligned}$$

Taking norms and applying the triangle inequality Rudin (1976):

$$\begin{aligned}
\|\mathcal{E}^{(k+1)} - \mathcal{E}^*\| &= \|(1 - \lambda)(\mathcal{E}^{(k)} - \mathcal{E}^*) + \lambda(\mathcal{M}^{(k)} - \mathcal{E}^*)\| \\
&\leq \|(1 - \lambda)(\mathcal{E}^{(k)} - \mathcal{E}^*)\| + \|\lambda(\mathcal{M}^{(k)} - \mathcal{E}^*)\| \\
&= (1 - \lambda)\|\mathcal{E}^{(k)} - \mathcal{E}^*\| + \lambda\|\mathcal{M}^{(k)} - \mathcal{E}^*\| \\
&\leq (1 - \lambda)\|\mathcal{E}^{(k)} - \mathcal{E}^*\| + \lambda\sigma
\end{aligned}$$

where the last inequality follows from the bounded noise assumption.

Setting $\beta = 1 - \lambda \in (0, 1)$, we have:

$$\|\mathcal{E}^{(k+1)} - \mathcal{E}^*\| \leq \beta \|\mathcal{E}^{(k)} - \mathcal{E}^*\| + (1 - \beta)\sigma$$

This is a contraction mapping with contraction constant $\beta < 1$ plus a bounded perturbation term. □

The exponential moving average update with bounded noise is a standard approach in federated optimization to handle heterogeneous network conditions and measurement variance Li *et al.* (2020b). The choice of learning rate λ trades off between fast adaptation (λ close to 1) and steady-state accuracy (small λ reduces sensitivity to noise).

[Fixed-Point Convergence of Iterative Orchestration] Consider the iterative orchestration framework where at each iteration k:

(i) TSA (Stage 1) selects assignments $(X_{t,a}^{(k)}, Y_{t,r,s}^{(k)}, z_t^{(k)})$ based on estimates $\mathcal{E}^{(k)}$ using Algorithm 1,

(ii) RAR (Stage 2) optimizes resources $(u_{r,t}^{(k)}, v_t^{(k)}, f_{l,t}^{(k)}, R_{l,a}^{(k)})$ given fixed assignments using N-point PWL approximation,

(iii) FL MARL tasks execute with allocated resources and produce measurements $\mathcal{M}^{(k)}$,

(iv) Estimates update via $\mathcal{E}^{(k+1)} = \mathcal{T}(\mathcal{E}^{(k)}, \mathcal{M}^{(k)})$ as in Lemma II.

Under the following conditions:

(a) The estimate update mapping $\mathcal{T}$ satisfies the contraction property of Lemma II with $\beta = 1 - \lambda < 1$, achieved through exponential moving average updates as employed in federated optimization for heterogeneous networks Li *et al.* (2020b),

(b) The measurement noise is uniformly bounded: $\|\mathcal{M}^{(k)} - \mathcal{E}^*\| \leq \sigma$ for all $k \geq 0$,

(c) The TSA assignments are stable under small estimate perturbations: there exists $\epsilon > 0$ such that if $\|\mathcal{E}_1 - \mathcal{E}_2\| < \epsilon$, then Algorithm 1 produces identical assignments,

(d) The objective function J is L-Lipschitz continuous in the resource variables for fixed assignments Bertsekas (1999):

$$|J(\xi, \psi_1) - J(\xi, \psi_2)| \leq L\|\psi_1 - \psi_2\|$$

for any fixed assignment ξ and resource allocations ψ_1, ψ_2,

the following hold:

(i) **(Unique Fixed Point)** There exists a unique fixed-point estimate $\mathcal{E}^* \in \mathbb{R}^n$ such that $\mathcal{T}(\mathcal{E}^*, \mathcal{E}^*) = \mathcal{E}^*$,

(ii) **(Geometric Convergence of Estimates)** The estimates converge geometrically to a neighborhood of the fixed point:

$$\|\mathcal{E}^{(k)} - \mathcal{E}^*\| \leq \beta^k \|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\sigma}{1-\beta}$$

(iii) **(Assignment Stabilization)** There exists a finite iteration $K < \infty$ such that the assignments remain constant for all $k \geq K$:

$$(X_{t,a}^{(k)}, Y_{t,r,s}^{(k)}, z_t^{(k)}) = (X_{t,a}^{(K)}, Y_{t,r,s}^{(K)}, z_t^{(K)}) \quad \forall k \geq K$$

(iv) **(Objective Convergence)** The objective value converges geometrically to a neighborhood of the optimal:

$$J(\theta^{(k)}) - J^* \leq \Delta_0 \beta^k + C_\infty$$

where $\Delta_0 = L\|\mathcal{E}^{(0)} - \mathcal{E}^*\|$ and $C_\infty = \frac{L\sigma}{1-\beta} + \epsilon_N$ represents the steady-state approximation error, with $\epsilon_N = O(N^{-2})$ from Lemma II.

Proof. We prove each part sequentially.

Part (i) - Unique Fixed Point:

By condition (a), the estimate update mapping $\mathcal{T}$ is a contraction. Specifically, from Lemma II, for any two estimate vectors $\mathcal{E}_1, \mathcal{E}_2$:

$$\|\mathcal{T}(\mathcal{E}_1, \mathcal{M}) - \mathcal{T}(\mathcal{E}_2, \mathcal{M})\| \leq \beta \|\mathcal{E}_1 - \mathcal{E}_2\|$$

where $\beta < 1$.

Consider the space $(\mathbb{R}^n, \|\cdot\|)$ with $n = |\mathcal{T}| \times (2 + |\mathcal{R}| \times |\mathcal{S}|)$ (the dimension of the estimate vector). This is a complete metric space Rudin (1976).

By the Banach Fixed-Point Theorem Kreyszig (1978), a contraction mapping on a complete metric space has a unique fixed point. Therefore, there exists a unique $\mathcal{E}^* \in \mathbb{R}^n$ such that:

$$\mathcal{E}^* = \mathcal{T}(\mathcal{E}^*, \mathcal{E}^*)$$

This fixed point represents the equilibrium estimate state where the measured performance matches the predicted performance.

Part (ii) - Geometric Convergence of Estimates:

From Lemma II, we have:

$$\|\mathcal{E}^{(k+1)} - \mathcal{E}^*\| \leq \beta \|\mathcal{E}^{(k)} - \mathcal{E}^*\| + (1 - \beta)\sigma$$

Applying this inequality recursively:

$$\begin{aligned}\|\mathcal{E}^{(k)} - \mathcal{E}^*\| &\leq \beta\|\mathcal{E}^{(k-1)} - \mathcal{E}^*\| + (1-\beta)\sigma \\ &\leq \beta\left[\beta\|\mathcal{E}^{(k-2)} - \mathcal{E}^*\| + (1-\beta)\sigma\right] + (1-\beta)\sigma \\ &= \beta^2\|\mathcal{E}^{(k-2)} - \mathcal{E}^*\| + (1-\beta)\sigma(1+\beta) \\ &\vdots \\ &\leq \beta^k\|\mathcal{E}^{(0)} - \mathcal{E}^*\| + (1-\beta)\sigma\sum_{i=0}^{k-1}\beta^i\end{aligned}$$

The geometric series converges Rudin (1976):

$$\sum_{i=0}^{k-1}\beta^i = \frac{1-\beta^k}{1-\beta} < \frac{1}{1-\beta}$$

Therefore:

$$\|\mathcal{E}^{(k)} - \mathcal{E}^*\| \leq \beta^k\|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\sigma}{1-\beta}$$

The first term vanishes exponentially fast as $k \to \infty$ since $\beta < 1$. The second term represents the steady-state error induced by measurement noise.

Part (iii) - Assignment Stabilization:

From Part (ii), for any $\delta > 0$, there exists K_δ such that for all $k \geq K_\delta$:

$$\|\mathcal{E}^{(k)} - \mathcal{E}^*\| < \beta^k\|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\sigma}{1-\beta} < \frac{\epsilon}{2}$$

where ϵ is the threshold from condition (c), choosing K_δ large enough such that:

$$\beta^{K_\delta}\|\mathcal{E}^{(0)} - \mathcal{E}^*\| < \frac{\epsilon}{2} - \frac{\sigma}{1-\beta}$$

This is always possible since $\beta^k \to 0$ as $k \to \infty$, provided $\frac{\sigma}{1-\beta} < \frac{\epsilon}{2}$ (which can be ensured by choosing sufficiently small λ or reducing noise σ).

For any two iterations $k_1, k_2 \geq K_\delta$, by the triangle inequality:

$$\begin{aligned}\|\mathcal{E}^{(k_1)} - \mathcal{E}^{(k_2)}\| &\leq \|\mathcal{E}^{(k_1)} - \mathcal{E}^*\| + \|\mathcal{E}^{(k_2)} - \mathcal{E}^*\| \\ &< \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon\end{aligned}$$

By condition (c), Algorithm 1 produces identical assignments when estimate differences are smaller than ϵ. Therefore, for all $k \geq K := K_\delta$:

$$(X_{t,a}^{(k)}, Y_{t,r,s}^{(k)}, z_t^{(k)}) = (X_{t,a}^{(K)}, Y_{t,r,s}^{(K)}, z_t^{(K)})$$

The assignments stabilize in finite time.

Part (iv) - Objective Convergence:

Let $\theta^{(k)} = (\xi^{(k)}, \psi^{(k)})$ denote the complete solution at iteration k, where $\xi^{(k)} = (X^{(k)}, Y^{(k)}, z^{(k)})$ are assignments and $\psi^{(k)} = (u^{(k)}, v^{(k)}, f^{(k)}, R^{(k)})$ are resource allocations.

Decompose the suboptimality:

$$\begin{aligned}J(\theta^{(k)}) - J^* &= J(\theta^{(k)}) - J(\theta^*) \\ &= \underbrace{[J(\xi^{(k)}, \psi^{(k)}) - J(\xi^{(k)}, \psi^*(\xi^{(k)}))]}_{\text{Term A: RAR suboptimality}} \\ &\quad + \underbrace{[J(\xi^{(k)}, \psi^*(\xi^{(k)})) - J(\xi^*, \psi^*(\xi^*))]}_{\text{Term B: TSA suboptimality}} \qquad \text{(A II-1)}\end{aligned}$$

where $\psi^*(\xi)$ denotes the optimal resource allocation for fixed assignment ξ.

Bound Term A (RAR Error):

In Stage 2, the RAR problem uses PWL approximation to linearize the nonlinear utility functions. Let J_2^N denote the linearized objective with N breakpoints, and let $\psi^{(k),N}$ be its optimal solution computed by the MILP solver.

Since the MILP is solved to global optimality for the linearized problem:

$$J_2^N(\xi^{(k)}, \psi^{(k),N}) \leq J_2^N(\xi^{(k)}, \psi^*(\xi^{(k)}))$$

By Lemma II, each utility function has approximation error $O(N^{-2})$. Since the objective (13) contains a constant number of utility function evaluations (proportional to $|\mathcal{T}| \cdot |\mathcal{R}| \cdot |\mathcal{S}|$), the total approximation error is:

$$|J(\xi, \psi) - J_2^N(\xi, \psi)| \leq C \cdot \frac{M(b-a)^2}{8N^2} =: \epsilon_N$$

for some constant C depending on the problem size and weight coefficients.

Therefore:

$$\begin{aligned} J(\xi^{(k)}, \psi^{(k)}) &\leq J_2^N(\xi^{(k)}, \psi^{(k)}) + \epsilon_N \\ &\leq J_2^N(\xi^{(k)}, \psi^*(\xi^{(k)})) + \epsilon_N \\ &\leq J(\xi^{(k)}, \psi^*(\xi^{(k)})) + 2\epsilon_N \end{aligned}$$

Thus: Term A $\leq 2\epsilon_N = O(N^{-2})$.

Bound Term B (TSA Error):

Algorithm 1 (greedy TSA) selects assignments based on estimates $\mathcal{E}^{(k)}$. The assignment quality depends on how close these estimates are to the true equilibrium.

By condition (d), the objective is L-Lipschitz in the resource variables. Since $\psi^*(\xi)$ is the optimal allocation for assignment ξ, we can write:

$$J(\xi^{(k)}, \psi^*(\xi^{(k)})) = \min_{\psi} J(\xi^{(k)}, \psi) =: \tilde{J}(\xi^{(k)})$$

The TSA decisions at iteration k depend on $\mathcal{E}^{(k)}$. By design, Algorithm 1 attempts to minimize the expected objective based on these estimates. While the greedy algorithm does not guarantee global optimality, the quality of its solution improves as estimates improve.

Assume the greedy TSA satisfies a stability property: there exists a constant $\kappa > 0$ such that:

$$\tilde{J}(\xi^{(k)}) - \tilde{J}(\xi^*) \leq \kappa \|\mathcal{E}^{(k)} - \mathcal{E}^*\|$$

This assumes that better estimates lead to better assignment decisions, which is reasonable for greedy algorithms that monotonically improve with better information Williamson & Shmoys (2011).

From Part (ii):

$$\|\mathcal{E}^{(k)} - \mathcal{E}^*\| \leq \beta^k \|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\sigma}{1-\beta}$$

Therefore:

$$\begin{aligned} \text{Term B} &= J(\xi^{(k)}, \psi^*(\xi^{(k)})) - J(\xi^*, \psi^*(\xi^*)) \\ &\leq \kappa \|\mathcal{E}^{(k)} - \mathcal{E}^*\| \\ &\leq \kappa \left[\beta^k \|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\sigma}{1-\beta} \right] \end{aligned}$$

Combining Both Terms:

From equation (A II-1):

$$\begin{aligned} J(\theta^{(k)}) - J^* &\leq \text{Term A} + \text{Term B} \\ &\leq 2\epsilon_N + \kappa\beta^k \|\mathcal{E}^{(0)} - \mathcal{E}^*\| + \frac{\kappa\sigma}{1-\beta} \end{aligned}$$

Setting $\Delta_0 = \kappa\|\mathcal{E}^{(0)} - \mathcal{E}^*\|$ (with κ absorbed into L for notational consistency) and $C_\infty = \frac{L\sigma}{1-\beta} + 2\epsilon_N$:

$$J(\theta^{(k)}) - J^* \leq \Delta_0\beta^k + C_\infty$$

The first term $\Delta_0\beta^k$ decays geometrically to zero, while C_∞ represents the irreducible steady-state error from measurement noise and PWL approximation. □

The convergence rate is characterized by the contraction constant $\beta = 1 - \lambda$. To achieve $J(\theta^{(k)}) - J^* \leq \epsilon + C_\infty$, we need:

$$\Delta_0\beta^k \leq \epsilon \implies k \geq \frac{\log(\epsilon/\Delta_0)}{\log(\beta)} = O\left(\frac{\log(1/\epsilon)}{\log(1/\beta)}\right)$$

This is logarithmic in the desired accuracy ϵ, demonstrating fast convergence. For example, with $\beta = 0.8$ (corresponding to $\lambda = 0.2$), reducing the error by a factor of 10 requires approximately $k \approx 10.3$ iterations.

The steady-state error C_∞ can be controlled by:

1. **Reducing measurement noise** (σ): Increase averaging window or improve measurement accuracy.
2. **Increasing PWL breakpoints** (N): Since $\epsilon_N = O(N^{-2})$, doubling N reduces linearization error by a factor of 4.
3. **Optimizing learning rate** (λ): Smaller λ reduces $\sigma/(1-\beta)$ but slows convergence (larger β).

Theorem II establishes that the proposed orchestration framework converges at a geometric rate to a solution whose quality is bounded by the steady-state error C_{∞}. This error can be made arbitrarily small by appropriate choice of system parameters, providing strong theoretical guarantees for the practical deployment of the O-FL rApp framework.

BIBLIOGRAPHY

3GPP. (2018). *NR; NR and NG-RAN Overall description; Stage-2* (Report n°TS 38.300).

3GPP. (2024, September). Technical Specification: 5G NG-RAN Architecture Description [Technical Specification TS 38.401 version 18.4.0 Release 18].

Alam, K., Habibi, M. A., Tammen, M., Krummacker, D., Saad, W., Renzo, M. D., Melodia, T., Costa-Pérez, X., Debbah, M., Dutta, A. & Schotten, H. D. (2025). A Comprehensive Tutorial and Survey of O-RAN: Exploring Slicing-Aware Architecture, Deployment Options, Use Cases, and Challenges. *IEEE Communications Surveys Tutorials*, 1-1. doi: 10.1109/COMST.2025.3598406.

Alliance, O.-R. (2021a, June). O-RAN Alliance Whitepaper: Minimum Viable Plan and Acceleration towards Commercialization [White Paper]. Retrieved from: https://www.o-ran.org/s/O-RAN-Minimum-Viable-Plan-and-Acceleration-towards-Commercialization-White-Paper-29-June-2021.pdf.

Alliance, O.-R. (2021b). Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN [Technical Specification WG6.CAD-V01.00.00]. Retrieved on 2021-03-17 from: https://www.o-ran.org.

Amiri, M. M., Gündüz, D., Kulkarni, S. R. & Poor, H. V. (2021). Convergence of Update Aware Device Scheduling for Federated Learning at the Wireless Edge. *IEEE Transactions on Wireless Communications*, 20(6), 3643-3658. doi: 10.1109/TWC.2021.3052681.

Atkinson, K. E. (1989). *An introduction to numerical analysis*. John Wiley & Sons.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H. B. et al. (2019). Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1, 374–388.

Boyd, S. & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Cao, Y., Lien, S.-Y., Liang, Y.-C., Chen, K.-C. & Shen, X. S. (2022). User access control in open radio access networks: A federated deep reinforcement learning approach. *IEEE Transactions on Wireless Communications*, 21(6), 3721–3736.

Chen, M., Yang, Z., Saad, W., Yin, C., Poor, H. V. & Cui, S. (2021a). A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 20(1), 269–283.

Chen, Y., Ning, Y., Slawski, M. & Rangwala, H. (2021b). Asynchronous federated learning with differential privacy for edge intelligence. *IEEE Transactions on Network Science and Engineering*.

Cheng, Z., Liwang, M., Xia, X., Min, M., Wang, X. & Du, X. (2022a). Auction-Promoted Trading for Multiple Federated Learning Services in UAV-Aided Networks. *IEEE Transactions on Vehicular Technology*, 71(10), 10960-10974. doi: 10.1109/TVT.2022.3184026.

Cheng, Z., Liwang, M., Xia, X., Min, M., Wang, X. & Du, X. (2022b). Auction-promoted trading for multiple federated learning services in UAV-aided networks. *IEEE Transactions on Vehicular Technology*, 71(10), 10960–10974.

Cho, Y. J., Wang, J. & Joshi, G. (2020). Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *AISTATS*.

Costa-Perez, X., Swetina, J., Guo, T., Mahindra, R. & Rangarajan, S. (2013). Radio access network virtualization for future mobile carrier networks. *IEEE Communications Magazine*, 51(7), 27–35.

Deng, L. (2012). The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6), 141-142. doi: 10.1109/MSP.2012.2211477.

Diamond, S. & Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83), 1–5.

Dinh, C. T., Tran, N. H., Nguyen, M. N., Hong, C. S., Sluganovic, I., Gong, W., Yang, K., Tsang, D. H. & Huh, E. N. (2021). Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Transactions on Networking*, 29(1), 398–409.

Duchi, J., Hazan, E. & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7).

Ericsson. (2021, July). Accelerating the adoption of AI in programmable 5G networks [White Paper]. Retrieved from: https://www.ericsson.com/4a3998/assets/local/reports-papers/white-papers/08172020-accelerating-the-adoption-of-ai-in-programmable-5g-networks-whitepaper.pdf.

Ericsson AB. (2023). *Ericsson Mobility Report June 2023*.

Feng, C., Yang, H. H., Hu, D., Zhao, Z., Quek, T. Q. & Min, G. (2022). Mobility-aware cluster federated learning in hierarchical wireless networks. *IEEE Transactions on Wireless Communications*, 21(10), 8441–8458.

Gao, Y., Zhao, Y. & Yu, H. (2023). Multi-Tier Client Selection for Mobile Federated Learning Networks. *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 666-671. doi: 10.1109/ICME55011.2023.00120.

Hamdan, M. Q., Lee, H., Triantafyllopoulou, D., Borralho, R., Kose, A., Amiri, E., Mulvey, D., Yu, W., Zitouni, R., Pozza, R. et al. (2023a). Recent advances in machine learning for network automation in the O-RAN. *Sensors*, 23(21), 8792.

Hamdan, M. Q., Lee, H., Triantafyllopoulou, D., Borralho, R., Kose, A., Amiri, E., Mulvey, D., Yu, W., Zitouni, R., Pozza, R., Hunt, B., Bagheri, H., Foh, C. H., Heliot, F., Chen, G., Xiao, P., Wang, N. & Tafazolli, R. (2023b). Recent Advances in Machine Learning for Network Automation in the O-RAN. *Sensors*, 23(21). doi: 10.3390/s23218792.

Horváth, S., Ho, C.-Y., Horváth, Ľ., Sahu, A. N., Canini, M. & Richtárik, P. (2022). Natural compression for distributed deep learning. *Mathematical and Scientific Machine Learning*, pp. 129–141.

ITU-R. (2015). *IMT Vision – Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond* (Report n°ITU-R M.2083-0).

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H. & Zhao, S. (2021). Advances and Open Problems in Federated Learning. Retrieved from: https://arxiv.org/abs/1912.04977.

Karmakar, R., Kaddoum, G. & Chattopadhyay, S. (2023). Mobility management in 5G and beyond: A novel smart handover with adaptive time-to-trigger and hysteresis margin. *IEEE Transactions on Mobile Computing*, 22(10), 5995–6010.

Kelly, F. P., Maulloo, A. K. & Tan, D. K. H. (1998). Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3), 237–252. doi: 10.1057/palgrave.jors.2600523.

Khaled, A. & Richtárik, P. (2019). Gradient descent with compressed iterates. *arXiv preprint arXiv:1909.04716*.

Kim, J., Kim, S., Hong, S., Kim, J., Jung, H.-k. & Yoo, H. (2021). A formally verified security scheme for Inter-gNB-DU handover in 5G vehicle-to-everything. *IEEE Access*, 9, 119100–119117.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Konečnỳ, J., McMahan, H. B., Ramage, D. & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*.

Koursioumpas, N. et al. (2024). Federated Reinforcement Learning for Wireless Networks: Fundamentals, Challenges and Future Research Trends. *IEEE Communications Surveys & Tutorials*.

Kreyszig, E. (1978). *Introductory functional analysis with applications*. John Wiley & Sons.

Krizhevsky, A., Nair, V. & Hinton, G. CIFAR-10 (Canadian Institute for Advanced Research). Retrieved from: http://www.cs.toronto.edu/~kriz/cifar.html.

Krouka, M., Elgabli, A., Issaid, C. B. & Bennis, M. (2022). Communication-Efficient and Federated Multi-Agent Reinforcement Learning. *IEEE Transactions on Cognitive Communications and Networking*, 8(1), 311-320. doi: 10.1109/TCCN.2021.3130993.

Lacava, A., Polese, M., Sivaraj, R., Soundrarajan, R., Bhati, B. S., Singh, T., Zugno, T., Cuomo, F. & Melodia, T. (2024). Programmable and customized intelligence for traffic steering in 5G networks using open RAN architectures. *IEEE Transactions on Mobile Computing*, 23(4), 2882–2897.

Lee, H., Song, Q. & Honorio, J. (2022). Support Recovery in Sparse PCA with Incomplete Data. *Advances in Neural Information Processing Systems*. Retrieved from: https://openreview.net/forum?id=x5ysKCMXR5s.

Lee, H., Cha, J., Kwon, D., Jeong, M. & Park, I. (2020). Hosting AI/ML workflows on O-RAN RIC platform. *IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6.

Li, L., Shi, D., Hou, R., Li, H., Pan, M. & Han, Z. (2021). To Talk or to Work: Flexible Communication Compression for Energy Efficient Federated Learning over Heterogeneous Mobile Edge Devices. *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1-10. doi: 10.1109/INFOCOM42981.2021.9488839.

Li, T., Sahu, A. K., Talwalkar, A. & Smith, V. (2020a). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3), 50–60.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A. & Smith, V. (2020b). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2, 429–450.

Li, Z., Kovalev, D., Qian, X. & Richtárik, P. (2020c). Acceleration for compressed gradient descent in distributed and federated optimization. *arXiv preprint arXiv:2002.11364*.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D. & Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3), 2031–2063.

Lin, F. P.-C., Hosseinalipour, S., Michelusi, N. & Brinton, C. G. (2024). Delay-Aware Hierarchical Federated Learning. *IEEE Transactions on Cognitive Communications and Networking*, 10(2), 674-688. doi: 10.1109/TCCN.2023.3329024.

Liu, J., Xu, H., Wang, L., Xu, Y., Qian, C., Huang, J. & Huang, H. (2023). Adaptive Asynchronous Federated Learning in Resource-Constrained Edge Computing. *IEEE Transactions on Mobile Computing*, 22(2), 674-690. doi: 10.1109/TMC.2021.3096846.

Liu, K. (2021). 5G User prediction data [Online dataset]. Retrieved on 2021-03-17 from: https://www.kaggle.com/liukunxin/dataset.

Liu, S., Yu, G., Chen, X. & Bennis, M. (2022). Joint user association and resource allocation for wireless hierarchical federated learning with IID and non-IID data. *IEEE Transactions on Wireless Communications*, 21(10), 7852–7866.

Liu, W., Chen, L., Chen, Y. & Zhang, W. (2020). Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 31(8), 1754–1766.

Lu, Y., Huang, P., Liu, Z., Liu, D. & Zhang, H. (2022). Federated Multi-Agent Deep Reinforcement Learning for Resource Management in Vehicular Edge Computing. *IEEE Transactions on Vehicular Technology*, 71(7), 7854-7866. doi: 10.1109/TVT.2022.3168853.

Luo, B., Han, P., Sun, P., Ouyang, X., Huang, J. & Ding, N. (2023). Optimization Design for Federated Learning in Heterogeneous 6G Networks. *IEEE Network*, 37(2), 38-43. doi: 10.1109/MNET.006.2200437.

Luo, S., Chen, X., Wu, Q., Zhou, Z. & Yu, S. (2020). HFEL: Joint Edge Association and Resource Allocation for Cost-Efficient Hierarchical Federated Edge Learning. *IEEE Transactions on Wireless Communications*, 19(10), 6535-6548. doi: 10.1109/TWC.2020.3003744.

Ma, C., Konečný, J., Jaggi, M., Smith, V., Jordan, M. I., Richtárik, P. & Takáč, M. (2017). Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4), 813–848. doi: 10.1080/10556788.2016.1278445.

Mao, Y., You, C., Zhang, J., Huang, K. & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358.

Marinova, S. & Leon-Garcia, A. (2024). Intelligent O-RAN beyond 5G: Architecture, use cases, challenges, and opportunities. *IEEE Access*, 12, 27088–27114.

Maternia, M. et al. (2016). *5G PPP use cases and performance evaluation models*.

McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics*, pp. 1273–1282.

Mo, X. & Xu, J. (2021). Energy-efficient federated edge learning with joint communication and computation design. *Journal of Communications and Information Networks*, 6(2), 110–124.

Munkres, J. R. (2000). *Topology*. Prentice Hall.

Nasir, Y. S. & Guo, D. (2019). Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 37(10), 2239-2250. doi: 10.1109/JSAC.2019.2933973.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate O (1/kˆ 2). *Proceedings of the USSR Academy of Sciences*, 269, 543–547.

Nguyen, M. N. H., Tran, N. H., Tun, Y. K., Han, Z. & Hong, C. S. (2023a). Toward Multiple Federated Learning Services Resource Sharing in Mobile Edge Networks. *IEEE Transactions on Mobile Computing*, 22(1), 541-555. doi: 10.1109/TMC.2021.3085979.

Nguyen, M. N., Tran, N. H., Tun, Y. K., Han, Z. & Hong, C. S. (2023b). Toward multiple federated learning services resource sharing in mobile edge networks. *IEEE Transactions on Mobile Computing*, 22(1), 541–555.

Nguyen, T. T., Nguyen, N. D. & Nahavandi, S. (2020). Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics*, 50(9), 3826-3839. doi: 10.1109/TCYB.2020.2977374.

Nishio, T. & Yonetani, R. (2019). Client selection for federated learning with heterogeneous resources in mobile edge. *IEEE International Conference on Communications (ICC)*, pp. 1–7.

Nocedal, J. & Wright, S. J. (2006). *Numerical optimization*. Springer.

Noor-A-Rahim, M., Liu, Z., Lee, H., Khyam, M. O., He, J., Pesch, D., Moessner, K., Saad, W. & Poor, H. V. (2022). A survey on resource allocation in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(2), 701–721.

O-RAN Alliance. (2019). *O-RAN architecture description* (Report n°O-RAN.WG1.O-RAN-Architecture-Description-v02.00).

O-RAN Alliance. (2020). *AI/ML workflow description and requirements* (Report n°O-RAN.WG2.AIML-v01.00).

O-RAN Alliance. (2024a). *O-RAN Architecture Description* (Report n°O-RAN.WG1.O-RAN-Architecture-Description).

O-RAN Alliance. (2024b). *O-RAN Near-Real-Time RAN Intelligent Controller Architecture* (Report n°O-RAN.WG3.Near-RT-RIC-ARCH).

O-RAN Alliance. (2024c). *74 New or Updated O-RAN Technical Documents Released since July 2024*.

O-RAN Software Community. [Accessed: Nov. 3, 2025]. (2025). O-DU High Overview. O-RAN Master Documentation. Retrieved from: https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-l2/en/latest/overview.html#o-du-high-interfaces.

O-RAN White Paper. (2023). *O-RAN Global Operator Commitments*.

O-RAN.WG1. (2023). *O-RAN Empowering Vertical Industry: Scenarios, Solutions and Best Practice*.

Polese, M., Bonati, L., D'Oro, S., Basagni, S. & Melodia, T. (2023a). Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys & Tutorials*, 25(2), 1376–1411.

Polese, M., Bonati, L., D'Oro, S., Basagni, S. & Melodia, T. (2023b). Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys & Tutorials*, 25(2), 1376-1411. doi: 10.1109/COMST.2023.3239220.

Prananto, B. H., Iskandar & Kurniawan, A. (2023). A new method to improve frequent-handover problem in high-mobility communications using RIC and machine learning. *IEEE Access*, 11, 72281–72294.

Qin, Z., Li, G. Y. & Ye, H. (2021). Federated Learning and Wireless Communications. *IEEE Wireless Communications*, 28(5), 134–140.

Qureshi, H. N., Manalastas, M., Zaidi, S. M. A., Imran, A. & Al Kalaa, M. O. (2021). Service level agreements for 5G and beyond: Overview, challenges and enablers of 5G-healthcare systems. *IEEE Access*, 9, 1044–1061.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S. & McMahan, H. B. (2020). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A. & Pedarsani, R. (2020). FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. *arXiv preprint arXiv:1909.13014*.

Rojas, J. S. [Accessed: April 2, 2025]. (2020). IP Network Traffic Flows Labeled with 87 Apps. Retrieved from: https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps.

Rudin, W. (1976). *Principles of mathematical analysis*. McGraw-Hill New York.

Shi, W., Li, J., Cheng, N., Liu, T., Lu, W. & Shen, X. (2022). Federated Multi-Agent Reinforcement Learning for Joint Trajectory Design and Resource Management in the Sky of Everything. *IEEE Journal on Selected Areas in Communications*, 40(5), 1599-1614. doi: 10.1109/JSAC.2022.3160913.

Shi, W., Zhou, S., Niu, Z., Jiang, M. & Geng, L. (2021). Joint Device Scheduling and Resource Allocation for Latency Constrained Wireless Federated Learning. *IEEE Transactions on Wireless Communications*, 20(1), 453-467. doi: 10.1109/TWC.2020.3025446.

Singh, A. K. & Khoa Nguyen, K. (2025). User Handover Aware Hierarchical Federated Learning for Open RAN-Based Next-Generation Mobile Networks. *IEEE Transactions on Machine Learning in Communications and Networking*, 3, 848-863. doi: 10.1109/TMLCN.2025.3587205.

Singh, A. K. & Nguyen, K. K. (2022a). MCORANFed: Communication Efficient Federated Learning in Open RAN. *2022 14th IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 15-22. doi: 10.23919/WMNC56391.2022.9954307.

Singh, A. K. & Nguyen, K. K. (2022b). Joint Selection of Local Trainers and Resource Allocation for Federated Learning in Open RAN Intelligent Controllers. *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1874–1879. doi: 10.1109/WCNC51071.2022.9771700.

Smith, V., Chiang, C.-K., Sanjabi, M. & Talwalkar, A. S. (2017). Federated multi-task learning. *Advances in Neural Information Processing Systems*, 30.

Sun, Y., Peng, M., Ren, Y., Chen, L., Yu, L. & Suo, S. (2021). Harmonizing Artificial Intelligence with Radio Access Networks: Advances, Case Study, and Open Issues. *IEEE Network*, 35(4), 144-151. doi: 10.1109/MNET.011.2000656.

Sutskever, I., Martens, J., Dahl, G. & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *International Conference on Machine Learning*, pp. 1139–1147.

Vandenberghe, L. & Boyd, S. (1996). Semidefinite Programming. *SIAM Review*, 38(1), 49-95. doi: 10.1137/1038003.

Wang, C., Yao, T., Fan, T., Peng, S., Xu, C. & Yu, S. (2024a). Modeling on Resource Allocation for Age-Sensitive Mobile-Edge Computing Using Federated Multi-agent Reinforcement Learning. *IEEE Internet of Things Journal*, 11(2), 3121-3131. doi: 10.1109/JIOT.2023.3294535.

Wang, J., Chen, P., Wang, J. & Yang, B. (2024b). A Hierarchical Federated Learning Paradigm in O-RAN for Resource-Constrained IoT Devices. *ICC 2024 - IEEE International Conference on Communications*, pp. 2555-2560. doi: 10.1109/ICC51166.2024.10623078.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T. & Chan, K. (2019a). Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6), 1205–1221.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T. & Chan, K. (2020). Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6), 1205–1221.

Wang, Y., Zhu, S. & Li, C. (2019b). Research on Multistep Time Series Prediction Based on LSTM. *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 1155-1159. doi: 10.1109/EITCE47263.2019.9095044.

Williamson, D. P. & Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge University Press.

Wolsey, L. A. (2020). *Integer programming*. John Wiley & Sons.

Xie, C., Koyejo, S. & Gupta, I. (2019). Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*.

Xu, J. & Wang, H. (2021). Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Transactions on Wireless Communications*, 20(2), 1188–1200.

Xu, J., Wang, H. & Chen, L. (2022a). Bandwidth allocation for multiple federated learning services in wireless edge networks. *IEEE Transactions on Wireless Communications*, 21(4), 2534–2546.

Xu, J., Wang, H. & Chen, L. (2022b). Bandwidth Allocation for Multiple Federated Learning Services in Wireless Edge Networks. *IEEE Transactions on Wireless Communications*, 21(4), 2534-2546. doi: 10.1109/TWC.2021.3113346.

Xu, Y., Qian, B., Yu, K., Ma, T., Zhao, L. & Zhou, H. (2023a). Federated Learning Over Fully-Decoupled RAN Architecture for Two-Tier Computing Acceleration. *IEEE Journal on Selected Areas in Communications*, 41(3), 789-801. doi: 10.1109/JSAC.2023.3236003.

Xu, Z., Luo, Y., Wu, F., Zhang, X., Yang, L., Song, M., Song, G. & Shen, S. (2023b). HierFedML: Aggregator placement and UE assignment for hierarchical federated learning in mobile edge computing. *IEEE Transactions on Parallel and Distributed Systems*, 34(1), 328–345.

Yang, H. H., Liu, Z., Quek, T. Q. & Poor, H. V. (2020). Scheduling policies for federated learning in wireless networks. *IEEE Transactions on Communications*, 68(1), 317–333.

Yang, Z., Chen, M., Saad, W., Hong, C. S. & Shikh-Bahaei, M. (2021). Energy efficient federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications*, 20(3), 1935–1949.

Yu, S., Wang, X. & Luan, T. H. (2023a). Federated multi-agent reinforcement learning for joint task offloading and content caching in vehicular edge computing. *IEEE Transactions on Wireless Communications*, 22(4), 2390–2404.

Yu, S., Wang, X. & Luan, T. H. (2023b). Federated Multi-Agent Reinforcement Learning for Joint Task Offloading and Content Caching in Vehicular Edge Computing. *IEEE Transactions on Wireless Communications*, 22(4), 2390-2404. doi: 10.1109/TWC.2022.3211532.

Zhan, Y., Guo, S., Li, P. & Zhang, J. (2022). A Survey on Deep Reinforcement Learning for Task Offloading in Edge Computing. *ACM Comput. Surv.*, 55(6). doi: 10.1145/3543823.

Zhao, Z., Feng, C., Yang, H. H., Luo, X., Quek, T. Q. & Min, G. (2020). Federated-learning-enabled intelligent fog radio access networks: Fundamental theory, key techniques, and future trends. *IEEE Wireless Communications*, 27(2), 22–28.

Zhao, Z., Li, A., Li, R., Yang, L. & Xu, X. (2025). FedPPO: Reinforcement learning-based client selection for federated learning with heterogeneous data. *IEEE Transactions on Cognitive Communications and Networking*. Early Access.

Zhou, H., Zheng, Y., Huang, H., Shu, J. & Jia, X. (2023). Toward Robust Hierarchical Federated Learning in Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(5), 5600-5614. doi: 10.1109/TITS.2023.3243003.

Zhu, G., Liu, D., Du, Y., You, C., Zhang, J. & Huang, K. (2019). Broadband analog aggregation for low-latency federated edge learning. *IEEE International Conference on Communications (ICC)*, pp. 1–6.

Zhuansun, Y. et al. (2024). Communication-Efficient Federated Learning with Adaptive Compression under Dynamic Bandwidth. *arXiv preprint arXiv:2405.03248*.