RAT Selection Algorithm for Multiservice Multimode Terminals in Next Generation Wireless Networks

Prepared by:
Bina E. Mukuyamba
MKYBIN001
Department of Electrical Engineering
University of Cape Town

Prepared for:
Associate Professor Olabisi Falowo
Department of Electrical Engineering
University of Cape Town

October 2023

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Electrical and Computer Engineering

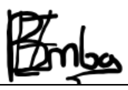
Key Words: Multimode, Multiservice, Heterogenous, RAT, Wireless Networks, 5G

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.
3. This final year project report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof

Word count : 18 171

Name: **Bina E. Mukuyamba**

Signature: 

Date: 30 October 2023

Terms of Reference

Student proposed?	*Y/N* ***N***	*If Y, student name*
ID:	**OF-7**	
SUPERVISOR:	Olabisi Falowo	
TITLE:	RAT Selection Algorithm for Multiservice Multimode Terminals in Next Generation Networks	
DESCRIPTION:	Most multimode terminals simultaneously support multiple services (such as voice, video streaming, and web browsing). However, most of the network selection algorithms proposed for vertical handoff in the literature are only suitable for mobile terminals running a single service. It is possible for multiples services from a multimode terminal to be handed over to a single or multiple RATs. The objective of this project is to develop a network selection algorithm for multiservice multimode terminals in heterogeneous wireless networks. The algorithm will consider user preferences, network attributes, and service characteristics in making vertical handoff decisions. These attributes will be combined to make vertical handoff decisions using the fuzzy analytic hierarchy process.	
DELIVERABLES:	A review of vertical handoff algorithms, implemented vertical handoff algorithm, simulation results, and report.	
SKILLS/REQUIREMENTS:	MATLAB or any other programming language, EEE4121F.	
GA1: Problem solving: *Identify, formulate, analyse and solve complex* engineering problems creatively and innovatively*	The student is expected to design and implement an algorithm for making vertical handover decisions for multiple services in heterogeneous wireless networks.	
GA 4**: Investigations, experiments and analysis: *Demonstrate competence to design and conduct investigations and experiments.*	The student is expected to investigate the performance of RAT selection algorithm.	
EXTRA INFORMATION:	For a student interested in pursuing a master's degree, the project can be expanded to an MSc dissertation.	
BROAD Research Area:	Wireless Networks	
ETHICS	The project does not require the use of human subject or animal.	
Project suitable for ME/ ECE/EE/All?	EE/ECE students who have taken EEE4121F course.	

Acknowledgements

I would like to first thank my mother and family for providing financial and emotional support over the course of my degree. Without her tremendous contribution, I would not be where I am today.

I would also like to thank my supervisor, Associate Professor Olabisi E. Falowo for all the support and encouragement I received throughout the project as well as for providing the necessary education I needed to understand the concepts of networks over the last 2 years.

I would like to thank my friends and classmates for all the support throughout my studies at UCT and for encouraging me to always stay motivated and follow my dreams. I also want to say thank you to all the teaching staff in the electrical engineering department at UCT for instructing and guiding me throughout the years, nurturing my passion for the field of Electrical Engineering.

Finally, I would like to thank God almighty for the opportunity to write this report and for seeing me through all the years I've been at UCT.

Abstract

Over the past decade, there has been a global increase in the number of mobile terminal users as communication technology continues to evolve at a rapid rate. These mobile devices/terminals can support multiple generations of Radio Access Technologies (RATs) and can run multiple applications (services) simultaneously. There is therefore a need for appropriate network selection algorithms that can select the best network for these devices. This will provide the best quality of service possible and ensure network resources are managed efficiently. Currently, most existing algorithms in the literature only consider mobile devices that run a single service, the few that do consider devices that run multiple services do not consider service characteristics, user preferences and network attributes effectively.

Therefore, in this report, a RAT selection algorithm for multiservice multimode terminals (MMTs) in next-generation wireless networks has been proposed. The algorithm considers selection factors including user preferences, service characteristics, and network attributes, to determine the optimal RAT for multiple running services on an MMT. The algorithm uses several multi-attribute decision making (MADM) techniques such as Entropy and (Fuzzy analytic hierarchy process (FAHP) to determine the comprehensive network attribute weights. These weights are combined with the comprehensive utility value matrix for multiple services to create a weighted normalized decision matrix. TOPSIS is used to calculate the scores of each candidate RAT and rank them based on this matrix to determine the optimal RAT. The contribution of each selection factor to the comprehensive weights of network attributes is determined by weight proportion parameters (α, β, γ) and a threshold (σ) is introduced to minimize the number of unnecessary handovers.

A heterogeneous wireless network environment was modelled and consisted of four RATs namely 4G-LTE, 3G-UMTS, WLAN and 5G, which all supported a group of 3 services (voice, video streaming and web browsing) that were simultaneously running on the mobile devices in the network. The selection criteria used by the algorithm in the network included bandwidth, cost, delay, and packet loss rate. Several experiments were conducted to evaluate the performance of the algorithm. These experiments were simulated in MATLAB for 1000 users using this network model.

The gain of the network for different weight parameter combinations was investigated to find an optimal combination, the best one was $\alpha = 0.2, \beta = ,0.5$ and $\gamma = 0.3$. The performance of the algorithm for varying service priority scenarios was investigated. (4G-LTE) was the most selected RAT though WLAN occasionally had more selections. Finally, the effect of the threshold on the number of handovers was investigated. For $\sigma = 1.3$, the number of handovers increased as the number of users increased at a gentle rate. As the threshold increased the number of handovers decreased and the percentage of unnecessary handovers was always less than 20% for 1000 users.

Overall, the algorithm was able to select the optimal RAT for a collection of services on the MMT because it successfully considered user preferences, service characteristics and network attributes. The use of a threshold effectively minimized the number of handovers and the percentage of unnecessary handovers.

Table of Contents

List of Tables

List of Figures

Acronyms

ABC – Always Best Connected
ADR – Aggregated Data Rate
AHP – Analytic Hierarchy Process
AP – Access Point
BD – Best Downlink
BER – Bit Error Rate
BGMNS – Bipartite Graph Matching Network Selection Algorithm
BS – Base Station
CRITIC – Criteria Importance Through Intercriteria Correlation
DASA – Dynamic radio Access selection and Slice Allocation
DIA – Distance to Ideal Alternative
ELECTRE – The Elimination and Choice Expressing Reality
eNB – enhanced Node B
FAHP – Fuzzy Analytic Hierarchy Process
FCM – Fuzzy Comparison Matrix
GDM – Group Decision Making
GRA – Grey Relational Analysis
GSM – Global System for Mobile Communications
HWN – Heterogeneous Wireless Networks
IF – Intuitionistic Fuzzy
LEO – Low Earth Orbit
LTE – Long Term Evolution
MAC – Media Access Control
MADM – Multi-Attribute Decision Making
MCDM – Multi-Criteria Decision Making
MCGDM – Multi-Criteria Group Decision Making
MEW – Multiplicative Exponential Weighting
MIHO – Multi-objective Intelligent HandOver scheme
MMT – Multiservice Multimode Terminal
MT – Mobile Terminal
MULTIMOORA – MULTlplicative–form with Multi-Objective Optimization Ratio Analysis
NFPSO – Neuro-Fuzzy and Particle Swarm Optimization scheme
NIS – Negative Ideal Solution
NR – New Radio
PIS – Positive Ideal Solution
PSO – Particle Swarm Optimization
QoE – Quality of Experience
QoS – Quality of Service
RAT – Radio Access Technology
RSS – Received Signal Strength
SAW – Simple Additive Weighing
SIM – Subscriber Identity Module
SINR – Signal Interference to Noise Ratio
STIN – Satellite Terrestrial Integrated Networks
TFN – Triangular Fuzzy Number
TOPSIS – Technique for Order Preference by Similarity to an Ideal Solution
UAV – Unmanned Aerial Vehicle
UMTS – Universal Mobile Telecommunications System
VHO – Vertical Handover
VN – Virtual Network
WIFI – Wireless Fidelity
WiMAX – Worldwide Interoperability for Microwave Access
WLAN – Wireless Local Area Network

1. Introduction

1.1 Background to the study

Communication is a fundamental aspect of today's society as it is through communication that information is exchanged, relationships are built, and knowledge is shared. Over the past few years, there has been a global increase in the number of mobile device users (subscribers) [1]. This is no surprise as radio access technologies (RATs) continue to evolve to provide lower latencies, higher bandwidths, and throughputs. As a result, there is an increasing demand for appropriate network resource management algorithms to cater for the increasing load of mobile users on wireless networks. In doing so, mobile subscribers will be provided with the best quality of service (QoS) possible and have ubiquitous coverage. On the operator side, costs will be reduced as the network resources will be managed more efficiently for optimal operation.

1.2 Objectives of this study

1.2.1 Problems to be investigated

The study investigates vertical handoff algorithms for multiservice multimode mobile terminals in next-generation wireless networks. Currently, in the literature, most of the existing RAT selection algorithms for vertical handoff are only suited to single-service multi-mode mobile terminals (mobile terminals running a single service). Therefore, there is a need to design and implement a RAT selection algorithm for mobile terminals running multiple services simultaneously.

1.2.2 Purpose of the study

The purpose of the study is to ultimately design and implement a vertical handoff algorithm suitable for multiservice multimode mobile terminals. This is a relevant study to undertake because there needs to be a balance between meeting user preferences, service requirements and network characteristics. This ensures users have the best possible quality of service (QoS) and coverage according to the always best-connected principle (ABC) and that network operators can manage network resources efficiently.

1.3 Scope and Limitations

The focus of this investigation is strictly on vertical handoff algorithms. Older access technologies such as GSM (2G) are not covered in the report as technological advancements in mobile communications have rendered them less popular. Likewise, 6G is not covered as it is still being researched and is not yet commonplace. The main limitation is that the algorithm has been implemented in simulation only and hasn’t been deployed on an actual MMT.

1.4 Plan of development

The rest of this paper has been sectioned as follows, section 2 provides a literature review discussing the principles of RAT selection and multi-criteria decision making. It also discusses the related work that has been undertaken on the topic. In section 3 the network model and algorithm design are presented. A numerical example is also provided to illustrate how the algorithm works. The section also outlines the experimental setup and simulation scenarios used to evaluate the performance of the algorithm. The results of the simulations are provided in section 4. A discussion of the results is presented in section 5. Conclusions drawn from the discussion are highlighted in section 6 and finally, section 7 provides recommendations to improve the investigation in future.

2. Literature Review

2.1 Next-generation wireless networks

Next-generation wireless networks are the evolution of heterogeneous wireless networks that incorporate multiple access networks on a common service platform [2]. In a heterogenous wireless network several RATs coexist within the same area and their coverage regions overlap. These networks provide users with seamless connectivity and the ability to access different services through a single terminal. The selection of the appropriate Radio Access Technology for multiservice multimode terminals in next-generation networks is crucial for ensuring enhanced network performance and improved user experience. Some examples of current next-generation wireless networks are as follows.

Firstly 5G (fifth generation) is the newest generation of wireless networks. Key features are significantly high data rates (in Gbps), low latency (1ms), network slicing using VN(Virtual networks), small cell densification, new air interface and multimode terminal connectivity. 5G networks use millimetre waves(mmWave), which allow for higher bandwidth. 5G has many applications in industry such as massive IoT, mission-critical services and enhanced mobile broadband.

6G (sixth generation) is the next generation of wireless networks that will succeed 5G and is predicted for use in 2030. The key performance indicators of this network would be much lower latency than 5G(0.1ms) and much higher rates (1Tbps). It will provide sensing as an additional service to the services provided by 5G. 6G is envisioned to have very large coverage including space and air by integrating 5G with non-terrestrial network networks such as satellite networks.

2.2 Multiservice and multimode terminals

Mobile terminals(mobile stations) are devices that are a combination of terminal equipment and subscriber data. The subscriber data is contained in the Subscriber identity module (SIM) card which contains info such as the ID number of the user and facilities for authentication.

Multiservice terminals refer to mobile terminals that can support multiple services and applications such as voice, video streaming and web browsing simultaneously.

Multi-mode terminals refer to mobile terminals that can connect to and operate on multiple generations of radio access technologies e.g., 3G to 5G.

As most modern-day mobile terminals are both multiservice and multimode, the RAT selection problem for an MMT is a GDM (Group decision-making) problem as a group of services simultaneously running on the MMT needs to be handed off to one or more appropriate RATs.

2.3 Vertical handoff and Radio resource management

Handoff is the process of switching the channel associated with the current connection while a call is in progress. It is a means of achieving continuous service for a user moving across a network cell and is how mobility is realised in mobile and wireless cellular networks. Vertical handoff refers to handing over calls of a mobile terminal between base stations/access points that support different network technologies e.g., between a 2G base station and 4G eNB access point. Whereas horizontal handoff is the handing over of calls of a mobile terminal between base stations that support the same network technology. Horizontal handoff is conducted for homogenous networks to achieve continuous coverage. Next generation networks are becoming increasingly heterogeneous in nature therefore there is increased research towards vertical handover algorithms.

For vertical handoff, many characteristics of the network are used to determine whether a handover should occur. Typical characteristics include Quality of Service (QoS), cost of service, security, power requirements and velocity. Several criteria are derived from these characteristics and are used in vertical handover algorithms to select the appropriate RAT.

Vertical Handoff algorithms facilitate handoff between different types of wireless access technologies in a heterogeneous wireless network . This provides an uninterrupted connection(continuous service) for users that move between access points across cells. The algorithm determines the best network based on one or more RAT selection criteria such as cost, delay, jitter, available bandwidth and so on. There are multiple kinds of vertical handover algorithms, and each uses different criteria and mathematical techniques to make a handoff decision. Examples of vertical handover algorithms include utility function and cost function algorithms, computational intelligence algorithms (using fuzzy logic), weighted sum algorithms and machine learning-based algorithms. These will be discussed in detail below.

2.4 A review of Vertical handoff algorithms

Traditional handover decision methods used in horizontal handover such as RSS(Received signal strength) are insufficient for making vertical handovers between RATs because they don't take user preferences and the current context into account [3]. Some more complex considerations and trade-offs need to be considered when making vertical handover decisions because each RAT in a heterogeneous wireless network is designed for different services which each have different QoS requirements. Users in heterogeneous wireless networks have a broader and diverse range of QoS and application requirements.

There are 5 main categories of vertical handover algorithms that are used to achieve the always best-connected principle (ABC) for optimal network selection in heterogeneous wireless networks. However, it should be noted that most of the algorithms designed in the literature are combinations of 2 or more categories of vertical handover algorithms and some of the categories are derived from the other categories.

Firstly, there are cost function-based algorithms that choose the best network by using a mathematical function that involves the sum of weighted functions of network parameters according to policies (can be user-specified.)

Secondly, there are user-centric algorithms. These algorithms primarily take user preferences into account in terms of cost and QoS. These algorithms are designed for user satisfaction and are suitable for non-real-time applications such as file transfer.

Thirdly and most used are Multiple Attribute/Multi-Criteria decision-making algorithms. The vertical handover problem is modelled as a Multi-Attribute Decision Making (MADM) problem and involves selecting the optimal RAT from a limited number of candidate networks concerning different attributes/criteria. A multi-criteria decision-making technique (MCDM) determines the most ideal alternative from a given set of candidates based on multiple decision criteria [4]. Popular approaches for this include AHP (The Analytic Hierarchy Process), multiplicative exponential weightings (MEW), ELECTRE (The Elimination and Choice Expressing Reality), Simple Additive Weighting (SAW) and Technique for order preference by similarity to ideal solution(TOPSIS)[4-6]. MCDM techniques provide a mathematical approach to critically evaluate the alternative options using selection criteria, thereby taking into account factors which impact the network's performance and user's QoS. Table 1 below summarises some popular MADM techniques.

Table 1: Common MADM approaches [4-6]

Strategy	Explanation	Pros	Cons
AHP	Computes the relative weights of different variables in a decision prototype	Calculates the similarity to the ideal solution	Results not always consistent
MEW	Calculates network variable levels by multiplication	Value is directly related to selection of preferred alternatives	Insensitive to dynamic parameters
TOPSIS	Best alternative is closest to ideal alternative and furthest from worst alternative	The best and worst alternatives are calculated using the corresponding best and worst variable values	Sensitive to network parameters and user's preferences
ELECTRE	Decision maker satisfaction is measured according to consistency	Uses multiple strategies for pairwise comparisons under a standard	Relations that are outranked could be unfinished

Fuzzy logic and neural network-based algorithms are another category of vertical handover algorithms. These algorithms are typically combined with the above MADM methods to develop more advanced algorithms for both real-time (e.g., video calls) and non-real-time (e.g., file transfer) applications. They are used to handle the imprecision and uncertainties that are found in decision criteria. For example, criteria such as network security level are described in linguistic terms rather than numerically. This introduces imprecision because terms such as low, high, and medium don't have specific numbers attached to them. Therefore, Fuzzy logic is used to eliminate the uncertainty associated with the linguistic description of the criterion by converting the words to crisp values that can be used in the algorithm.

Neural network-based algorithms use machine learning and need to be configured using prior knowledge of the network environment and are usually responsible for choosing the time to handover whereas fuzzy logic is used to choose the most appropriate RAT according to user preferences.

Finally, context-aware algorithms are algorithms that use the contextual information from the mobile terminal and network to make intelligent network selections. The contextual information used includes the decision criteria linked to the mobile terminal such as its location and speed, and criteria linked to the network itself e.g., QoS and service type. These algorithms satisfy primary objectives based on the values of context parameters. However, it should be noted that these algorithms have increased overhead in radio links due to frequent communication between the MT and the network.

2.5 Related Work

In the literature, there are very few works that address the RAT selection problem for multiple calls from an MMT. Most of the algorithms that have been developed in the literature are only suitable for vertical handoff of a single session/service from a mobile terminal. Below is a review of the most recent works that address RAT selection for both single and group calls from an MMT.

H. Yu, Y. Ma, and J. Yu in [7] proposed a RAT selection algorithm that considers user preferences, service characteristics and network attributes. They developed a RAT selection algorithm for group call handover from a mobile terminal. They used Entropy and fuzzy AHP to calculate the objective weights of service characteristics and network attributes. User preferences and service priority were used to obtain the comprehensive network attribute weights. They used utility functions to calculate network attribute utility values for 3 services. The SAW technique was used to synthesize utility values and comprehensive weights. Then TOPSIS and a threshold were used to select the best network for the group of calls.

MATLAB was used for their network model simulation and included services such as voice, video streaming, and web browsing. The RATs used in their simulation included “UMTS, LTE, WLAN and WiMAX”. The network attribute parameters used were “bandwidth, delay, jitter, loss rate and cost”. They also used weight proportion parameter values set by the network operator according to policies to vary how much user preferences, service characteristics and network attributes contributed to the RAT selection algorithm. They conducted three types of simulation. Firstly, they simulated network selection at a single instance of time, then simulated multiple network selections in different group service priority situations and finally compared their proposed algorithm with two existing MMT(Fuzzy MCDGM and utility GDM) algorithms in a dynamic simulation environment.

Key findings from their experiments were that their proposed algorithm avoided making unnecessary network selections due to users having too subjective preferences (simulation 2). For the comparison between the other two MMT algorithms, the metrics used to compare the algorithm’s performances included candidate network selection probability, the gain of network selection, the number of handovers and their probabilities. Results showed that their algorithm took the user preferences, service characteristics and network conditions into account. So, WLAN and 3G had similar results.

In terms of the gain, the proposed algorithm wasn’t the best for a single attribute e.g., it had the highest average loss rate of the three algorithms, but the overall gain was higher than the other two MMT algorithms according to how they defined gain since it considers user preferences and service requirements. The proposed algorithm had the least number of handoffs and the lowest percentage of unnecessary handoffs because of the threshold used. They concluded that the proposed algorithm reduced the ping-pong effect, leading to better user experience and less resource wastage. Their results also showed that the proposed algorithm was stable when new networks were added as the probability of unnecessary handoffs remained roughly the same.

The advantages of their algorithm were that it managed to consider user preferences, service characteristics and network attributes(conditions) for network selection resulting in an algorithm that isn't too objective or subjective thereby providing users and services with an adequate QoS. Also, despite having $O(N^3)$ time complexity the number of RATs and criteria considered by the algorithm are very small (usually less than 10) therefore the computational demands on the MT are kept minimal in addition to the MT already having strong computing power. Furthermore, the spatial complexity of the algorithm is also kept at a minimum because the allocation of weights and priorities for the network attributes are only done once, and these values are stored on the MT.

Their only drawback was that they used static weight proportion parameters for simulation. These parameters were not optimal. They also only considered network selection for a single MMT at a time. Thus, load balancing between multiple MMTs, power consumption and mobility (moving speed) were not factored into the simulation. Also, their algorithm only selected a single RAT for group call handoff from an MT and did not consider making independent call decisions to select multiple RATs.

In [8] Falowo and Chan proposed dynamic RAT selection for multiple calls from an MT using a modified TOPSIS-GDM algorithm. It addressed the issue of vertical handover for multiple calls from a mobile terminal. The algorithm considered user preferences according to call class by having users specify the weights of each criterion for each service. They treated the problem as an MCGDM problem where the alternatives were RATs, the criteria were network attributes, and the decision makers were the ongoing calls/services from an MT. It was made dynamic by varying the number of available RATs and calls with time. The algorithm was triggered when the number of calls changed or when the RAT availability changed.

To evaluate the performance of their algorithm, they measured the proportion of calls admitted into each RAT under different scenarios where the weights of the criteria and service priorities were varied. To minimize vertical handovers, they introduced a RAT preference margin where the preference margin is how much better the new RAT must be than the current RAT before the newly preferred RAT can be selected for the calls from the MT. In general, the higher the preference margin was, the lower the frequency of vertical handovers. However, if this value was too high the new RAT would seldom be selected and the less preferred RAT would be selected most of the time. Results showed that the vertical handoff likelihood decreased as the number of current calls on the MT increased.

The advantage of their approach was that selecting a single RAT for the group of calls eliminated the issue of having to coordinate the handover if multiple RATs were selected for each call in the group(less overhead).

The disadvantage of their approach was that their algorithm only considered user preferences and network attributes so did not account for service characteristics, so it was too subjective.

Falowo and Taiwo [9] proposed a network selection algorithm based on consensus level. Consensus-based Multi Criteria GDM(MCGDM) was used to reach a mutual agreement between decision-makers before a decision was made to enhance the quality of the decisions and remove conflict between decision-makers. The algorithm consisted of the consensus process and the RAT selection process. They used a soft consensus model for their consensus process and then used MCGDM for the RAT selection process. For a single call, the algorithm selects a single RAT by MCGDM. For a group of calls, the algorithm “determines the consensus level among the handoff calls to be admitted into a RAT. If the consensus level is above a certain threshold, the best single RAT is selected for the group”. Otherwise, “independent call decisions” are done. The consensus level was used to preserve the QoS of individual calls in the group so that their quality isn't compromised beyond acceptable levels in the group decision. Otherwise, it admitted them independently.

To evaluate the performance of their algorithm they measured the number of calls selected into each RAT for each type of call and the proportions that were based on group decisions and independent decisions. They compared results with an algorithm that didn’t use consensus. The consensus threshold and service priority levels were varied. By increasing the consensus level there was an increased probability of call-by-call decisions rather than group decisions. When the priority was changed there were fewer group decisions than when all the calls had the same priority. They concluded that "multiple calls could not easily reach a consensus on a single suitable RAT for them without compromising the users' preferences for individual calls beyond acceptable limits". Also, priority affects the distribution of calls in individual RATs.

The benefits of their approach were that the proposed consensus-based algorithm was able to combine the advantages of both independent and group call decisions. It was able to effectively find the compromise between making independent calls and group calls to meet individual QoS requirements.

The disadvantage of their approach was that it also only considered user preferences and network attributes.

The authors in [10] proposed a Fuzzy MCGDM algorithm for multiple call handover. The algorithm used call priorities, user-specified weights for RAT selection criteria per call, and individual RAT characteristics for the RAT to select the most appropriate RAT for the group of calls from the MT. The algorithm first aggregates different call weights for each criterion to obtain a group weight per criterion for all the RAT-selection criteria. It then aggregates the weights across all criteria to make a RAT selection for the group of calls.

To evaluate the performance of their algorithm they measured the number of calls admitted/selected into each RAT under different scenarios where the priorities for each service were varied and the user-assigned weights were randomized. They concluded that selection decisions for group calls in heterogeneous wireless networks varied extensively with individual call priorities.

The advantage of their approach was the use of fuzzy logic to eliminate uncertainties and imprecisions introduced in the algorithm by linguistic terms used to describe certain RAT selection criteria such as security and price resulting in a more well-defined selection decision for the group.

They only considered making group decisions and not making independent call decisions by taking the consensus level into account as done in [9].

In [11] Taiwo and Falowo conducted a comparative evaluation of four GDM algorithms for RAT selection of group handoff calls from an MT. They compared four possible algorithms that can be adapted for making MCGDM for multiple handoff calls from MTs in heterogeneous networks. The evaluation of the four algorithms was done by a sensitivity analysis of the criteria they chose for network selection. Criterion sensitivity was defined as how much a criterion influences the selection decision and load distribution for multiple sessions in HWNs, due to user-specified preference weight levels assigned to the criterion. The criteria that were assessed included throughput, service cost, security, battery power consumption and latency.

The results showed that TOPSIS and DIA (Distance to Ideal Alternative) algorithms were not as sensitive to the change of weights than SAW and MEW as they had more uniform distributions for multiple handoffs across RATs for the different criteria assessed. They therefore concluded that TOPSIS-GDM and DIA-GDM algorithms are better suited to RAT selection for group hand-off calls from a MT than SAW and MEW GDM algorithms since they better distribute the calls among the RATs thereby better utilizing resources and load balancing the network load.

The authors in [12] proposed an intuitionistic Fuzzy TOPSIS framework to select the best RAT for various users running multiple services. Their framework also considered dynamic parameters in terms of network cost when selecting the optimal RAT. The proposed algorithm used IF-TOPSIS to model the uncertainties and ambiguities in heterogeneous wireless networks. Intuitionistic fuzzy logic which is an advanced form of fuzzy set theory was used to improve the precision of the network parameters such as delay, jitter, and throughput as these cannot be accurately measured in a dynamic network environment. The proposed algorithm also considered different classes of mobile subscribers by simulating users with different tariff plans (willingness to pay).

They compared their algorithm performance with TOPSIS and measured the number of calls admitted into each RAT for the different subscriber classes, the network price was varied to mimic "time-dependent pricing by network operators". From their results, they concluded that the proposed scheme could eliminate the rank reversal problem and select the best RAT depending on the characteristics of the subscriber's request.

The advantages of their approach were the reduction of the rank reversal problem which affects algorithms such as TOPSIS. When the worst-performing RAT was removed the proposed algorithm remained stable in terms of network ranking as opposed to TOPSIS. Also, their model was made more realistic by considering different classes of users and willingness to pay. And their use of fuzzy logic resulted in more accurate dynamic parameters being used for their solution.

The disadvantage of their method was that they only considered user preferences and network attributes.

In [13] Luo et al proposed a RAT selection algorithm that uses AHP, GDM and utility functions to obtain a weight vector for the network attributes per service, synthesize it and normalize the network attributes respectively. The synthesized weight vector and attribute utility are combined to determine the most appropriate RAT for a group call handoff on a mobile terminal. They compared the performance of their algorithm with TOPSIS by using Markov chains to simulate changes in the network attributes. Each network attribute was evaluated with respect to state transition probability. Each service was treated as a decision-maker when using the GDM algorithm and the network was selected according to service characteristics.

They concluded that the proposed GDM-based algorithm can better meet the QoS requirements of the services in the network than TOPSIS.

The advantage of their approach is that they considered service characteristics to the extent that all services on the MT can meet their QoS requirements. However, the disadvantage of their approach was that the algorithm only considered service requirements and ignored user preferences resulting in poor user experience and a solution that was too objective.

Obayiuwana and Falowo [14] proposed a RAT selection algorithm for group calls that had multiple criteria that were dynamic. The algorithm was termed "MULTlplicative–form with Multi–Objective Optimization Ratio Analysis (MULTIMOORA)". The impact of dynamic criterion and the level of importance of a call in group calls was investigated in their work. It was the first paper to investigate the impact of criterion dynamics on VHO decisions for group calls. The proposed algorithm integrated "three ranking approaches (ratio system, reference point system and multiplicative form)" and then used the "theory of dominance" to integrate the three independent ranking systems into a way of ranking the alternatives. The rank of an alternative was based on its dominance in the ratio, reference points and multiplication form systems.

The performance of the algorithm was evaluated by measuring the number of calls admitted into each RAT as dynamic criteria such as MT speed were varied. This was done to indicate the algorithm stability in selecting a RAT in high-speed regions. The MULTIMOORA algorithm was compared with TOPSIS for a range of mobile terminal speeds at various degrees of importance of triple calls occurring on the MT.

They concluded that the proposed algorithm had better stability than TOPSIS in high-speed regions. The advantage of their algorithm was that it considered dynamic criteria such as MT speed and traffic load variation in the selection decisions.

To conclude the related work undertaken for RAT selection algorithms for MMTs in heterogeneous wireless networks in the literature shows that they can reasonably select the optimal network for multiple simultaneous calls on the MMT. However, their main shortcomings are that they either only take user preferences into account resulting in very subjective selection decisions [12-18] or they are too objective by only considering service characteristics such as [13]. There is a poor combination of the factors needed for network selection decisions (user preferences, service characteristics and network attributes) for group calls as discussed in [7]. Also, most of the works except [9] only consider group decisions in handing over the group of calls from the MT to a single RAT. They do not consider the QoS requirements of the individual services sufficiently in terms of (level of agreement between the decision makers) resulting in individual QoS requirements getting compromised for the sake of the group decision. I.e., the best RAT for the group isn't necessarily the best RAT for each service in the group since they do not consider selecting multiple RATs for independent call-by-call decisions for the services in the group as done by [9].

The rest of the works reviewed below are for the RAT selection of a single call from an MMT. These are the most recent works done in the literature so are reviewed for completeness.

A.Sgora et al, propose a network selection algorithm for 5G heterogeneous environments [15]. They propose using a combination of MADM techniques for network selection in heterogeneous environments.
They compared the performance of two MADM techniques for network selection (TOPSIS and VIKOR). For weight calculation, they combined "the subjective weighting method AHP with two objective methods, the Criteria Importance Through Intercriteria Correlation (CRITIC) method and the entropy weighting method". To compare the performance of the 2 algorithms they used the same approach as [16]. The RATs used in their simulation were "5G, WLAN,3G and LTE-A" on the assumption that a mobile terminal connects to one RAT at a time with all the appropriate interfaces. The network attributes used for the simulation were the "data rate (DR), the packet loss rate (PLR), the jitter (JIT), the price rate (PR) and the delay (DE)". The traffic classes used to determine network selection were "conversational, interactive, background and streaming classes".

AHP determined the weights that represent the relative importance of different applications. CRITIC and entropy calculated objective weights for all criteria. For analysis, they used positive reciprocal matrices. The weights of the network decision parameters for each traffic (call) class were graphed after calculating using both AHP and AHP combined with Entropy and CRITIC. Then TOPSIS and VIKOR were used to Rank the networks(choose the best network for a particular class of call) after the weights calculation for the different network decision parameters of the different traffic classes. Then the performance of the new ranking methods was examined.

The conclusions they drew were that the weight parameters affect network selection for conversational and interactive classes. When only AHP was applied to the classes, all of them except the interactive class had identical rankings. Their limitation was that they had a limited number of scenarios for simulation.

In [17] a network selection algorithm called "BGMNS (Bipartite Graph Matching Network Selection Algorithm)" was proposed for edge users in 5G heterogeneous networks. The algorithm was designed to be adaptive and for multi-access. The problem of network selection for edge users was modelled as a bipartite graph. The algorithm was based on a weighted bipartite graph. The BGMNS algorithm combined AHP and GRA(Grey relational Analysis) to analyse multiple service preferences to obtain the QoE of edge users. This was used to create a fair allocation of network resources.

The RATs used for their network model simulation were "5G, LTE, WiMAX, UMTS, and Wi-Fi 6". The network attribute parameters used in the simulation were "bandwidth, energy efficiency, delay, packet loss rate, jitter, price, and convergence". The traffic classes requested by users included Virtual reality and augmented reality, Industrial machinery, office automation, background services and streaming where Office automation is an intelligent office that makes use of Artificial intelligence and Machine learning for services such as 4K video conferencing, intelligent security, and remote monitoring. Industrial machinery involves using 5G communication combined with industrial internet to achieve automation for real-time monitoring and remote control of machinery. They verified the performance of the BGMNS algorithm in a network that had fluctuating status.

The impact on QoE was tested for fluctuating network status, The algorithm enabled users to access the best network according to service priority. The algorithm was robust in the dynamic network environment. The mean QoE was consistent with service priority.

The impact of network status fluctuation was also assessed by observing changes to the system fairness index of four algorithms (BGMNS, HUMANS, FTNS, and DVHD). According to their definition of fairness index, "the system fairness index was expected to be as large as possible with a maximum value of 1". The result of the algorithm varied between "0.75 and 0.78", while the others varied more. The result showed that the proposed BGMNS algorithm had good stability when network status fluctuated.

The Impact of blocking probability was also assessed when the network status fluctuated. The results for their algorithm "changed steadily, remaining between 0.208 and 0.225" while the other 3 algorithms had more fluctuation. The result showed that when the network status fluctuates, the user blocking probability obtained by BGMNS changes steadily.

The effect of the number of users on user access was also assessed by showing the ratio of edge users requesting different services. Results showed that “UMTS can provide a good QoE for users who requested traditional services”. As the number of users increased, the network could not meet the bandwidth requirements of the requested services. Their algorithm accounted for this by adjusting users according to available resources.
System fairness was also compared by varying the number of users in the simulation. As the number of users increased the available bandwidth of networks decreased resulting in some users getting blocked. However, the proposed BGMNS algorithm had the gentlest downward trend because it considered the “fair allocation of network resources and the importance of services” whereas the other algorithms only considered optimal network selection.

The blocking probability as the number of users increased was also compared between the different algorithms. The user blocking probability increased due to the limited bandwidth. When the number of users increased, the network couldn’t meet the access needs of all users causing increased blocking probability. The results showed that BGMNS had a lower probability than the others because it “reduced the risk of all users accessing the same network by effectively considering the service's importance”.

The average energy efficiency experienced by users was compared between algorithms. Results showed that BGMNS was the most energy efficient due to its “fair allocation of network resources among users requesting different services and taking personal requirements of different services into account”.

After conducting all these experiments, they concluded that the proposed BGMNS algorithm had stable performance for a fluctuating network. It also reduces the user blocking probability and packet loss rate and improves the average energy efficiency. Because of how it allocates resources and takes users' different service requirements into account. Therefore, their algorithm would provide users with the best QoE.

S. Sridevi et al proposed a network selection algorithm for beyond 5G networks in [18].
The proposed algorithm was for the selection of an unmanned aerial vehicle (UAV) using an AHP (Analytic Hierarchy Process) based selection mechanism. The algorithm essentially captured the trade-off between power consumption and packet loss and secrecy throughput(Throughput not overhead by other mobile terminals) when associating a mobile terminal with a UAV.

The performance of their proposed algorithm was compared against the best downlink (BD) selection mechanism in terms of packet loss rate(packets lost per unit time), user throughput and secrecy throughput. The traffic load on user-perceived throughput was assessed with the AHP-based mechanism outperforming the BD mechanism. The AHP-based mechanism selected the lower altitude UAV due to its better throughput than high altitude UAVs which had higher power consumption.

The effect of user velocity on packet loss rate was assessed and the AHP-based mechanism outperformed the BD mechanism due to it selecting high-altitude UAVs which had lower packet loss rates, whereas BD always chose a UAV based on downlink condition irrespective of altitude. It resulted in lower handovers for increasing velocity.

The effect of traffic load on secrecy throughput was assessed and the AHP algorithm outperformed the BD algorithm by having a higher secrecy throughput. The BD algorithm always selected a UAV providing the maximum downlink SINR (Signal interference to noise ratio) irrespective of altitude whereas the AHP mechanism prioritized the mobile terminal by accounting for trade-offs with altitude. However, throughput decreased with increased traffic load at any altitude, and it was observed that in extreme traffic loads (>400 mobile terminals) the secrecy throughput of BD was roughly the same as AHP.

They concluded that the proposed AHP algorithm selects the best UAV based on the application class since it outperforms the existing best downlink (BD) based approach in terms of user throughput, secrecy throughput and packet loss rate. Therefore, it has potential use in next-generation heterogeneous networks such as beyond 5G which make use of non-terrestrial access points such as the UAVs.

The authors in [19] proposed an intelligent network selection mechanism for hybrid classical-quantum systems. The algorithm was based on FAHP and TOPSIS and was designed to handle hybrid communication scenarios involving both classical and quantum communication systems. This was aimed at addressing the issues when generalizing classical RAT selection algorithms to hybrid systems. E.g., the bit error rate which is a classical attribute is described differently in a quantum channel therefore generalization isn't straightforward.

Their simulation introduced energy awareness into the system by adding a power consumption module so that QoS requirements are met for various application classes. The RATs used for the simulation were UMTS and 5G-NR. A simulation was dynamically done using Python, with the attribute values varied between the standard min and max for "1000-time steps and tracking attribute values of the selected network for each application class". The network attributes used for simulation were Bandwidth, Qubit transfer rate, Average fidelity, Bit Error Rate (BER), packet loss, delay, and price. The traffic classes simulated were conversational, streaming, interactive, background, entanglement distribution and quantum key distribution which are quantum classes. The dynamic performance of the FAHP-TOPSIS algorithm was evaluated for both classical and hybrid communication systems by tracking the values of each network parameter over time and plotting these values for all traffic classes per graph. The dynamic system used Quantum Key distribution and entanglement distribution. A power consumption analysis was also done for each selected network by tracking the power consumption of each traffic class individually.

Simulation results showed that the selected network was able to keep a "standard level of the attribute values for each application class while at the same time minimizing power consumption for the selected networks." They concluded that the proposed algorithm provides a "reasonable level of performance" while also "maintaining energy awareness" simultaneously.

The limitations of their study, however, were that there was a lack of datasets "related to quantum communication systems" as the field is still new (since 2022). An increased amount of parameter tuning is needed to obtain optimal operation due to the large number of free parameters(triangular fuzzy numbers). They therefore recommended using the proposed model as a template for further optimization using machine learning and due to the lack of datasets, reinforcement learning based on agent-environment models was opted for as a more feasible option.

In [20] González et al proposed a "Dynamic radio Access selection and Slice Allocation (DASA)" algorithm for traffic management on future mobile networks. The algorithm combined the network slicing paradigm with software-defined networking and network function virtualization to provide flexible traffic management for future wireless networks. The radio access selection was based on MADM and AHP. The proposed algorithm was said to differentiate between novel services such as virtual reality, cloud gaming and IoT(Internet of things) in terms of throughput, "delay, jitter, packet loss ratio and energy consumption". The algorithm was evaluated by simulation at network level, emphasising flexibility and the utilization of network resources during RAT selection and load-balancing.

The network was simulated using OMNeT++ with Simu5G and Python tools. Python was used to implement the algorithm and test its performance. The RATs used for simulation were 5G-NR and WLAN(802.11ac). The simulation also modelled pedestrians being stationary or moving linearly in the cells. The network attributes used for their simulation were bandwidth, data rate and transmission power. Two scenarios were conducted by simulation with the first scenario evaluating the network selection process by having background users in the area move around the base stations, requesting up to 4 services simultaneously. A new user was introduced and changed their request at fixed times to test the algorithm's performance under different conditions. The second scenario was similar to the first but had increased users to evaluate the network selection and load-balancing mechanisms in terms of ADR(Aggregated data rate), throughput satisfaction, and satisfaction degree.

The results from the first scenario simulation showed that the DASA algorithm always selected the best RAT and network slices, by considering user priority, network conditions, and the Network slices' access.

They concluded that since the proposed DASA algorithm was able to consider users that had different subscription plans, device resolutions, and service preferences and was also able to consider users requesting multiple services simultaneously, it was guaranteed to choose the most resourceful combination of access networks and network slices to satisfy the user requirements.

Dai et al [21] proposed a multi-objective intelligent handover scheme (MIHO) for network selection in satellite-terrestrial integrated networks (STIN). The scheme uses a handover algorithm based on discrete particle swarm optimization (ffiPSO-HO) which is designed by jointly considering the achievable rate and load balance. The study aimed to address the problem of continuous communication/coverage for low earth orbit satellites (LEO) which experience high delays and time-varying network topology due to their fast periodic motion. The MIHO scheme was implemented using the IBPSO-based handover algorithm (IBPSO-HO) to maximize throughput and balance load under the constraint of handover delay. The network model used for simulation consisted of Low Earth Orbit satellites and terrestrial base stations. Handover delay was modelled to guarantee QoS requirements. Randomly distributed users were represented as a mathematical set. The height, minimum elevation, transmit power channel model and coverage radius for the satellites and base stations were simulated. The network simulation could service multiple users simultaneously but only one user selected the network at a time.

The performance of the proposed algorithm (IBPSO-HO) was compared with the received signal strength-based handover scheme (RSS-HO) and Neuro-Fuzzy and particle swarm optimization combined handover scheme (NFPSO-HO). The throughput performance of the proposed algorithm was lower than RSS-HO because it accounted for the trade-off between throughput and load balancing whereas RSS-HO only considered throughput. The variance of the number of users served by each network for IBPSO-HO, was smaller than that of the other two algorithms because it considered load balancing. In terms of average handover delay, the MIHO scheme had the lowest average handover delay of the three.

Based on the obtained results they concluded that the IBPSO-HO algorithm obtained optimal solutions and had smaller computational complexity than other algorithms. They recommended that future studies should consider resource allocation after handover and include machine learning techniques. Their only limitation was that they had insufficient time to produce more results.

In [22] a multi-attribute access selection algorithm was proposed for heterogenous wireless networks. This study was done to address the uncertainty in network attributes The algorithm was based on fuzzy network attributes where the subjective weights were calculated by AHP, and the objective weights were calculated by the entropy. The combined weights were obtained by the "longest geometric distance to the negative ideal solution". The scores of the candidate RATs were determined by GRA based on an intuitionistic fuzzy decision matrix.

MATLAB was used to compare the algorithms under investigation. The network model used "bandwidth, delay, jitter, and error" as the network attributes. These were dynamically changed to simulate user movement. The RATs used for simulation included "UMTS, LTE, WLAN and WiMAX". The application classes/services used were voice, video, and data. The simulation first tested the performance of the algorithm. Then the second simulation involved comparing the performance between algorithms. This was in terms of the number of selections of candidate networks, the number of handoffs and unnecessary handoffs for different services.

The results showed that the proposed algorithm can reduce unnecessary handoffs to a greater extent than the rest since it has both a lower number of handoffs and unnecessary handoffs than the other algorithms for each application.

They concluded that the algorithm could select the best RAT and reduce unnecessary handoffs with inaccurate network attribute values. The algorithm is suitable for access selection in the scenario of network attribute value uncertainty.

Xu et al [23] proposed two online RAT selection heuristic algorithms for a 5G multi-RAT network. The proposed algorithms did not require knowledge of the statistics of system dynamics and had low computational and storage complexity. Network simulator 3 (ns-3) was used for simulation based on 5G-NR setup using the 5G stack for mmWave with customizable PHY and MAC layers. The performance of the proposed algorithms was compared with traditional RAT selection schemes under many practical scenarios which included user mobility. Performance was assessed in terms of blocking probability of high-priority users, offloading probability of low-priority users and throughput. They also compared the performances of the proposed algorithms with the association scheme adopted in the existing network and compared the performances of the proposed algorithms with on-the-spot offloading in the face of user mobility.

They concluded that their proposed algorithms did not suffer from high storage complexity and slow convergence issues which are present in learning-based schemes.

3. Methodology

The methodology used for RAT selection was a modified version of the algorithm implementation used in [7]. In addition to the limitations discussed in section 1.3 above, the following assumptions were made concerning the network model, algorithm design and simulation setup.

3.1 Assumptions

- A user/MMT can connect to any RAT in the network i.e., all RATs in the network can support the services on the MMT(have unlimited resources).
- A user/MMT only connects to (selects) one RAT at a time i.e., the algorithm only considers group decisions and not independent call decisions done in [9] for vertical handoff of a group of calls from the MMTs. The MMT power consumption and coordination between RATs for multiple simultaneous independent call handovers hasn't been simulated for simplicity.
- It is assumed one or more services can simultaneously run on the MMTs in the network being simulated.
- It is assumed the MMT has the intelligence for the VHO i.e., the algorithm is executed by the MMT.
- It is assumed the handoff is a hybrid form of mobile controlled handoff but using information provided by the network.
- It is assumed the algorithm has already been triggered initially (At time 0) i.e., the events triggering the algorithm are assumed to have already occurred when the simulation begins.
- It is assumed the MMTs are within the coverage of all RATs in the network.

3.2 Problem Definition

Let $R=\{r_1, r_2 \ldots, r_{|R|}\}$, $|R| \geq 2$ be the set of RATs present in the network. Let $S=\{ s_1, s_2 \ldots s_{|S|}\}$, $|S| \geq 1$ be the set of supported services in the network. The set of RAT selection criteria used for the RAT selection by the MMT is denoted by $C = \{ c_1, c_2 \ldots c_N \}$, $N \geq 2$. Let $S^t = \{ s^t_1, \ldots, s^t_g, \ldots s^t_Y \}$, $s^t_g \in S$, $Y \leq |S|$ be the set representing the group of services running simultaneously on an MMT (where t refers to an MMT). $|X|$ is the number of members in a set , N is the number of criteria and Y is the number of services. The goal of the algorithm is to select the best RAT $r_i \in R$ to admit the set of services S^t on the mobile terminal [7].

3.3 Network Model

Figure 1 below shows the network environment modelled for the designed algorithm. Clip art obtained from [24].

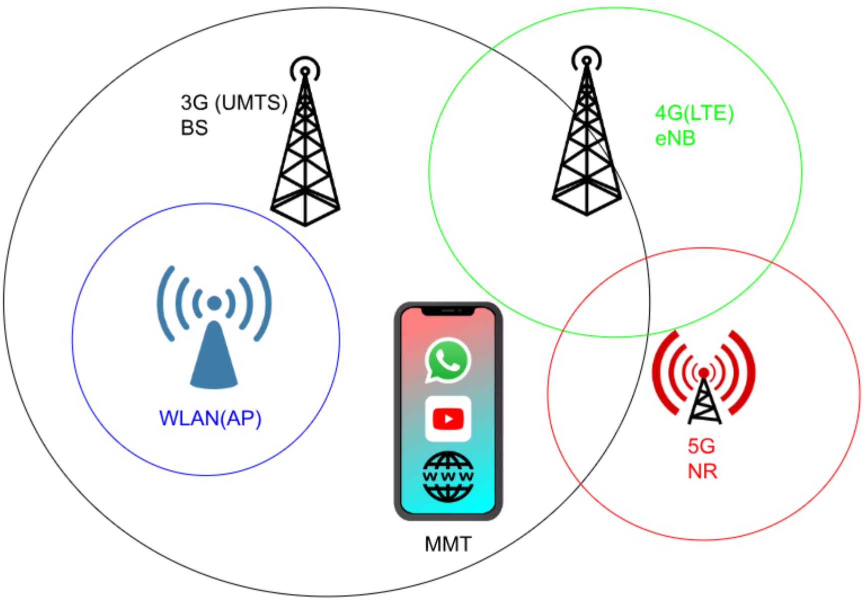


Figure 1: Network Model

The RATs used for the algorithm were 5G(nR), 4G(LTE), 3G (UMTS) and WLAN (IEEE 802.11.ax). The set of services running simultaneously on the MMT were voice (s^t_1), video streaming (s^t_2) and web browsing (s^t_3). The network attributes/RAT selection criteria used for the model were bandwidth(c_1) (benefit criteria), cost(c_2), delay(c_3) and packet loss rate(c_4) (cost criteria) as shown in table 2 below. The table shows the range of performance ratings that the RATs have for each criterion. The values obtained are a combination of values across the literature [7], [15], [25] and [26].

Table 2: Network attribute values [7] [15] [25] [26]

RAT	Bandwidth (Mbps)	Cost ($/GB)	Delay (ms)	Packet loss rate (%)
3G (UMTS)	0.7-2	3.5	10-50	2-10
4G (LTE)	0.8-100	4.5	40-80	6-20
WLAN	1-100	0.5	70-90	4-15
5G	100-1000	7	1-25	0.1-8

The choice of RATs was due to them being the latest and most common access technologies used around the world. Likewise, the services used were chosen because they are some of the most common applications that MMTs run, and the choice of the above RAT selection criteria was due to them being the most used throughout the literature on RAT selection algorithms [4].

3.4 Algorithm Design

Figure 2 below shows the submodules that the algorithm is composed of. Each submodule will be explained in detail below. Numeric examples will be used to illustrate each step.

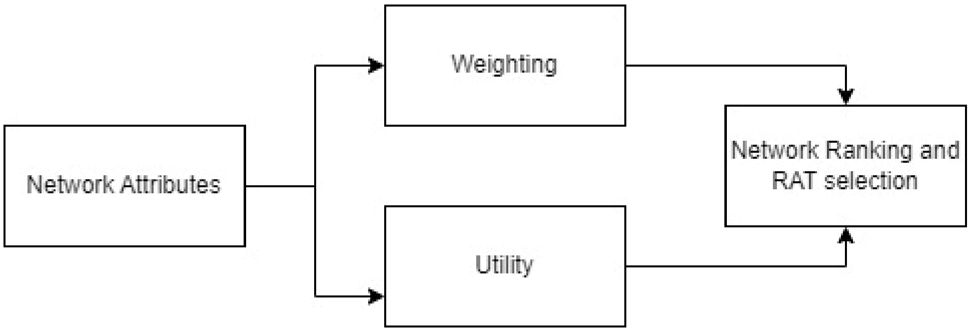


Figure 2: Block diagram of algorithm submodules

3.4.1 Network Attributes

The inputs to this submodule were the performance ratings of the RAT selection criteria/network attributes and the output was a normalized matrix of these values. The performance ratings for each RAT in the network were detected by the MMT at an arbitrary point in time t. For the example scenario below the values used in table 3 are obtained from table 2 and represent the detected network attributes at time t. This table was used for all the numerical examples in each sub-section.

Table 3: detected Network attribute values at time t

RAT	Bandwidth (Mbps)	Cost($/GB)	Delay (ms)	Packet loss rate (%)
3G (UMTS)	1.35	3.5	30	6
4G (LTE)	2.4	4.5	60	13
WLAN	4.5	0.5	80	10
5G	1000	7	13	4

Step 1: Create and populate a network attribute matrix M^t using values from table 3 as shown by (1).

$$M^t = (m_{ij})_{MxN} \quad (1)\ [7]$$

where m_{ij} is an element of the matrix which is the MMT detected value of criterion c_j for RAT r_i (performance rating) and (i=1, …, M and j=1, …, N) M is the number of RATs in the network and N is the number of RAT selection criteria.

Example matrix using table 3 values.

$$M^t = \begin{bmatrix} 1.350 & 3.5 & 30 & 6 \\ 2.400 & 4.5 & 60 & 13 \\ 4.500 & 0.5 & 80 & 10 \\ 1000 & 7 & 13 & 4 \end{bmatrix}$$

Step 2: Normalize the matrix M^t using (2) below to create the normalized network attribute matrix $\overline{M^t} = (\overline{m_{ij}})_{MxN}$

$$\overline{m_{ij}} = \frac{m_{ij}}{\sum_{i=1}^{M} m_{ij}} \text{ }, j = 1, \dots, N \quad (2)\ [7]$$

Example matrix

$$\overline{M}^t = \begin{bmatrix} 0.001289 & 0.2258 & 0.1639 & 0.1818 \\ 0.002380 & 0.2903 & 0.3279 & 0.3939 \\ 0.004463 & 0.0323 & 0.4371 & 0.3030 \\ 0.9918 & 0.4516 & 0.07104 & 0.1212 \end{bmatrix}$$

(i.e., the normalized element is the current element divided by the sum of elements in the current column)

3.4.2 Weighting

The inputs to this submodule were the normalized network attribute matrix ($\overline{M^t}$) obtained from section 3.4.1, the user-specified attribute weights and priority per service, service characteristics and service-determined priority for each service. The output of this submodule is the comprehensive weight vector which is a weighted combination of the user-specified subjective weights, service characteristic weights and network attribute objective weights.

i. Objective weight vector calculation

Let $W^O = \{w_1^O, \dots, w_j^O, \dots, w_N^O\}$ be the network attribute objective weight vector. The elements of the vector are calculated using Entropy as shown below.

Step 1: Calculate the entropy of each criterion using (3) as shown below.

$$E_j = -\frac{1}{\ln(M)} \sum_{i=1}^{M} \overline{m_{ij}} \ln(\overline{m_{ij}}) \quad (3)\ [7]$$

Where E_j is the entropy value of criterion c_j, M = number of RATs and $\overline{m_{ij}}$ is the normalized criterion from $\overline{M^t}$ and j=1, …, N(number of attributes). The entropy value for criterion c_j is found by multiplying the constant with the sum of products of the element with its natural logarithm for a fixed column.

Example calculation
For instance, to calculate the entropy of c_1 = bandwidth, M=4

$$E_1 = -\frac{1}{\ln(4)} \sum_{i=1}^{M} \overline{m_{i1}} \ln(\overline{m_{i1}}) = 0.04$$

Then repeat the same for the other criteria.
Step 2: calculate the objective weight w_j^O of criterion c_j using (4) below.

$$w_j^O = \frac{1 - E_j}{N - \sum_{j=1}^{N} E_j} \quad (4)\ [7]$$

Example calculation
To find objective weight of c_1 (bandwidth)

$$w_1^O = \frac{1 - E_1}{4 - 2.6881} = 0.7317$$

Step 3: Populate W^O using the values obtained from step 2 to get the network attribute objective weight vector.

Example

Applying (4) to all criteria $\boldsymbol{W^{0}} = \{\mathbf{0.7317}, \mathbf{0.1218}, \mathbf{0.0960}, \mathbf{0.0505}\}$

ii. Subjective user preference weight vector calculation

Let $W^{t,U} = \{w_{g,j}^{t,U}\}$ be user-specified weights (superscript U refers to the user) where each element is the weight of criterion c_j for service s^t_g. In a similar manner let $W^{t,S} = \{w_{g,j}^{t,S}\}$ be the service-determined weights (superscript S refers to service) where each element is the weight of criterion c_j for service s^t_g. Let $P^{t,U} = \{p_g^{t,U}\}$ represent the set of user-specified priorities for each service in S^t and let $P^{t,S} = \{p_g^{t,S}\}$ be the service-determined priority for each service in S^t. Where g=1 ,…, Y and j=1 ,…, N. For all subsequent calculations assume $P^{t,U} = P^{t,S}$.

Priority values measure the importance of a service in the group to a user. The priority values ranged from 1(Very low) to 5(Very High). The user-specified weights were based on a 9-point scale [1,9] where 1=criterion not important to user and 9=criterion very important to the user. The allocation of user-specified weights and service priority are done once and are always used to select the RAT, but these can be modified based on user preferences.

Step 4: Allocate the user-specified weights $w_{g,j}^{t,U}$ for each criterion into the weight vectors $W_g^{t,U}$ for each service s^t_g where g = 1, …, Y

$$W_g^{t,U} = \{w_{g,1}^{t,U}, \dots, w_{g,j}^{t,U}, \dots, w_{g,N}^{t,U}\}$$

Example of a single user's preferences for each service.

Voice $W_1^{t,U} = \{6,8,9,5\}$

Video streaming $W_2^{t,U} = \{9,5,6,8\}$

Web browsing $W_3^{t,U} = \{6,9,9,6\}$

These weights were chosen at random, but values typically reflect what a user for a specific service may deem more important to them.

Step 5: Normalize the weight vectors $W_g^{t,U}$ for each service using (5) as shown below.

$$\overline{w}_{g,j}^{t,U} = \frac{w_{g,j}^{t,U}}{\sum_{j=1}^{N} w_{g,j}^{t,U}} \quad (5)\ [7]$$

i.e., the normalized weight is the current weight divided by the sum of weights in the vector. This results in the normalized weight vectors $\overline{W}_g^{t,U}$ for each service.

Example

Voice $\overline{W}_1^{t,U} = \{0.2143, 0.2857, 0.3214, 0.1786\}$

Video streaming $\overline{W}_2^{t,U} = \{0.3214, 0.1786, 0.2142, 0.2857\}$

Web browsing $\overline{W}_3^{t,U} = \{0.2, 0.3, 0.3, 0.2\}$

Step 6: Allocate the user-specified priorities $p_g^{t,U}$ for each service into the service priority vector $P^{t,U}$

$$P^{t,U} = \{p_1^{t,U}, \dots, p_g^{t,U}, \dots, p_Y^{t,U}\}$$

Example

$$P^{t,U} = \{3,3,3\}$$

Step 7: Normalize $P^{t,U}$ using (6) below to obtain the normalized service priority vector specified by users $\bar{P}^{t,U}$

$$\overline{p_g}^{t,U} = \frac{p_g^{t,U}}{\sum_{g=1}^{Y} p_g^{t,U}} \quad (6)\ [7]$$

Example

$$\bar{P}^{t,U} = \{0.33, 0.33, 0.33\}$$

Where $\bar{p}_g^{t,U}$ is an element of $\bar{P}^{t,U}$ and is the normalized priority of service s^t_g.

Let $W^{U} = \{w_1^{U}, \dots, w_j^{U}, \dots, w_N^{U}\}$ be the user determined network attribute weight vector for multiservice.

Step 8: To obtain W^U, synthesize(aggregate) $\bar{P}^{t,U}$ and $\bar{W}_g^{t,U}$ by using (7) below.

$$w_j^U = \sum_{g=1}^{Y} \bar{p}_g^{t,U} * \bar{w}_{g,j}^{t,U} \quad (7)\ [7]$$

Example

NB: the 3 weight vectors for each service are combined into a matrix (each row represents a different service) so that they can be used in step 8.

$$w_1^U = \bar{p}_1^{t,U} * \bar{w}_{1,1}^{t,U} + \bar{p}_2^{t,U} * \bar{w}_{2,1}^{t,U} + \bar{p}_3^{t,U} * \bar{w}_{3,1}^{t,U} = 0.2452$$

Repeat the same for all criteria then $\boldsymbol{W^U} = \{\mathbf{0.2452, 0.2548, 0.2786, 0.2214}\}$

Where Y= number of services simultaneously running on the MMT and w_j^U is the weight of criterion c_j for a service group determined by the user. i.e., each element in W^U is found by calculating the sum of products of each normalized service priority with each normalized weight of a criterion j for all services.

iii. Service determined weight vector calculation.

The same definitions for $W^{t,S}$ and $P^{t,S}$ defined above are used here and it is assumed $P^{t,U} = P^{t,S}$. The Fuzzy analytic hierarchy process (FAHP) is used to determine the service-determined weight vector of network attributes defined as $W^S = \{w_1^S, \dots, w_j^S, \dots, w_N^S\}$. FAHP eliminates the ambiguity associated with decision criteria by use of fuzzy numbers. It differs from AHP in the sense that the consistency of the pairwise comparison matrices is already guaranteed when they are formed so doesn't need to be judged [27]. Fuzzification is the process used to convert linguistic descriptions into crisp values using a conversion scale such as in [14].

It is a very common approach used to determine subjective weights for network parameters in the literature as seen by [7], [27] and [28]. The decision-making process is placed into different hierarchies and the comparison of criteria for each layer are the fuzzy numbers. Similar to AHP, comparison matrices are formed to compare each attribute with respect to the goal. The FAHP approach for this report was obtained from [7] and is based on the extent analysis method.

Triangular fuzzy numbers are used to represent the fuzziness of preferences. The TFN definition and TFN arithmetic used in this approach were the same as [7]. The membership function for fuzzy numbers has been repeated in table 4 below. The value for equal importance was changed from (1,1,3) in [7] to (1,1,1) for this report.

Table 4: Triangular fuzzy numbers used [7]

Definition	Triangular fuzzy number	reciprocal
Equally important	(1,1,1)	(1,1,1)
Intermediate values	(1,2,4)	(0.25,0.5,1)
Moderate Importance	(1,3,5)	(0.2,0.33,1)
Intermediate values	(2,4,6)	(0.166,0.25,0.5)
Strongly important	(3,5,7)	(0.142,0.2,0.33)
Intermediate values	(4,6,8)	(0.125,0.166,0.25)
Very Strong importance	(5,7,9)	(0.111,0.142,0.2)
Intermediate values	(6,8,9)	(0.111,0.125,0.166)
Extremely important	(7,9,9)	(0.111,0.111,0.142)

Step 9: Determine the normalized service-determined priority vector $\bar{P}^{t,S} = \{\bar{p}_g^{t,S}\}$. Since $P^{t,U} = P^{t,S}$ Then $\bar{P}^{t,S} = \bar{P}^{t,U}$. So just use the same vectors calculated in step 6 and 7 above.

Example

$$P^{t,S} = \{3,3,3\}$$
$$\bar{P}^{t,S} = \{0.33,0.33,0.33\}$$

Step 10: create the hierarchy of the RAT selection problem to be used by FAHP as shown in figure 3 below for example.

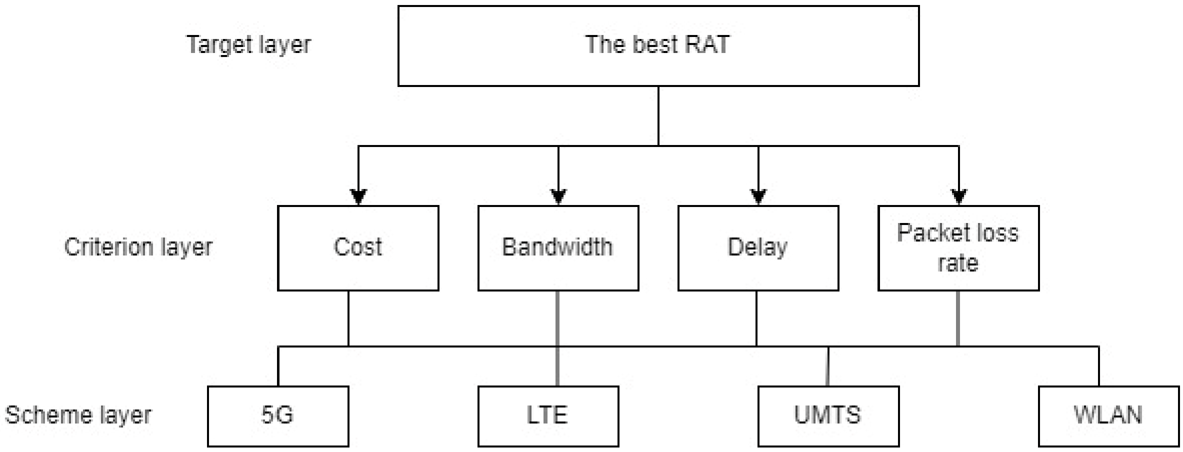


Figure 3: FAHP hierarchical structure

Step 11: Construct a pairwise comparison matrix A^g for each service using table 4 as shown in (8) below.

$$A^g = \left(a_{ij}\right)_{NxN} \quad (8)\ [7]$$

Where $a_{ij} = (l_{ij}, m_{ij}, u_{ij})$, (g=1 ,…, Y) is a triangular fuzzy number which represents the relative importance of criterion c_i compared to c_j for service s^t_g where $i \neq j, a_{ji} = \frac{1}{a_{ij}}$ else , $a_{ii} = (1,1,1)$. For the case of this network model the matrices were constructed below using information from tables 6-8 in [7].

Example

Table 5: Fuzzy comparison matrix (voice)

Voice	Bandwidth	Cost	Delay	Packet loss rate
Bandwidth	(1,1,1)	(0.125,0.1667,0.25)	(0.25,0.5,1)	(1,2,4)
Cost	(4,6,8)	(1,1,1)	(3,5,7)	(5,7,9)
Delay	(1,2,4)	(0.1429,0.2,0.333)	(1,1,1)	(1,3,5)
Packet loss rate	(0.25,0.5,1)	(0.111,0.1429,0.2)	(0.2,0.33,1)	(1,1,1)

Table 6: Fuzzy comparison matrix (video)

Video	Bandwidth	Cost	Delay	Packet loss rate
Bandwidth	(1,1,1)	(4,6,8)	(1,2,4)	(4,6,8)
Cost	(0.125,0.1667,0.25)	(1,1,1)	(0.2,0.33,1)	(1,1,1)
Delay	(0.25,0.5,1)	(1,3,5)	(1,1,1)	(1,3,5)
Packet loss rate	(0.125,0.1667,0.25)	(1,1,1)	(0.2,0.333,1)	(1,1,1)

Table 7: Fuzzy comparison matrix (Web browsing)

Web browsing	Bandwidth	Cost	Delay	Packet loss rate
Bandwidth	(1,1,1)	(1,2,4)	(3,5,7)	(0.25,0.5,1)
Cost	(0.25,0.5,1)	(1,1,1)	(2,4,6)	(0.2,0.33,1)
Delay	(0.1429,0.2,0.33)	(0.1667.0.25,0.5)	(1,1,1)	(0.125,0.1667,0.25)
Packet loss rate	(1,2,4)	(1,3,5)	(4,6,8)	(1,1,1)

Step 12: Using tables 5 to 7 and (9) below, calculate the comprehensive fuzzy value $S_i = (l_i, m_i, u_i)$ of criterion c_i for each service.

$$S_i = \sum_{j=1}^{n} \alpha_{ij} \otimes \left[\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}\right]^{-1} \quad (9)\ [7]$$

Where $\sum_{j=1}^{n} \alpha_{ij} = (\sum_{j=1}^{n} l_{ij}, \sum_{j=1}^{n} m_{ij}, \sum_{j=1}^{n} u_{ij})$ and $\left[\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}\right]^{-1} = \left(\frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n} u_{ij}}, \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}}, \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij}}\right)$.

Example

For voice(s_1) using (9) and table 5 to calculate the S_i value for bandwidth (c_1) we get the row sum

$$\sum_{j=1}^{n} \alpha_{1j} = (l_1 + \cdots + l_n, m_1 + \cdots + m_n, u_1 + \cdots + u_n) = (2.3750,3.6667,6.2500)$$

Repeating this for each row of table 5 we obtain the sums for each row. (2.3750,3.6667,6.2500),(13,19,25) (3.1429,6.2000,10.3333) ,(1.5611,1.9762,3.2000). Then find the sum of each column to obtain the column sum as shown in table 8 below.

Table 8: column sum

Row number	l	m	u
1	2.3750	3.6667	6.2500
2	13	19	25
3	3.1429	6.2000	10.3333
4	1.5611	1.9762	3.2000
Total	20.0790	30.8429	44.7833

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij} = (20.0790,30.8429,44.7833)$$

Find the inverse of the column sum as shown below. The arithmetic is defined in [7]

$$\left[\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}\right]^{-1} = (\frac{1}{44.7833}, \frac{1}{30.8429}, \frac{1}{20.0790})$$

Then the S_i value is the elementwise product of the two triangular fuzzy numbers. So, for bandwidth for the voice service, we get:

$$S_1 = (2.3750,3.6667,6.2500) * \left(\frac{1}{44.7833}, \frac{1}{30.8429}, \frac{1}{20.0790}\right) = (0.0530,0.1189,0.3113)$$

Repeating the above for the other rows we obtain

$$S_2 = (0.2903,0.6160,1.2451)$$
$$S_3 = (0.0702,0.2010,0.5146)$$
$$S_4 = (0.0349,0.0641,0.1594)$$

Step 13: Calculate the probability(degree of possibility) $V(S_j \geq S_i)$that S_j is greater than S_i using (10) below where S_i and S_j are the calculated comprehensive fuzzy values for criteria c_j and c_i in step 12 and i$\neq j$. Note the conditional statements are slightly different from [7].

$$V(S_j \geq S_i) = \{ 1 \quad if \quad m_j \geq m_i$$
$$0 \ \ if \quad l_i \geq u_j$$
$$\frac{l_j - u_i}{(m_j - u_j) - (m_i - l_i)} \quad otherwise\} \quad (10)\ [7]$$

Since n=4 there are 4 S_i values and there are 12 comparisons that need to be made (3 for each Si value). For S_1 we obtain $V(S_1 \geq S_2) = 0.405,\ V(S_1 \geq S_3) = 0.7459,\ V(S_1 \geq S_4) = 1$

Repeating the comparisons for the other Si values we obtain table 9 below.

Table 9: Degrees of possibility example

S_1 comparisons	0.0405	0.7459	1
S_2 comparisons	1	1	1
S_3 comparisons	1	0.3509	1
S_4 comparisons	0.6599	0	0.3944

Step 14: Calculate the initial service determined weight $w_{g,j}^{t,S\prime}$ of criterion c_j using (11) below.

$$w_{g,j}^{t,S\prime} = minV(S_j \geq S_i) = minV(S_j \geq S_i, S_2, \dots, S_N) \quad (11)\ [7]$$

Example

The initial service determined weight is the smallest degree of possibility from each row in table 9 e.g., for criterion 1 for voice :

$$w_{1,1}^{t,S\prime} = 0.0405$$
$$w_{1,j}^{t,S\prime} = \{0.0405,1,0.3509,0\}$$

Step 15: normalize the initial weights calculated in step 14 using (12) below to obtain $W_g^{t,S} = \{w_{g,j}^{t,S}\}$

$$w_{g,j}^{t,S} = \frac{w_{g,j}^{t,S\prime}}{\sum_{j=1}^{N} w_{g,j}^{t,S\prime}} \quad (12)\ [7]$$

Example

Normalizing the initial weights by (12) we obtain $W_1^{t,S} = \{0.0291,0.7187,0.2522,0\}$

Repeating steps 12-15 for each service to get the Si values and weights for video and web browsing. we obtain $W_2^{t,S} = \{0.6232,0,0.3768,0\}$, $W_3^{t,S} = \{0.3377,0.2533,0,0.4090\}$

Step 16: Place the calculated weights for the criteria in steps 12-15 into weight vectors for each service.

Example

$$W_1^{t,S} = \{0.0291,0.7187,0.2522,0\}\ voice$$
$$W_2^{t,S} = \{0.6232,0,0.3768,0\}\ video$$
$$W_3^{t,S} = \{0.3377,0.2533,0,0.4090\}\ web\ browsing$$

Step 17: To obtain W^S , synthesize(aggregate) $\bar{P}^{t,S}$ and $W_g^{t,S}$ using (13) below (same procedure as step 8).

$$w_j^S = \sum_{g=1}^{Y} \bar{p}_g^{t,S} * w_{g,j}^{t,S} \quad (13)\ [7]$$

Example

$$w_1^S = \bar{p}_1^{t,S} * \bar{w}_{1,1}^{t,S} + \bar{p}_2^{t,S} * \bar{w}_{2,1}^{t,S} + \bar{p}_3^{t,S} * \bar{w}_{3,1}^{t,S} = 0.3300$$

Repeat the same for all criteria then $\boldsymbol{W^S} = \{\mathbf{0.3300}, \mathbf{0.3240}, \mathbf{0.2097}, \mathbf{0.1363}\}$

iv. Comprehensive weight vector calculation

Let $\alpha, \beta, \gamma \in (0,1)$ and $\alpha + \beta + \gamma = 1$ [7] be defined as the weight proportion parameters which represent the contributions of user preferences, network attributes, and service characteristics to the comprehensive weight vector W.

Step 17: determine W by combining W^O, W^U, and W^S with the weight proportion parameters using (14) below. Let $\alpha = 0.2$, $\beta = 0.5$, $\gamma = 0.3$ as this combination resulted in the maximum gain in [7].

$$W = [w_1, \dots, w_N] = \alpha W^U + \beta W^O + \gamma W^S \quad (14)\ [7]$$

Example

$$W = 0.2 * \{0.2452,0.2548,0.2786,0.2214\} + 0.5 * \{0.7317,0.1218,0.0960,0.0505\} + 0.3 * \{0.3300,0.3240,0.2097,0.1363\} = \{\mathbf{0.5139}, \mathbf{0.2090}, \mathbf{0.1666}, \mathbf{0.1104}\}$$

3.4.3 Utility

The inputs to this submodule are the performance ratings of the RAT selection criteria detected by the MMT (m_{ij}) and the normalized determined priority vector for each service ($\bar{P}^{t,S}$). The output of this submodule is the comprehensive utility value matrix of the network attributes $U = (u_{ij})_{MxN}$.

Utility is defined as the degree of consumer satisfaction by a service or good for their need [7]. A utility function refers to the utility derived from using a good or service by the consumer. Utility functions are used to normalize the criteria so that the QoS requirements of the different services can be considered. In this method, Sigmoid utility function is used for criteria that have both upper and lower threshold limits for their

QoS requirements while linear utility functions and inverse functions are used for criteria that have only 1 QoS requirement threshold.
The sigmoid (13-14) and linear utility functions (15-18) for benefit and cost criteria are defined using the following equations below.

$$f(x) = \frac{1}{1+e^{-a(x-b)}} \ \textit{sigmoid benefit} \quad (15)\ [7]$$

$$g(x) = 1 - f(x)\ \textit{sigmoid cost} \quad (16)\ [7]$$

$$u(x) = 1 - \frac{g}{x}\ \textit{inverse proportional benefit} \quad (17)\ [7]$$

$$h(x) = 1 - gx\ \textit{linear cost} \quad (18)\ [7]$$

where a, b, and g are constants, and x is the value of the network attribute of a particular RAT.

Step 1: calculate the utility values of the criteria per service by substituting the appropriate performance rating (m_{ij}) from M^t that was created using (1) in section 3.4.1 into one of the above equations (15-18) according to the QoS requirements of that criterion for that service using table 10 below. Use the appropriate constants (a,b or g) according to the table. Bandwidth was converted to kbps for the calculations for utility.

Table 10: QoS requirements and utility functions per service [7]

Service		Bandwidth(kbps)	Delay(ms)	Packet loss rate(%)	cost
voice	Range	32-64	50-100	<30	<50
	Function	f(x)	g(x)	h(x)	h(x)
	Constants	a=0.25, b=48	a=0.1,b=75	g=1/30	g=1/50
Video	Range	512-5000	75-150	<30	<50
	Function	f(x)	g(x)	h(x)	h(x)
	constants	a=0.003, b=2000	a=0.1,b=112.5	g=1/30	g=1/50
Web browsing	Range	128-1000	250-500	<30	<50
	Function	f(x)	g(x)	h(x)	h(x)
	Constants	a=0.01, b=564	a=0.03,b=375	g=1/30	g=1/50

Example (note BW is in kbps)

$$M^t = \begin{bmatrix} 1350 & 3.5 & 30 & 6 \\ 2400 & 4.5 & 60 & 13 \\ 4500 & 0.5 & 80 & 10 \\ 1000000 & 7 & 13 & 4 \end{bmatrix}$$

For voice for example using table 10, (15) and substituting the appropriate values we get:

$$f(1350) = \frac{1}{1+e^{-0.25(1350-48)}} = 1$$

Step 2: Create the network attribute utility value matrix for each service s^t_g using the calculated network attribute utility values from step 1 using (19) below.

$$U^g = \left(u^g_{ij}\right)_{MxN} \quad (19)\ [7]$$

Where u^g_{ij} is the normalized utility value of criterion cj, RAT r_i for service s^t_g. i=1, …, M and j=1, …, N

Example
Repeating the calculation for all network attributes the following are obtained.

$$U^1(voice) = \begin{bmatrix} 1 & 0.9300 & 0.9890 & 0.800 \\ 1 & 0.9100 & 0.8176 & 0.5667 \\ 1 & 0.9900 & 0.3775 & 0.6667 \\ 1 & 0.8600 & 0.9980 & 0.8667 \end{bmatrix}$$

$$U^2(video) = \begin{bmatrix} 0.1246 & 0.9300 & 0.9997 & 0.800 \\ 0.7685 & 0.9100 & 0.9948 & 0.5667 \\ 0.9994 & 0.9900 & 0.9627 & 0.6667 \\ 1 & 0.8600 & 1 & 0.8667 \end{bmatrix}$$

$$U^3(web\ browsing) = \begin{bmatrix} 0.9996 & 0.9300 & 1 & 0.800 \\ 1 & 0.9100 & 0.9999 & 0.5667 \\ 1 & 0.9900 & 0.9999 & 0.6667 \\ 1 & 0.8600 & 1 & 0.8667 \end{bmatrix}$$

Step 3: To obtain the comprehensive utility value matrix for multiple services $U = \left(u_{ij}\right)_{MxN}$, synthesize (aggregate) $\bar{P}^{t,S}$ (normalized determined priority vector for each service) with U^g (network attribute utility value matrix) using (20) below.

$$u_{ij} = \sum_{g=1}^{Y} u_{ij}^{g} * \bar{p}_{g}^{t,S} \quad (20)\ [7]$$

Example

$$u_{11} = \sum_{g=1}^{Y} u_{11}^{g} * \bar{p}_{g}^{t,S} = \ (u_{11}^{1} * \ \bar{p}_{1}^{t,S}) + (u_{11}^{2} * \ \bar{p}_{2}^{t,S}) + (u_{11}^{3} * \ \bar{p}_{3}^{t,S}) = 0.7081$$

Repeating the above calculation for all attributes we obtain:

$$U = \begin{bmatrix} 0.7081 & 0.9300 & 0.9962 & 0.800 \\ 0.9228 & 0.9100 & 0.9374 & 0.5667 \\ 0.9998 & 0.9900 & 0.7800 & 0.6667 \\ 1 & 0.8600 & 0.9993 & 0.8667 \end{bmatrix}$$

3.4.4 Network Ranking and Selection

The inputs to this submodule are the previously calculated matrix U and the calculated weight vector W. Both are calculated in sections 3.4.2 and 3.4.3 respectively. The output of this module is a single RAT ($r_i \in R$) that represents the best network for vertical handover of the group of services S^t on the MMT.

To score and rank the candidate RATs (all RATs are candidate RATs in this algorithm) TOPSIS was chosen as the MADM technique to determine the scores of the RATs. This is because TOPSIS is a common approach taken to select RATs for group call handoff in the literature (refer to the discussion about [11] in section 2) and is simple to implement.

For TOPSIS, the Euclidean distance between the candidate solution and the PIS(Positive Ideal Solution) and NIS(Negative Ideal Solution) is calculated. The best option/candidate is the one with the shortest distance to the PIS and the longest distance from the NIS. To select the best RAT for Vertical handover of a group of calls the following steps are taken.

Step 1: Create the normalized decision matrix. In this case, it is the same as U.

$$Normalized\ decision\ matrix = U$$

Example

$$U = \begin{bmatrix} 0.7081 & 0.9300 & 0.9962 & 0.800 \\ 0.9228 & 0.9100 & 0.9374 & 0.5667 \\ 0.9998 & 0.9900 & 0.7800 & 0.6667 \\ 1 & 0.8600 & 0.9993 & 0.8667 \end{bmatrix}$$

Step 2: Create the weighted normalized decision matrix D by multiplying each row of U with weight vector W of the network attributes.

$$D = \left(d_{ij}\right)_{MxN} \quad (21)\ [7]$$

$$d_{ij} = u_{ij} * w_j \text{ i=1, …, M and j=1, …, N} \quad (22)\ [7]$$

Where w_j is the comprehensive weight and u_{ij} is the comprehensive utility value of criterion c_j. For RAT r_i.
Example

$$D = \begin{bmatrix} 0.3639 & 0.1944 & 0.1660 & 0.0883 \\ 0.4743 & 0.1902 & 0.1562 & 0.0626 \\ 0.5138 & 0.2069 & 0.1300 & 0.0736 \\ 0.5139 & 0.1798 & 0.1665 & 0.0957 \end{bmatrix}$$

Step 3: calculate the PIS (D^+) and the NIS (D^-) using (23) below.

$$D^+ = \left[d_1^+, d_2^+, \dots, d_j^+, \dots, d_N^+\right]$$
$$d_j^+ = \max\,(d_{ij}, i = 1, \dots M)$$
$$D^- = \left[d_1^-, d_2^-, \dots, d_j^-, \dots, d_N^-\right] \quad (23)\ [7]$$
$$d_j^- = \min\,(d_{ij}, i = 1, \dots M)$$

Where d_j^+ indicates the best value, d_j^- indicates the worst value of criterion c_j among all RATs. For the benefit criteria $d_j^+ = max(d_{ij}, i = 1, \dots M)$ and $d_j^- = \min\,(d_{ij}, i = 1, \dots M)$. For cost criteria, it is the opposite.

Example
For bandwidth

$$d_1^+ = \max(d_{i1}, i = 1, \dots 4) = 0.5139$$
$$d_1^- = \min(d_{i1}, i = 1, \dots 4) = 0.3639$$

Repeating for all criteria we get :

$$D^+ = [0.5139, 0.1798, 0.13, 0.0626]$$
$$D^- = [0.3639, 0.2069, 0.1665, 0.0957]$$

Step 4: calculate S_i^+ and S_i^- of each RAT r_i to D^+ and D^- using (24) below. The equations used differ from those in [7] as they used a modified version of TOPSIS whereas the proposed algorithm in this report used the ordinary TOPSIS.

$$S_i^+ = \sqrt{\sum_{j=1}^{N} \left(d_{ij} - d_j^+\right)^2}$$

$$S_i^- = \sqrt{\sum_{j=1}^{N} \left(d_{ij} - d_j^-\right)^2} \quad (24)\ [7]$$

Where i = 1,…, M

Example

$$S_1^+ = \sqrt{\sum_{j=1}^{N} \left(d_{ij} - d_j^+\right)^2} = \sqrt{\left(d_{ij} - d_j^+\right)^2 + \cdots + (d_{1N} - d_N^+)^2} = 0.1571$$

$$S_1^- = \sqrt{\sum_{j=1}^{N} \left(d_{ij} - d_j^-\right)^2} = \sqrt{\left(d_{ij} - d_j^-\right)^2 + \cdots + (d_{1N} - d_N^-)^2} = 0.0146$$

Repeating the above calculations for all S_i values we obtain table 11 below.

Table 11: Example S_i values

i	S_i^+	S_i^-
1	0.1571	0.0146
2	0.0487	0.1169
3	0.0293	0.1559
4	0.0493	0.1525

Step 5: Calculate the score sc_i of RAT r_i using (25) below.

$$sc_i = \frac{S_i^-}{S_i^- + S_i^+} \quad (25)$$

Example

For RAT r_1 (3G)

$$sc_1 = \frac{S_1^-}{S_1^- + S_1^+} = 0.0848$$

Step 6: Store the scores of each RAT in a vector SC = $\{sc_1, \dots, sc_i, \dots, sc_M\}$

Example

Repeating the above calculation for each RAT we get:

$$\text{SC} = \{0.0848, 0.7060, 0.8416, 0.7556\}$$

Step 7: Select the RAT with the greatest score.

$$RAT = \max(SC)$$

Example
For the example scenario, the best RAT would be **RAT 3 (WLAN)** as it has the highest score(0.8416).
To minimize the number of unnecessary vertical handoffs (ping-ponging), a threshold σ is introduced to implement this. If there is no RAT connected to the MMT initially (at time 0) the RAT with the highest score is always selected. Otherwise, if the MMT is already currently connected to RAT r_j that has score sc_j and there is a RAT r_i with the highest score sc_i $(i \neq j)$ then the following condition is tested (NB: $\sigma > 1$) Using the same value as the threshold in [7] the threshold used in this algorithm is $\sigma = 1.3$

$$\frac{sc_i}{sc_j} > \sigma \quad (26)$$

If the above condition is true, the MMT's calls are handed over to RAT r_i otherwise it maintains its current connection.

Example
Using the above example scenario, assuming the MMT is connected to RAT 1 initially i.e., $sc_j = 0.0848$, RAT 3 is the highest scoring RAT with score $sc_i = 0.8416$. Therefore using (26)

$$\frac{0.8416}{0.0848} = 9.925 > 1.3$$

Therefore, the MMT calls would be handed over to RAT 3.
The flowchart in figure 4 below provides a summary of the algorithm designed for RAT selection discussed above.

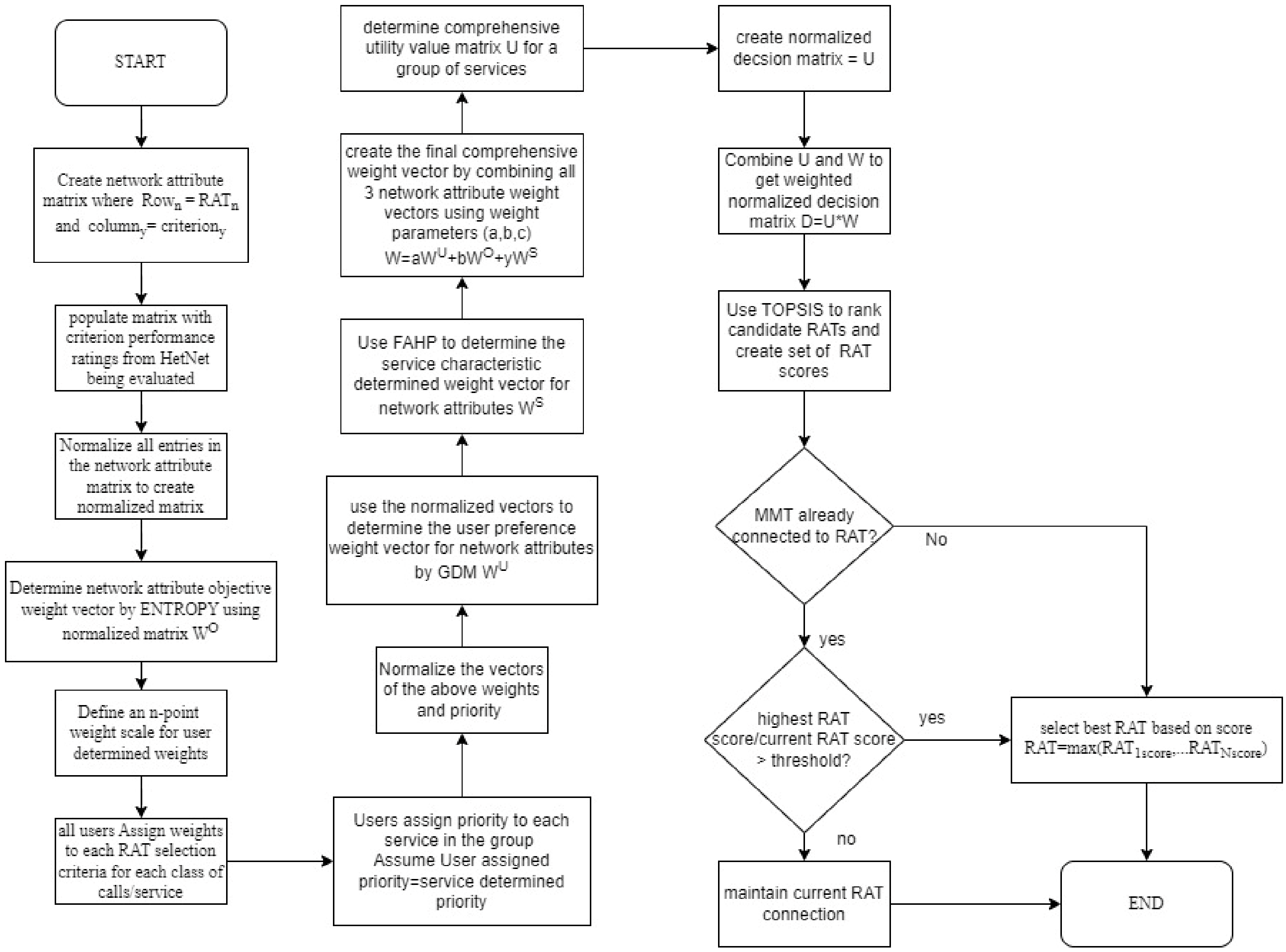


Figure 4: Summary of proposed algorithm

3.4.5 Complexity analysis of algorithm

Determining the time and space complexity of a designed algorithm is important because it provides a means of understanding how an algorithm performs as the input size increases. The time complexity and space complexity of an algorithm inform the designers' decisions on how to optimize the algorithm to use computing resources efficiently. A big-O analysis was conducted for the designed algorithm. Since it uses the same algorithms in its submodules as [7], the designed algorithm in this report has a time complexity of $O(N^3)$ and space complexity of $O(N^2)$. Despite the high time complexity, the number of RATs and criteria is small(4 each) therefore the MMTs have enough computing power to execute the algorithm quickly.

3.5 Simulation and Experiments

The following section outlines the experiments that were performed to evaluate the performance of the algorithm. The code implementation for all simulations can be found in the appendix. From the literature [7-18], the number of vertical handovers, unnecessary handovers and calls admitted into each RAT are important metrics used to evaluate the performance of a RAT selection algorithm. For group calls investigating the effect of varying service priorities of the calls in the group is a common approach to evaluate the performance of the algorithm. Therefore, the following experiments were conducted.

3.5.1 Hardware and software specifications

The simulations were implemented using MATLAB. The Processor used to execute the simulations was an Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz Installed RAM 12,0 GB (11,9 GB usable), on a Lenovo laptop that had a Windows 11(Home Single language version) 64-bit operating system, x64-based processor.

3.5.2 Experiment 1: Determination of weight proportion parameters

The authors in [7] defined the weight proportion parameters as $\alpha, \beta, \gamma \in (0,1)$ and $\alpha + \beta + \gamma = 1$. They conducted experiments to determine what values to use for them. They determined the best combination to use was "α = 0.2, β = 0.5, γ = 0.3" [7] because it provided the highest average gain. They defined gain according to (27) below.

$$gain = \sum_{j=1}^{N} u_{ij} * w_j \quad (27)\ [7]$$

Where $1 \leq i \leq M$ and u_{ij} is the comprehensive utility value of criterion c_j for multiple services and w_j is the comprehensive weight of attribute c_j .

For this report, it was assumed that the same combination as the one used in [7] was the best one to use. However, to verify this, the gain for each combination of parameters in table 12 below was calculated and the highest mean gain was determined for 1000 users(MMTs). The user-specified weights were randomly allocated, the service priority vector was kept the same for all users and the service priorities were equal $P^{t,S}$=(3,3,3). The network conditions at time t(t is an arbitrary point in time we let the network selection happen) used values from table 3. The objective weights, service characteristic determined weights and comprehensive utility matrices were the same for all users in each scenario. The simulation was run 3 times for the 1000 MMTs and the mean gains from each scenario were calculated. The code implementation is available in appendix A. The calculation of W^S by FAHP is available in appendix B.

Table 12: parameter combinations tested [7]

Scenario	α	β	γ
1	0.2	0.3	0.5
2	0.3	0.2	0.5
3	0.2	0.5	0.3
4	0.3	0.5	0.2
5	0.5	0.2	0.3
6	0.5	0.3	0.2
7	1/3	1/3	1/3

The above experiment was then repeated for each scenario but using dynamic network conditions, i.e., the criteria performance ratings for each RAT were varied randomly using the ranges of values specified in table 2 for each iteration of the loop in the simulation. This was done to make the simulation more realistic as network conditions are always changing (not static in reality) to simulate mobility.

To consider all possible parameter combinations (to 1dp), the gain of all possible combinations of parameters (1331 combinations in total) was calculated for a single MMT using dynamic network conditions. This calculation was repeated 1000 times so that each iteration of the loop was equivalent to the calculation for a different MMT. The combination which had the highest gain in each iteration was recorded. The code implementation is available in appendix C.

3.5.3 Experiment 2: Effect of different service priorities on algorithm performance

For this experiment, the performance of the algorithm was evaluated by varying the priorities of the services running simultaneously on the MMT ($P^{t,U}$). Ten priority scenes were simulated as shown in table 13 below, of which the first six and the last one were adapted from [7].

Table 13: Service priority scenarios [7]

Scenario	Priority of the service		
	Voice	Video	Web browsing
1	5	1	3
2	5	3	1
3	1	5	3
4	3	5	1
5	1	3	5
6	3	1	5

7	5	1	1
8	1	5	1
9	1	1	5
10	5	5	5

Where 1=very low priority, 2=low priority, 3=medium priority, 4= high priority and 5=very high priority for the service in the group of services on the MMT.

For each scenario, the RAT selection was simulated for 1000 users under dynamic network conditions by randomly varying the values of the criteria performance ratings from the range of values in table 2 for each user. The user-specified weights were randomly allocated for each user, and the objective weights were unique for each user. All MMTs had the same service priority vector $P^{t,U}$ specified by table 13 for a given scenario. The number of users (MMTs) admitted into each RAT was measured for each scenario. The combination of weight proportion parameters that yielded the highest gain from experiment 1 was used for this simulation (see section 4.1.1). The code implementation is available in appendix D.

The above experiment was repeated twice to test if user preferences had been considered by the algorithm, the user-determined weights for each service defined in section 3.4.2 were biased to be higher according to what criteria they deemed more important. In the first instance, the criterion they deemed more important had a higher weight for all services. In simulation, this was done by randomly assigning a weight between 7 and 9 for the important criterion, while assigning a weight between 1 and 3 for the other 3 criteria. Then in the second instance, all the criteria for one service were assigned higher weights to stress the importance of that service to the users while the criteria for the other services were assigned low weights. The code implementation is available in appendix E. Table 14 below provides a summary of the cases that were simulated.

Table 14: Varying service priorities with specific user preferences

Case Number	Criterion/service more important to user
1	Bandwidth more important
2	Cost more important
3	Delay more important
4	Packet loss rate more important
5	Voice more important
6	Video more important
7	Web browsing more important

3.5.4 Experiment 3: Investigating threshold on number of handovers

To measure the number of handovers and the effect of varying the threshold, handover situations needed to be simulated. This was achieved by implementing the pseudocode used to determine the number of handovers in [7]. Each MMT was assigned a variable (called current_RAT in the simulation) which represented the RAT it was connected to initially (when the algorithm was triggered). A value of 1 implied that the MMT was connected to RAT 1 initially for example. The priority of all services was equal and was the same for all MMTs. The user-specified weights for each criterion per service were randomly allocated. The network attribute values were varied for each iteration of the loop in the simulation using the range of values in table 2(dynamic network conditions). Also, the initial RAT connected to an MMT was randomized at the start of each iteration. The simulation had 1000 iterations where each iteration of the loop represented the potential handover situation for a single user. The number of handovers was measured. The pseudocode adapted from [7] is shown below in figure 5.

Handover simulation

```
Num_Handovers=0; σ = 1.3;
Input Num_users; α = 0.2; β = 0.5; γ = 0.3;
for i=1:Num_Users
current_RAT=random(1,4);
Mt=randomValues(RAT1,RAT2,RAT3,RAT4);
Wo=entropy(Mt);
```

```
Wu= getWu(Pu,UserWeights);
Ws=FAHP(Ps,FCMmatrices);
W= αWu + βWo + γWs;
U=utility(Mt);
D=U*W;
RAT_Scores=TOPSIS(D);
Best_RAT=max(RAT_Scores);
If(Best_RAT== current_RAT)
   current_RAT=Best_RAT;
If(score(Best_RAT)/score(current_RAT)> σ)
 Num_Handovers+=1;
 current_RAT=Best_RAT
else
 current_RAT=current_RAT;
end
```

Figure 5: Pseudocode for handover simulation [7]

The procedure was repeated for a constant threshold of σ =1.3 but the number of users (iterations) was varied as shown in table 15.

Table 15: Scenarios for varying numbers of users

Scenario	1	2	3	4	5	6	7	8	9	10
No of users	100	200	300	400	500	600	700	800	900	1000

The above experiment was repeated but the threshold was varied as shown in table 16 below to observe the effect of varying the threshold. The number of users (iterations) was kept constant at 1000. The choice of thresholds was arbitrary. The code implementation is available in appendix F.

Table 16: Scenarios for varying thresholds

Scenario	1	2	3	4	5	6	7	8	9	10	11	12	13
Threshold	0.03	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3.0	10.0	15.0

The final investigation for this experiment involved investigating the number of unnecessary handovers. An unnecessary handover is defined as the algorithm causing an MMT to handover when its last connected RAT is the same as the best RAT in the current iteration [7]. These handovers are unnecessary because they provide no additional benefit or improvement to QoS for the MMT. For this simulation dynamic network conditions were simulated by using random values from the ranges of values in table 2. The priority of each service was kept constant and equal. The threshold was kept constant at σ=1.3 and the program was executed 10 times where each execution simulated selections/handovers for 1000 MMTs. The number of handovers, unnecessary handovers and percentage of unnecessary handovers was measured and calculated. The implementation for the simulation was adapted from the pseudocode in [7] shown in figure 6 below. The code implementation is available in appendix G.

Unnecessary Handover simulation

```
Num_Handovers=0; σ = 1.3;
Num_Unecessary=0;
Last_RAT=0;
 t1=t2=0;
current_RAT=0;
Input Num_users; α = 0.2; β = 0.5; γ = 0.3;
for i=1:Num_Users
current_RAT=random(1,4);
Mt=randomValues(RAT1,RAT2,RAT3,RAT4);
Wo=entropy(Mt);
Wu= getWu(Pu,UserWeights);
Ws=FAHP(Ps,FCMmatrices);
```

```
W= αWu + βWo + γWs;
U=utility(Mt);
D=U*W;
RAT_Scores=TOPSIS(D);
Best_RAT=max(RAT_Scores);
if(i==1) //first iteration
   Last_RAT=current_RAT;
   current_RAT=Best_RAT;
   t2=t1;
   t1=1;
else
 if(current_RAT==Best_RAT OR (current_RAT!=Best_RAT AND
score(Best_RAT)/score(current_RAT)<= σ))
  Best_RAT=current_RAT;
 else if(Last_RAT!=0 AND Last_RAT==Best_RAT AND (t1-t2)==2)
    Num_Unecessary+=1;
 Num_Handovers+=1;
 current_RAT=Best_RAT
 current_RAT=current_RAT;
end
```

Figure 6: Pseudocode for unnecessary handovers [7]

The percentage of unnecessary handovers was calculated using (28) below.

$$\%\, unnecessary\ HO = \frac{number\ of\ unnecessary\ handoffs}{total\ number\ of\ handoffs} * 100 \qquad (28)$$

4. Results

4.1.1 Experiment 1: Determination of weight proportion parameters

Applying the methods to calculate the mean gain for the scenarios in section 3.5.2 the following results were obtained as shown in tables 17 and 18 below.

Table 17: Mean gains for 1000 users in table 3 network conditions

Scenario	α	β	γ	Mean gain for 1000 MMTs (static network conditions)			
				1	2	3	Mean
1	0.2	0.3	0.5	3.5628	3.5635	3.5632	3.5631
2	0.3	0.2	0.5	3.5501	3.5511	3.5506	3.5506
3	0.2	0.5	0.3	3.5711	3.5717	3.5714	3.5714
4	0.3	0.5	0.2	3.5625	3.5635	3.5630	3.5630
5	0.5	0.2	0.3	3.5330	3.5345	3.5338	3.5338
6	0.5	0.3	0.2	3.5371	3.5387	3.5379	3.5379
7	1/3	1/3	1/3	3.5528	3.5538	3.5533	3.5533

Table 18: Mean gains for 1000 users in dynamic network conditions

Scenario	α	β	γ	Mean gain for 1000 MMTs (Dynamic network conditions)			
				1	2	3	Mean
1	0.2	0.3	0.5	3.5725	3.5911	3.6423	3.6020
2	0.3	0.2	0.5	3.5719	3.5846	3.6281	3.5949
3	0.2	0.5	0.3	3.5648	3.5946	3.6552	3.6049
4	0.3	0.5	0.2	3.5603	3.5899	3.6474	3.5992
5	0.5	0.2	0.3	3.5628	3.5753	3.6126	3.5836
6	0.5	0.3	0.2	3.5589	3.5770	3.6191	3.5850
7	1/3	1/3	1/3	3.5652	3.5854	3.6341	3.5949

From the results in table 17 and table 18, the combination that had the highest average gain for 1000 MMTs was combination 3 $(\boldsymbol{\alpha = 0.2},\ \boldsymbol{\beta = 0.5}, \boldsymbol{\gamma = 0.3})$ for both static network conditions and dynamic network conditions. Therefore, this combination was used for all subsequent experiments.

For the third simulation in appendix B of running through all possible combinations of parameters, the best combination was found to be $(\boldsymbol{\alpha = 0},\ \boldsymbol{\beta = 0},\ \boldsymbol{\gamma = 1})$. However, this value was not used for reasons discussed in (section 5.1.1). The best usable value from all combinations was found to be $(\boldsymbol{\alpha = 0.1},\ \boldsymbol{\beta = 0.8},\ \boldsymbol{\gamma = 0.1})$. However, this value also wasn't used for the subsequent experiments for reasons discussed in (section 5.1.1).

4.1.2 Experiment 2: Effect of different service priorities on algorithm performance

Applying the different service priorities to each simulation scenario the number of users admitted into each RAT was measured and the results were plotted as bar charts using MATLAB. Figures 7 to 16 show the results for each scenario and figure 17 shows the results for all scenarios under dynamic network conditions.

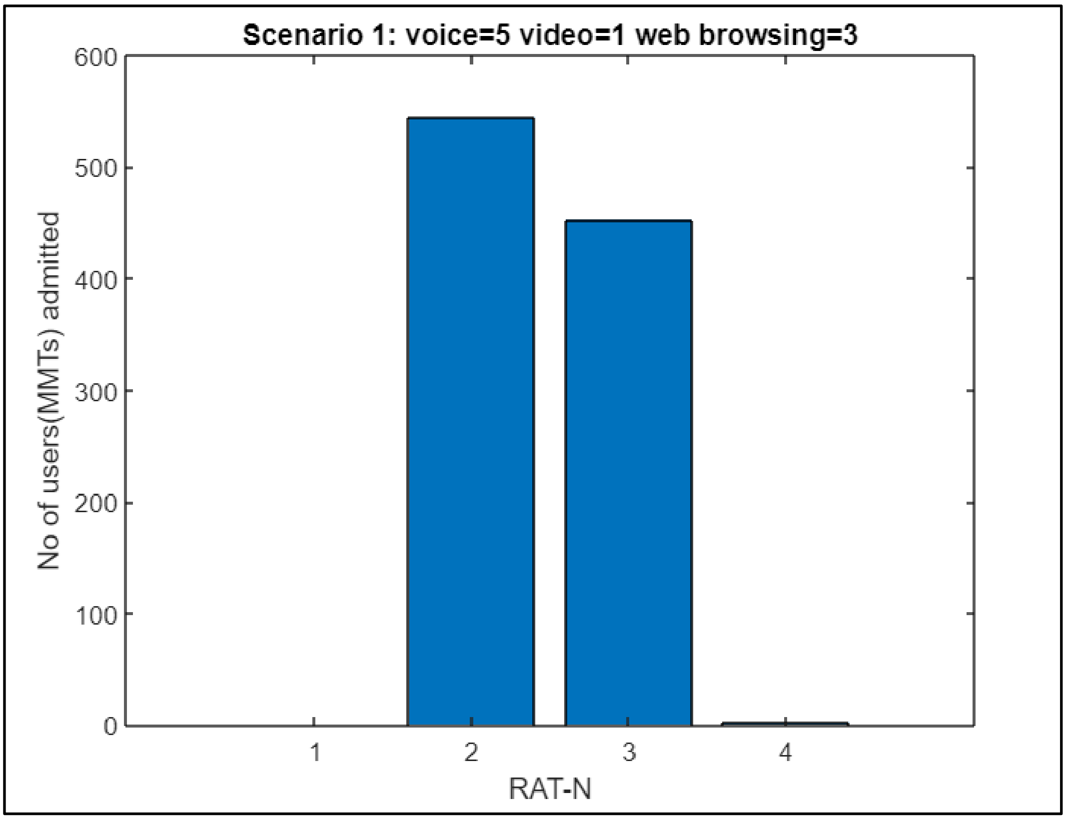


Figure 7: Number of users admitted into each RAT for scenario 1

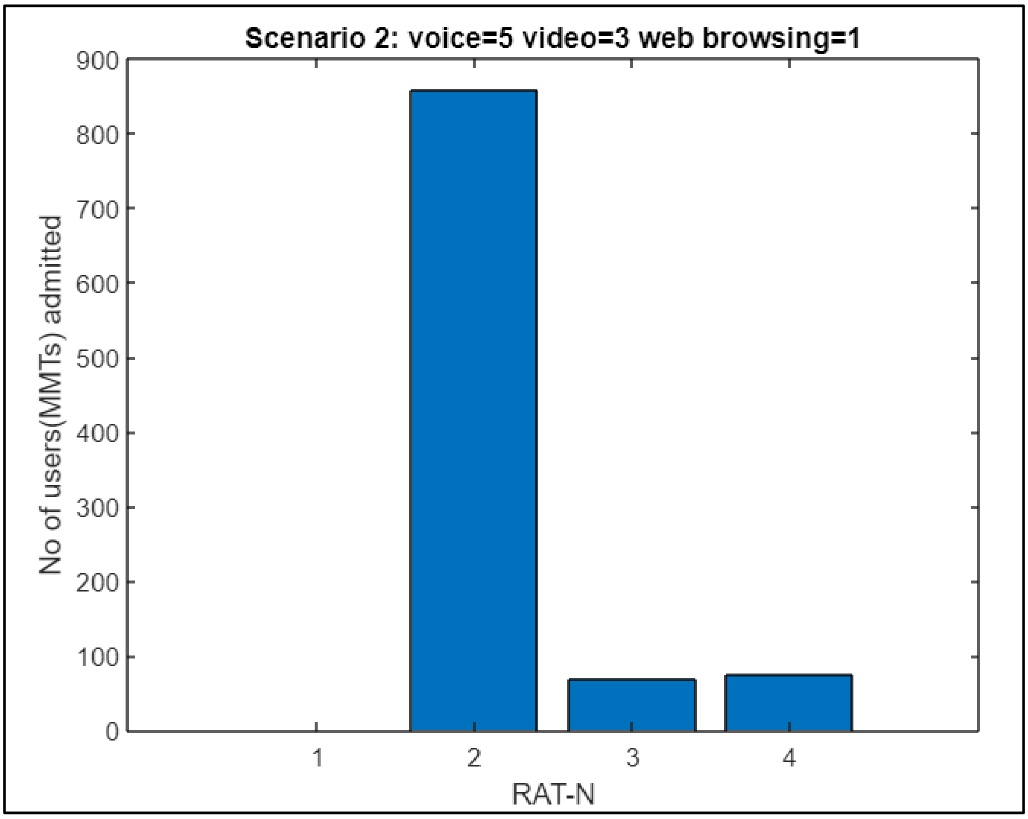


Figure 8: Number of users admitted into each RAT for scenario 2

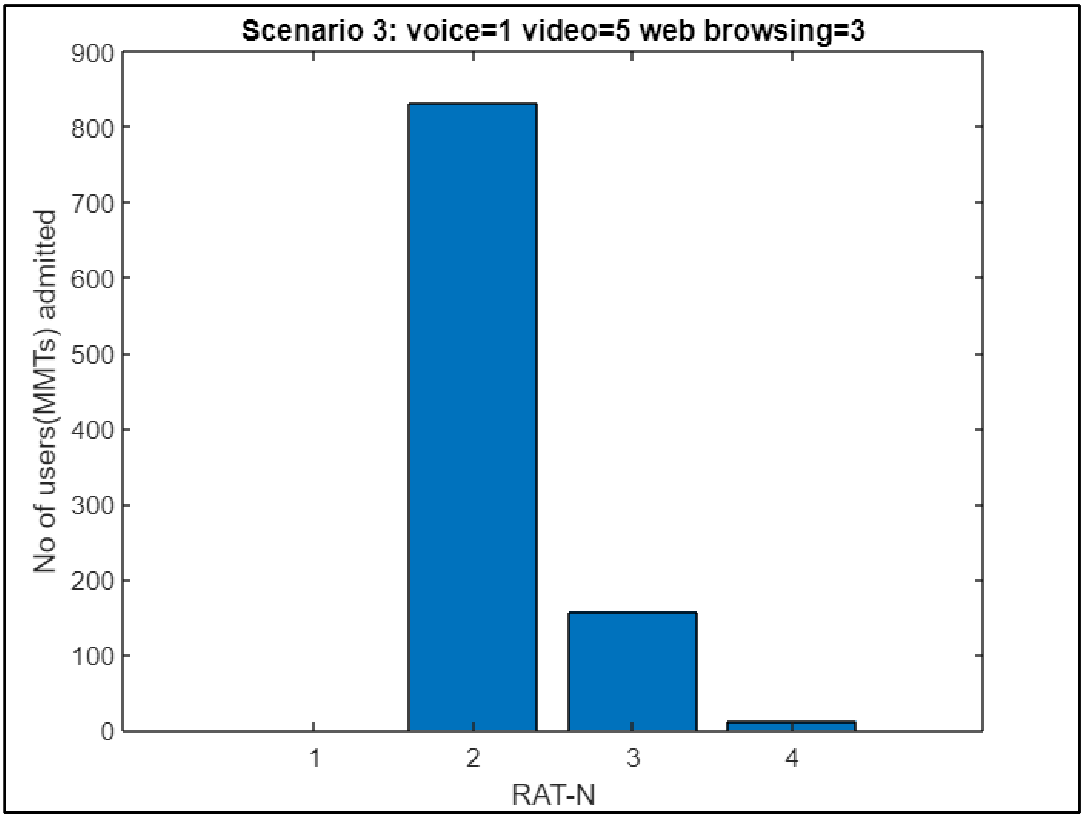


Figure 9: Number of users admitted into each RAT for scenario 3

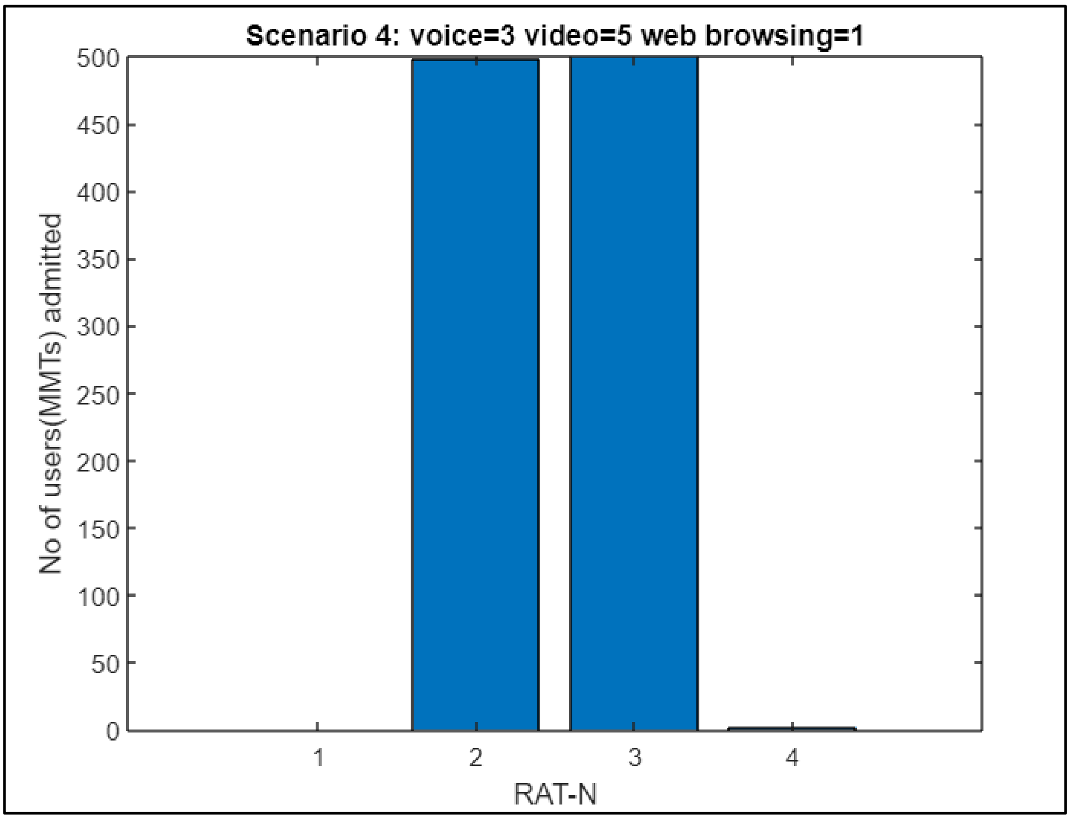


Figure 10: Number of users admitted into each RAT for scenario 4

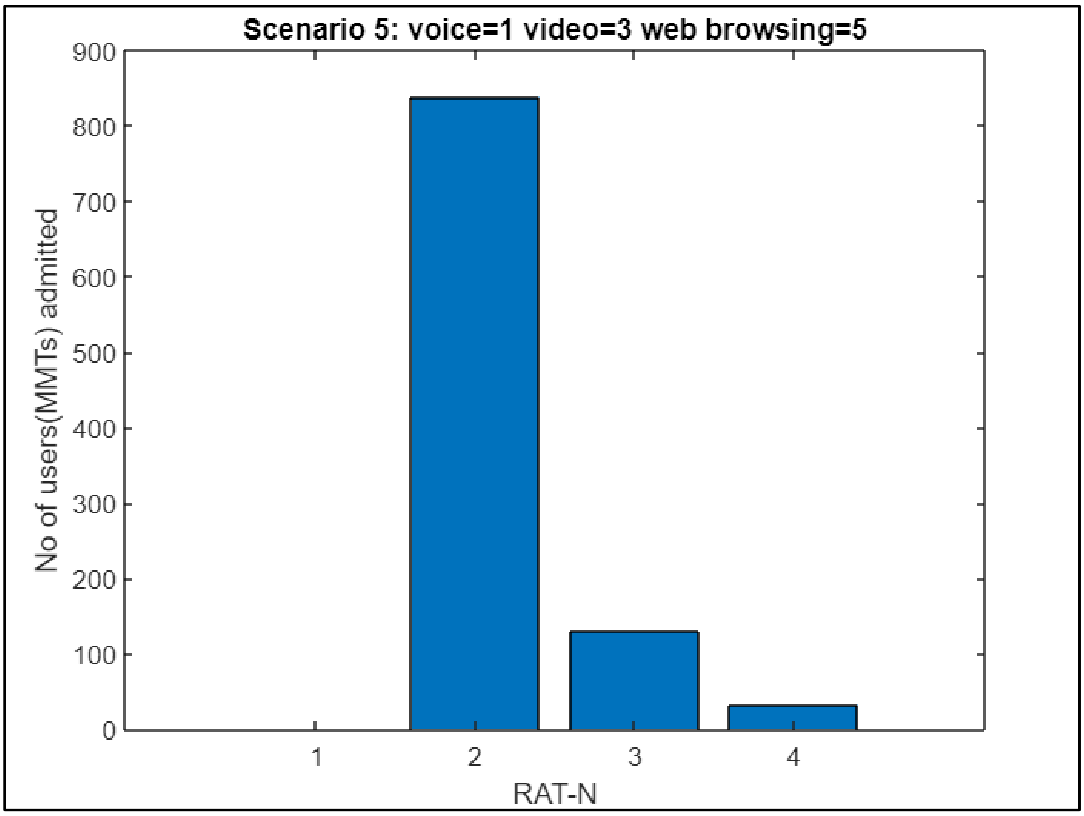


Figure 11: Number of users admitted into each RAT for scenario 5

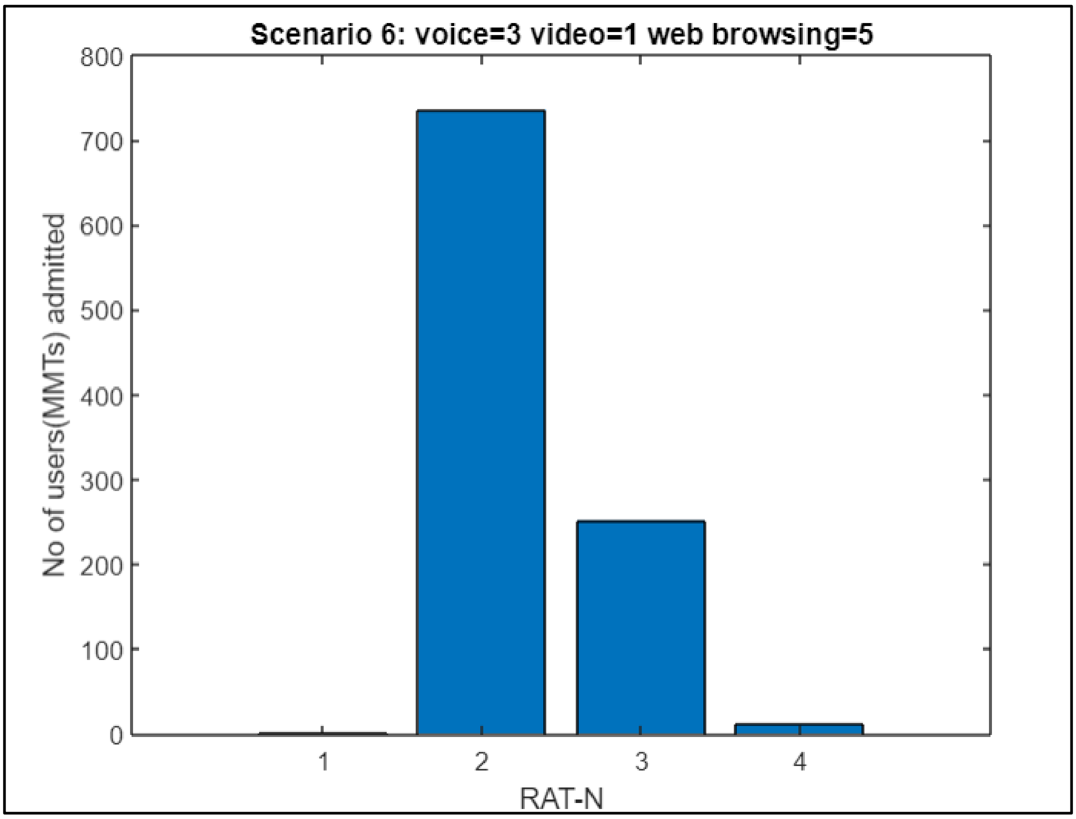


Figure 12: Number of users admitted into each RAT for scenario 6

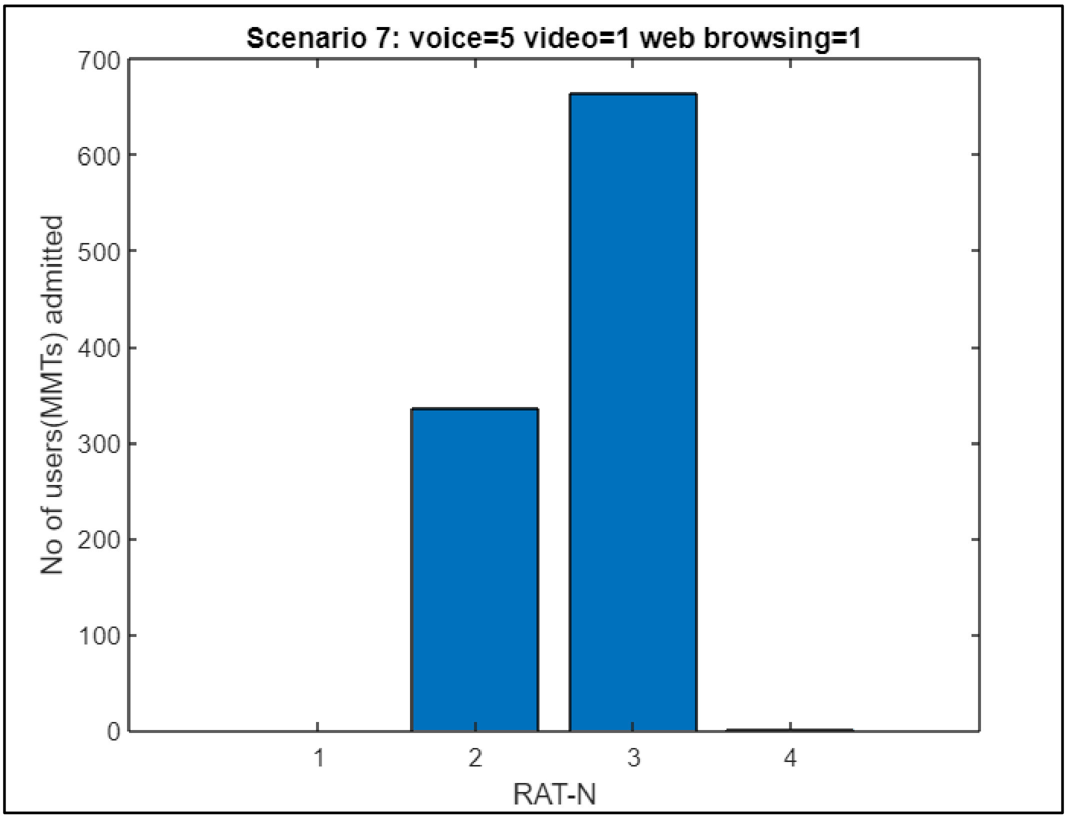


Figure 13: Number of users admitted into each RAT for scenario 7

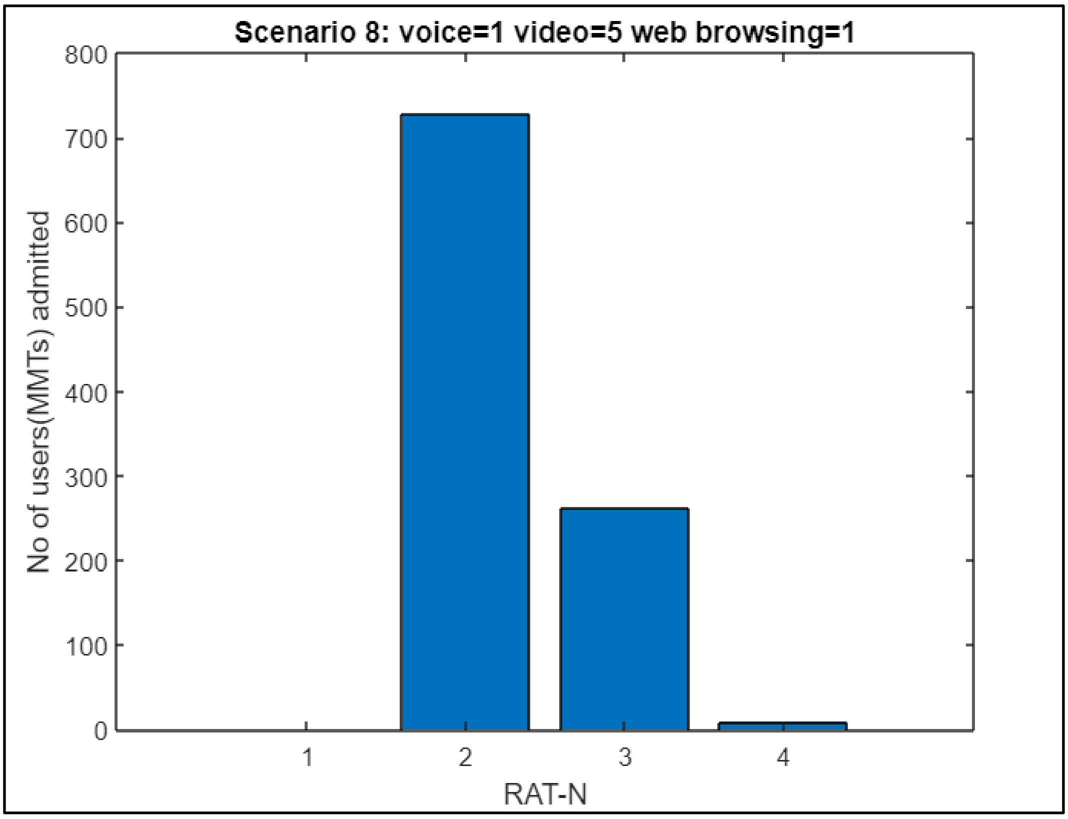


Figure 14: Number of users admitted into each RAT for scenario 8

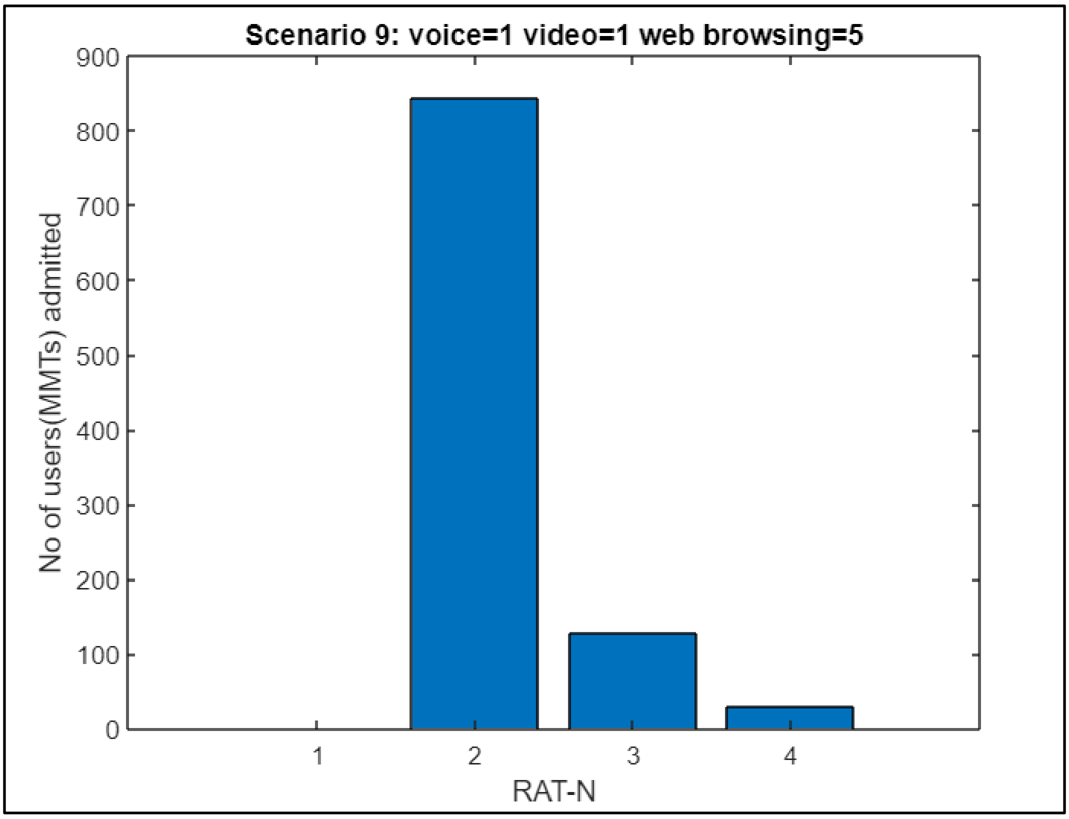


Figure 15: Number of users admitted into each RAT for scenario 9

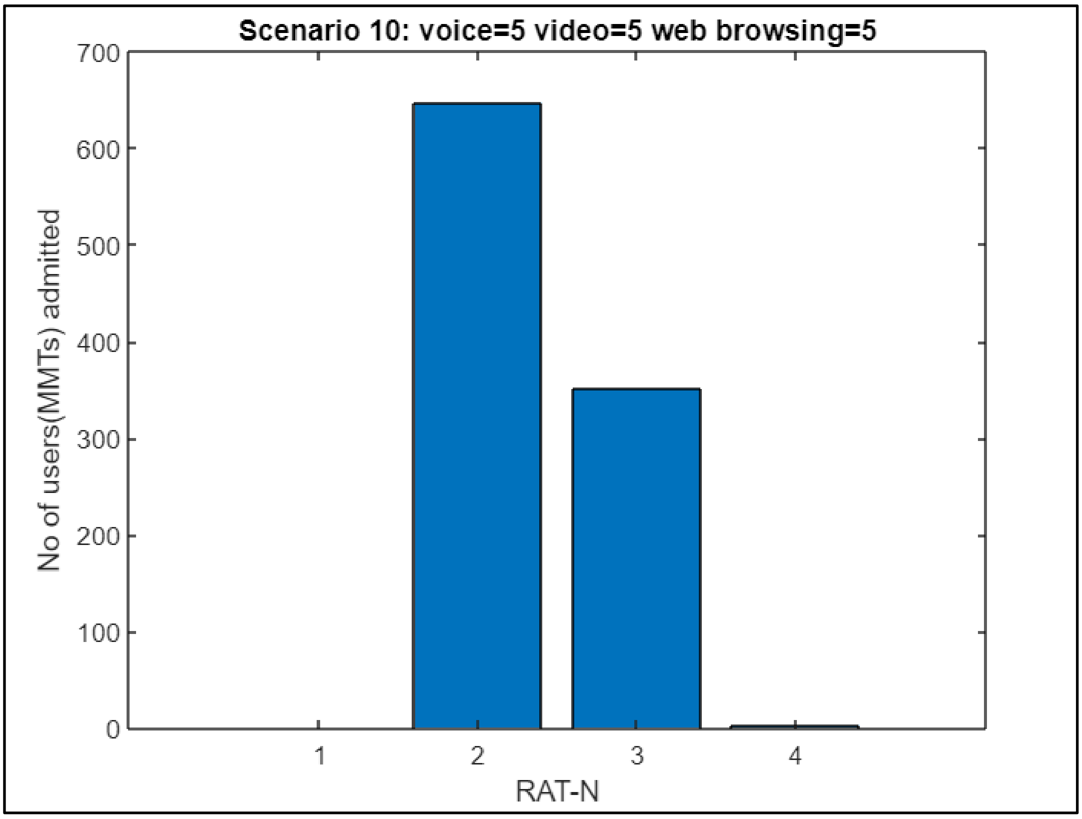


Figure 16: Number of users admitted into each RAT for scenario 10

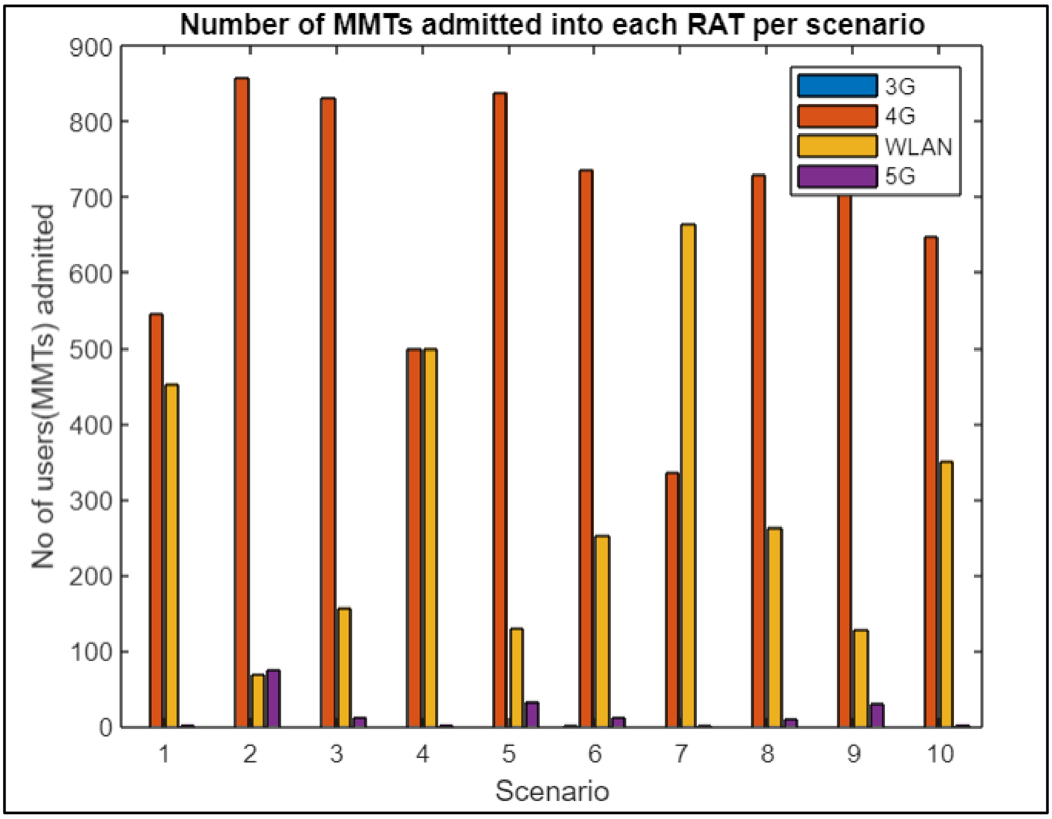


Figure 17: Number of users admitted for all scenarios

Figures 18-24 show the results of all the cases that were tested in the second part of experiment 2. Each figure shows the results for the number of users admitted into each RAT for all scenarios.

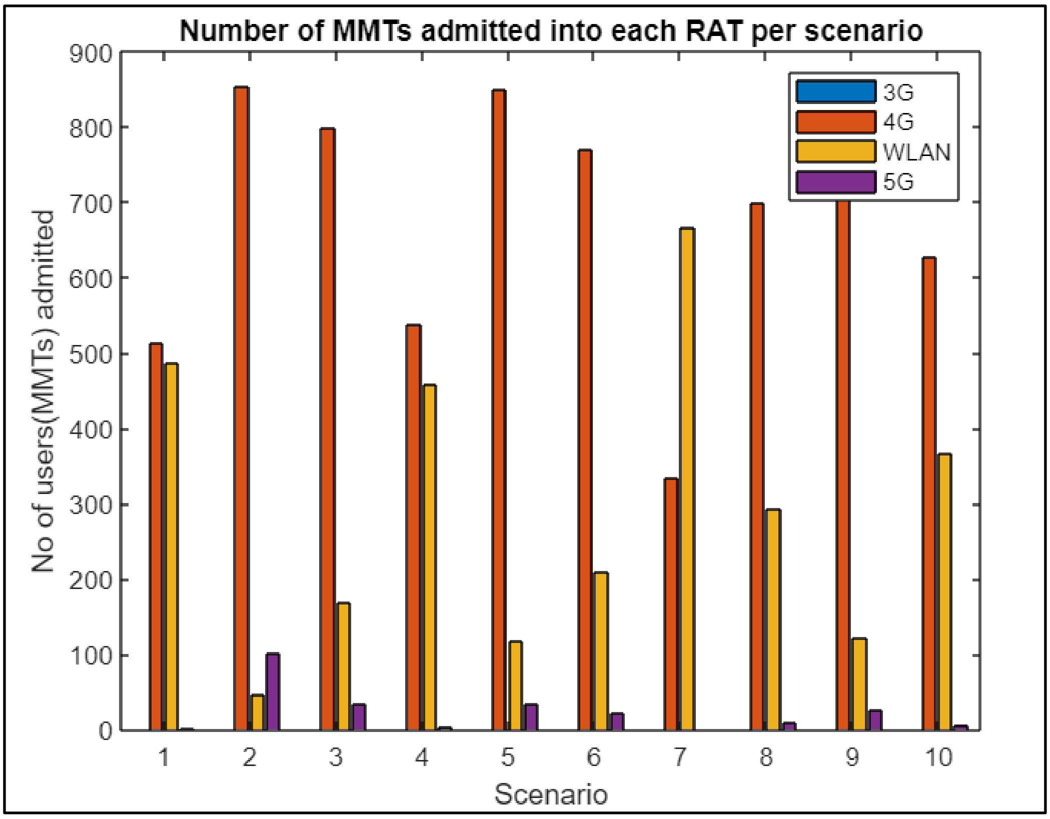


Figure 18: Users admitted in all scenarios for case 1

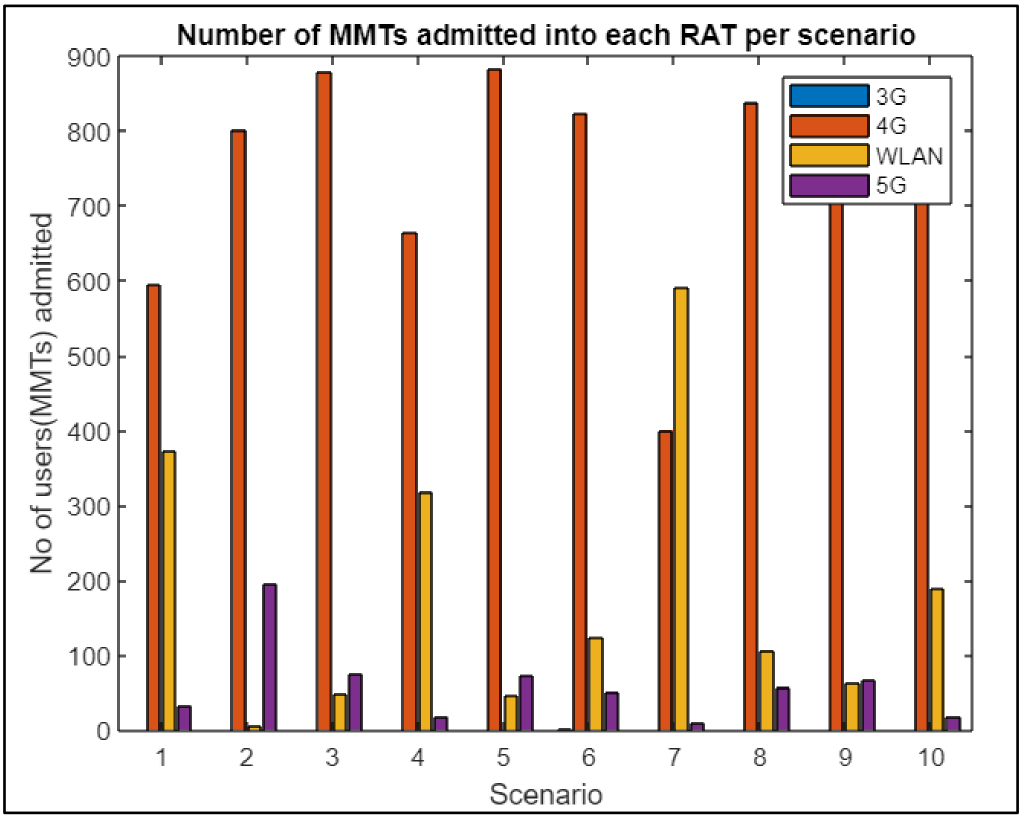


Figure 19: Users admitted in all scenarios for case 2

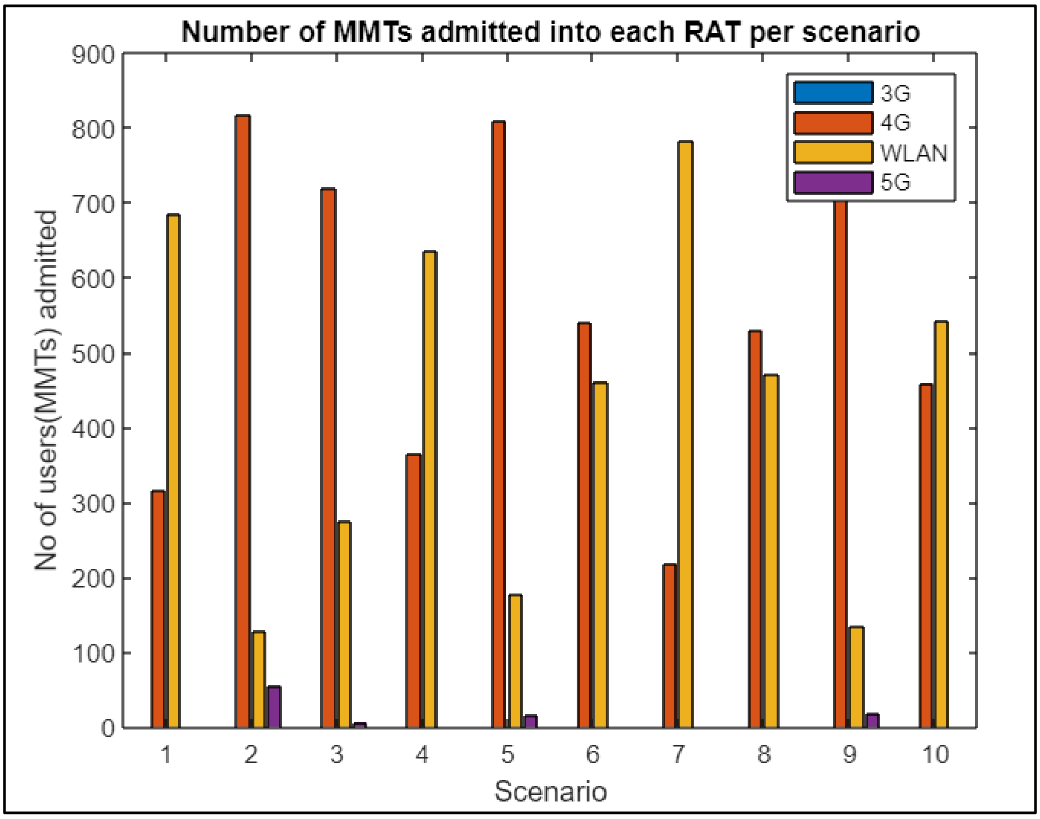


Figure 20: Users admitted in all scenarios for case 3

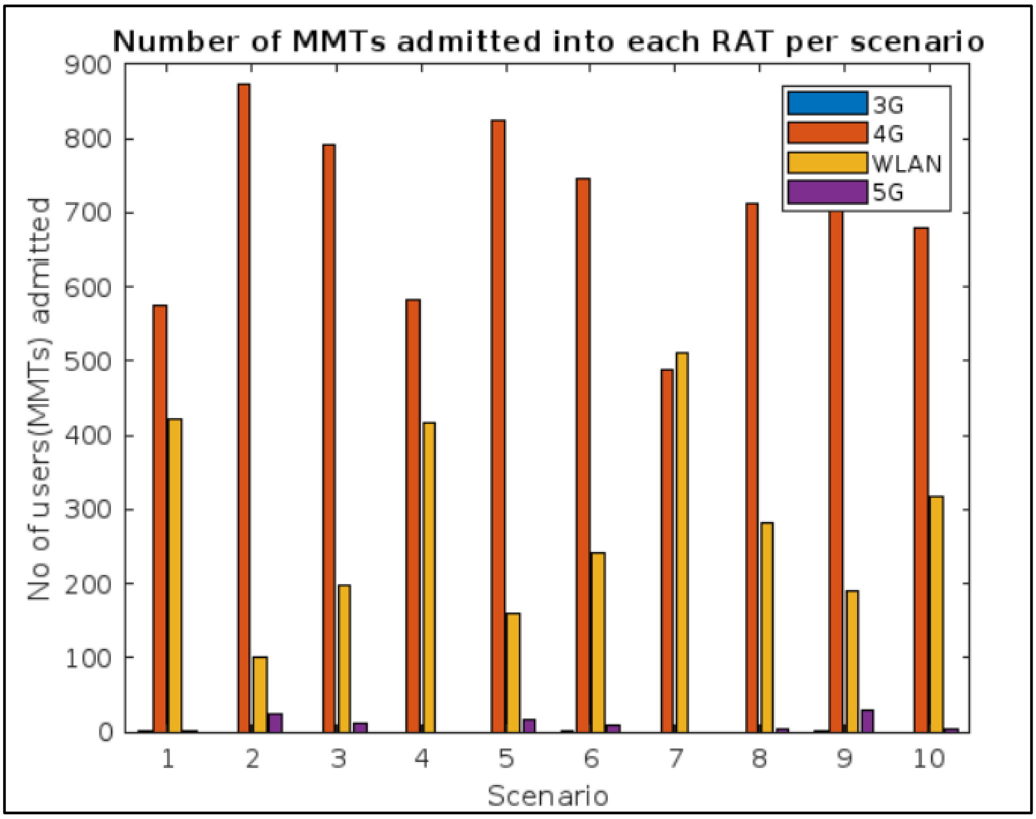


Figure 21: Users admitted in all scenarios for case 4

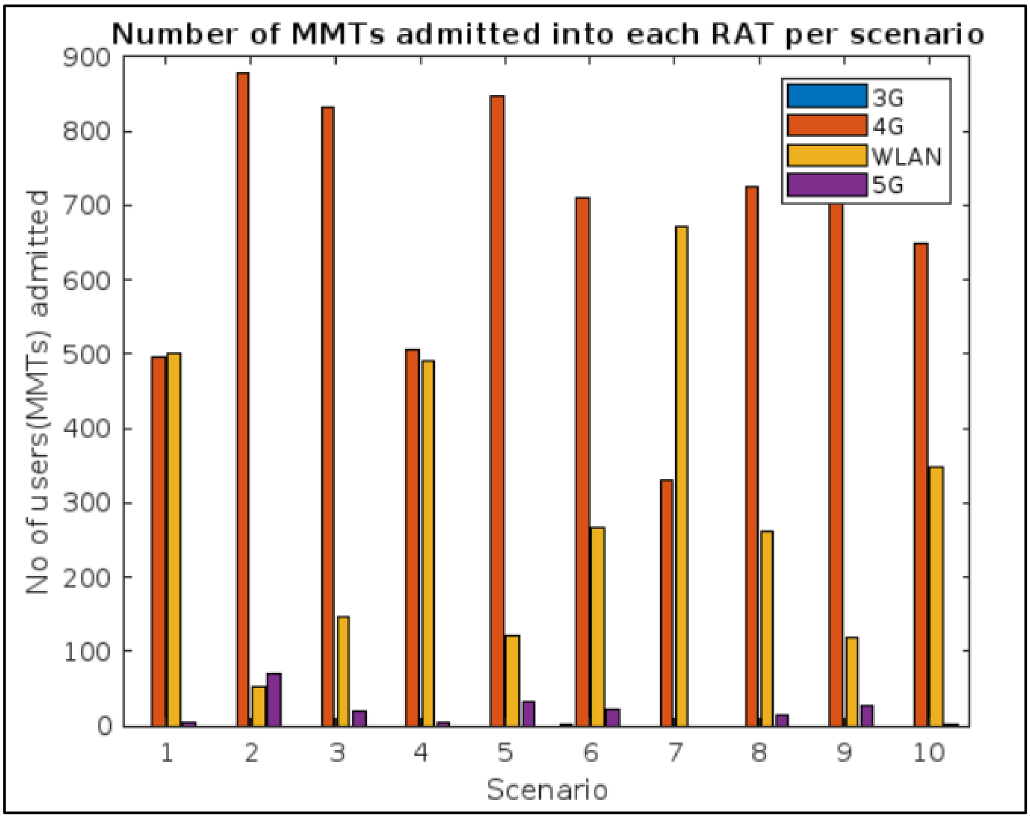


Figure 22: Users admitted in all scenarios for case 5

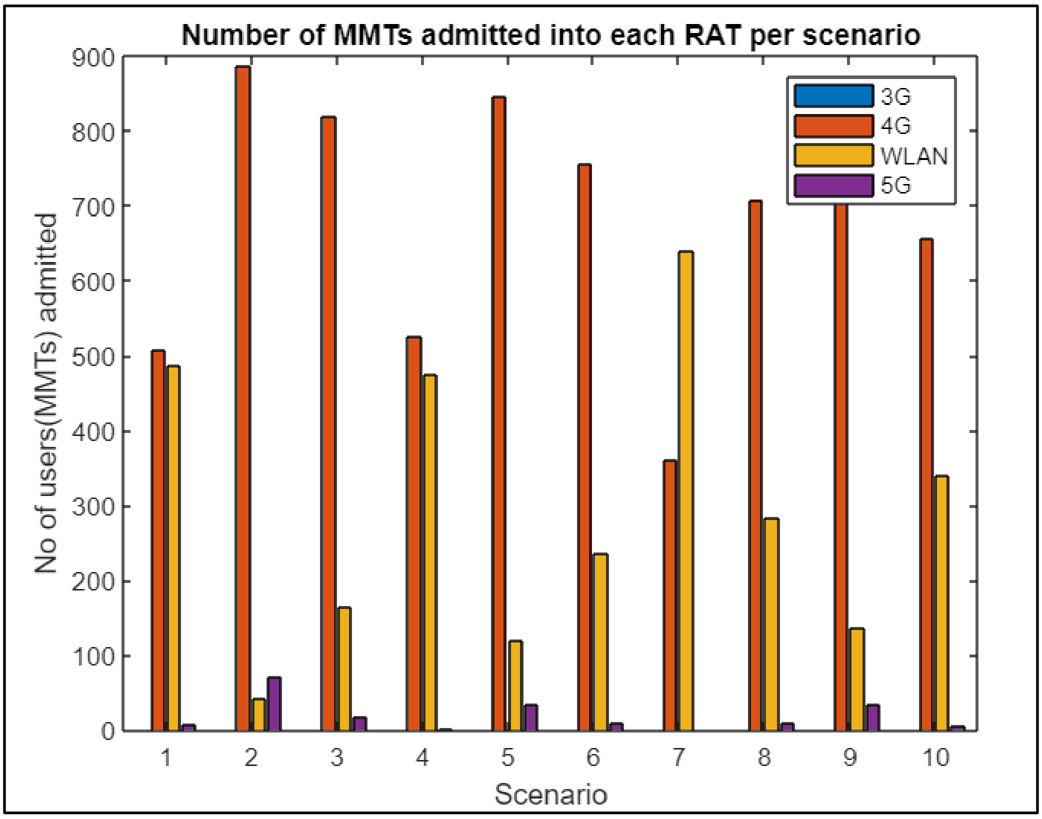


Figure 23: Users admitted in all scenarios for case 6

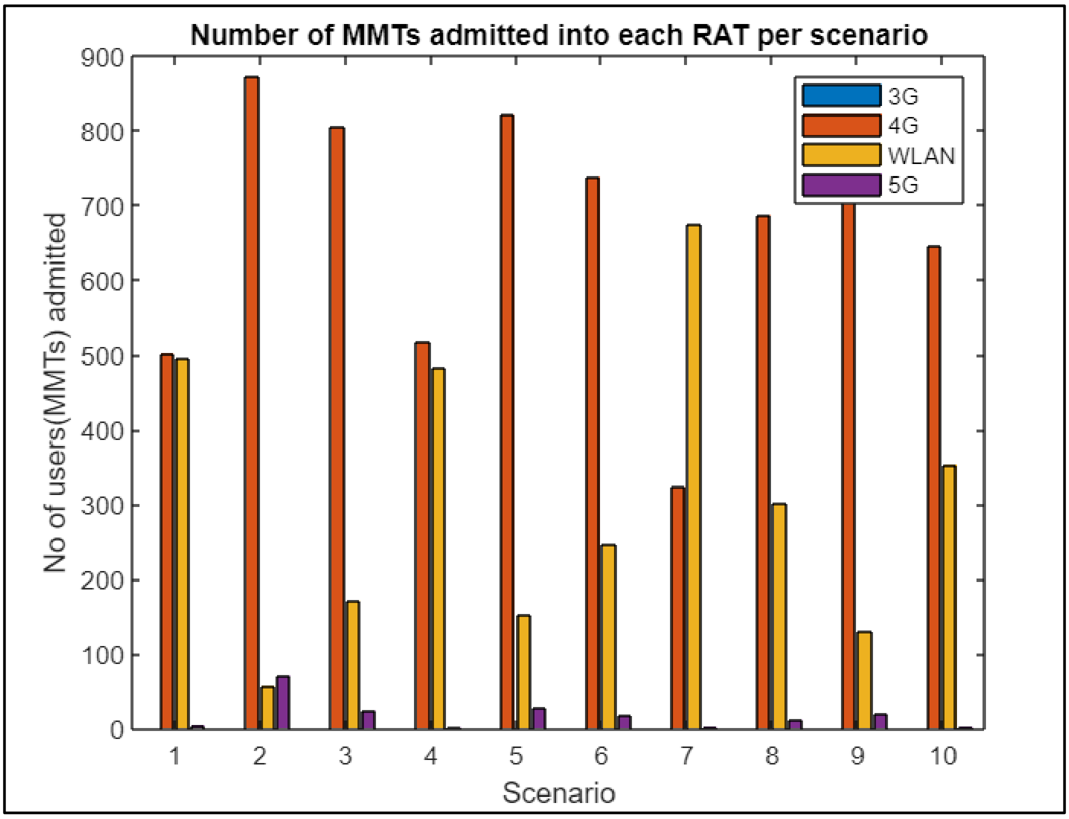


Figure 24: Users admitted in all scenarios for case 7

4.1.3 Experiment 3: Investigating threshold on number of handovers

Using the procedures outlined in section 3.5.4 the number of handovers for different numbers of users was measured and the results were tabulated as shown in table 19 below. The values were average values after running the simulation 3 times for a specific number of users.

Table 19: Mean number of handovers for σ =1.3

Scenario	Number of users	Number of Handovers
1	100	35
2	200	77
3	300	125
4	400	162
5	500	199
6	600	229
7	700	274
8	800	298
9	900	344
10	1000	387

Using table 19 above a graph of the number of handovers against the number of users was plotted using MATLAB as shown in figure 25 below.

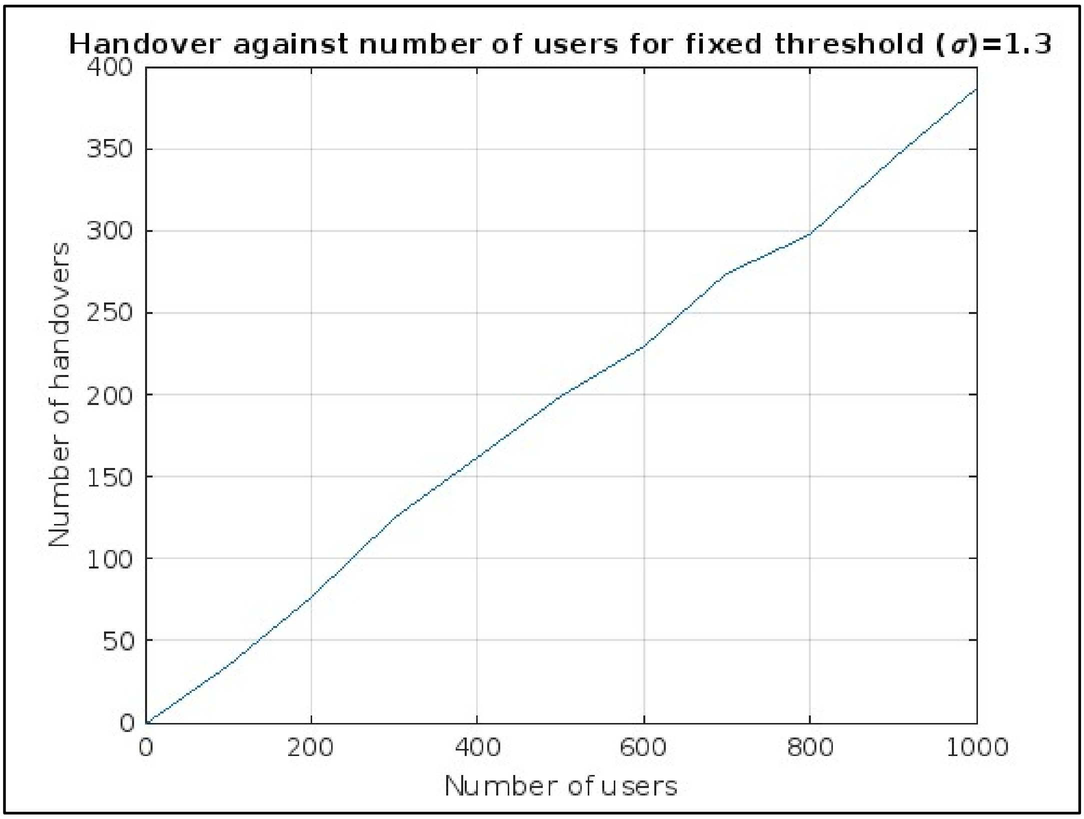


Figure 25: Graph of handovers against number of users

Using the procedures outlined in section 3.5.4 the number of handovers for varying thresholds for 1000 users was measured and the results were tabulated as shown in table 20 below. Note the values were average values after running the simulation 3 times for a specific threshold.

Table 20: Mean number of handovers for 1000 users

Scenario	Threshold	Number of Handovers
1	0.03	1000
2	0.3	1000
3	0.6	1000
4	0.9	1000
5	1.2	587
6	1.5	337
7	1.8	277

8	2.1	275
9	2.4	250
10	2.7	242
11	3.0	137
12	10	44
13	15	0

Using table 20 above a graph of the number of handovers against threshold value was plotted using MATLAB as shown in figure 26 below.

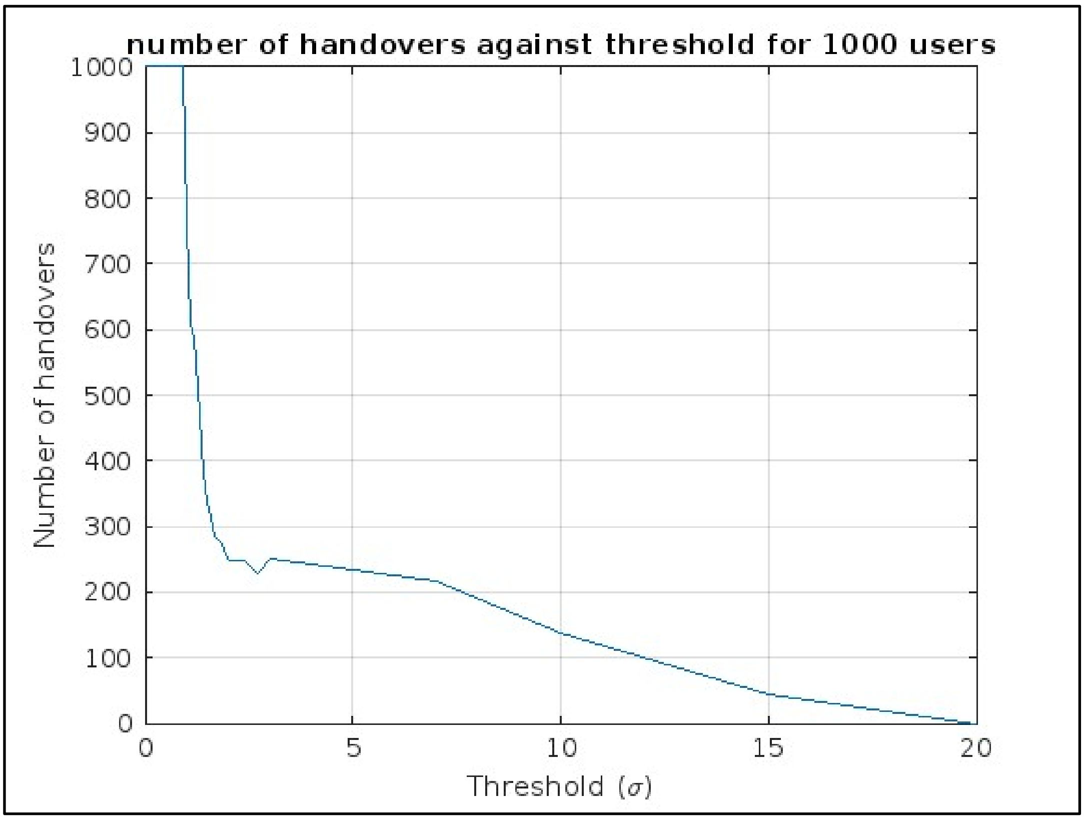


Figure 26: Graph of number of handovers against threshold

Using the procedures outlined in section 3.5.4 the percentage of unnecessary handovers was calculated using (26) and a table of the percentage of unnecessary handovers for each execution of the program was drawn as shown in table 21 below.

Table 21: Percentage of unnecessary handovers

Execution number	% of unnecessary handovers
1	5.7130
2	2.0000
3	9.0909
4	13.9535
5	4.7619
6	7.8947
7	4.5455
8	2.0408
9	2.4390
10	8.3333

A bar chart showing the output of the 1st execution is shown in figure 27 below.

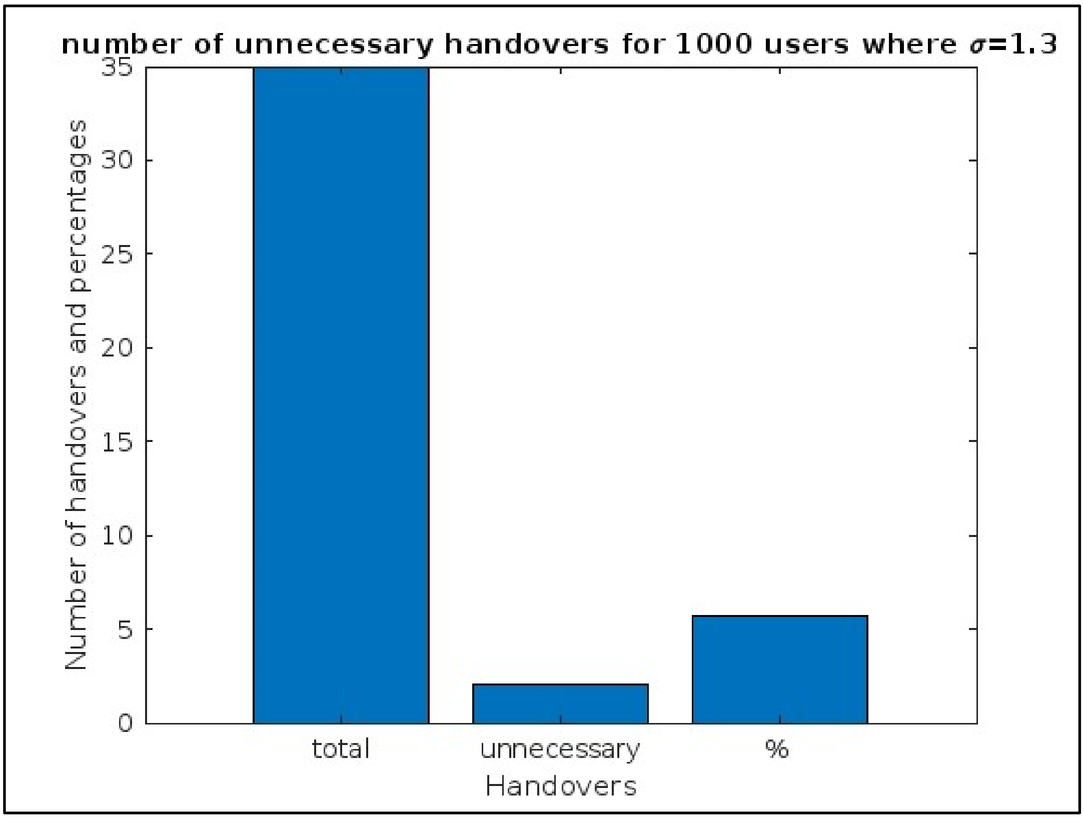


Figure 27: percentage of unnecessary handovers for the 1st execution

Using table 21, a graph of the percentage of unnecessary handovers against execution number was plotted in MATLAB as shown in figure 28 below.

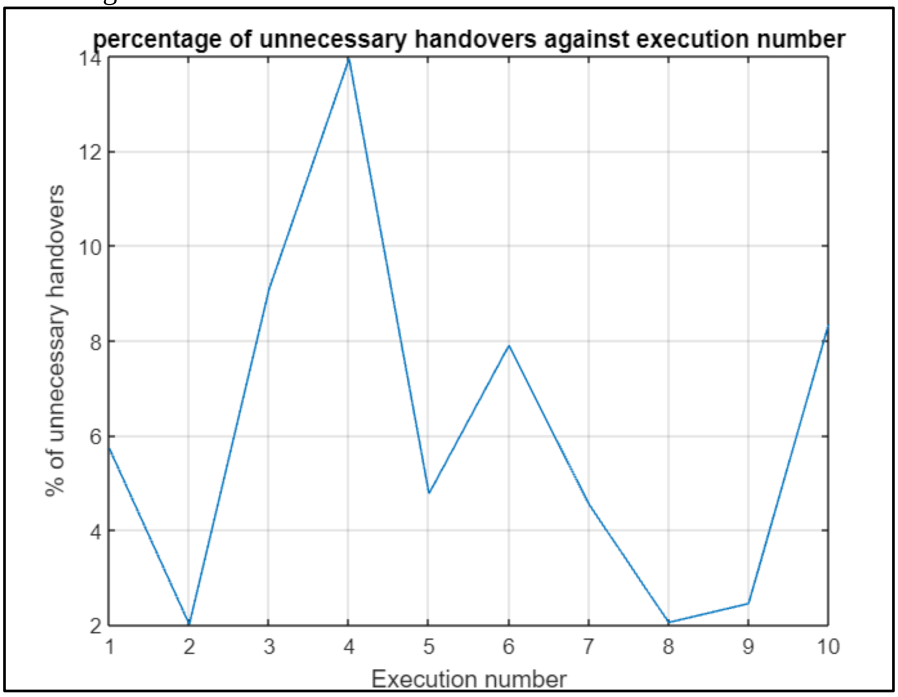


Figure 28: Percentage of unnecessary handovers

5. Discussion

In this section, the results obtained in section 4 are interpreted and analysed to critically evaluate the performance of the designed algorithm. However, due to time constraints, real-time comparisons of the designed algorithm with already existing algorithms in the literature were not done, because the other existing algorithms were not implemented in this report's network model. Therefore, the comparisons are made based on the existing results of other algorithms' performances in the literature.

5.1.1 Experiment 1: Determination of weight proportion parameters

Based on the results in tables 17 and 18, the best combination of weight proportion parameters for the designed algorithm was the same as that in [7]. They varied their network conditions randomly for each combination while the method in this report simulated both static and dynamic network conditions for each combination. The gain calculation in this report was done in the same way as [7] however their calculated gain values differed from the ones in this report. Despite this, combination 3 **(**$\alpha = 0.2,\ \beta = 0.5, \gamma = 0.3$) had the highest gain.

The gain depends on W(comprehensive weight vector of network attributes) and U(comprehensive utility value matrix). Since the objective weights are derived directly from the network conditions (network attribute matrix Mt) the choice of β influences how much the network conditions affect the W value. According to (14) when $\beta = 0.5$ the objective weights contribute the most to W and since they are derived from the network conditions the gain would be higher for larger network attribute values. If β is small, then W would be smaller, and the gain would be limited even for large network attribute values since U is also derived from network conditions.

Then for testing all possible combinations to 1dp the combination that yielded the highest gain was ($\alpha = 0$, $\beta = 0,\ \gamma = 1$). However, this could not be used for the algorithm because the whole purpose of the project was to design a RAT selection algorithm that takes user preferences, service characteristics and network attributes into consideration. The parameters determine how much each of these factors contributes to the algorithm. By making one or more of them 0, it implies that the algorithm no longer takes that factor into consideration. In this case $\alpha = 0,\ \beta = 0$ implies that the algorithm would only consider service characteristics and be too objective leading to a bad user experience as user preferences and network conditions are not considered.

The second-best combination ($\alpha = 0.1,\ \beta = 0.8,\ \gamma = 0.1$) was also not used because the β value was much higher than the other 2 parameters. This would result in the algorithm considering network attributes much more than user preferences and service characteristics even though they have been considered. The contributions of the factors mentioned above to the algorithm would be too uneven making the algorithm lean more towards being too objective. Therefore combination 3 was chosen as the parameter values are close enough to one another to provide a more a less even contribution of the factors to the algorithm.

5.1.2 Experiment 2: Effect of different service priorities on algorithm performance

Based on the results in section 4.1.2 the service priorities of a group of services on an MMT significantly influences the optimal RAT that is determined by the algorithm.

For scenario 1, voice service had the highest priority (5), video streaming had the lowest priority (1) and web browsing had moderate priority (3). As shown by figure 7 and 17, most MMTs were admitted into 4G(LTE) (about 545 users) followed by WLAN (about 453 users) while only 2 users were admitted into 5G, and no users were admitted into 3G (UMTS). Based on the QoS requirements for voice from table 10, voice service requires a very low delay (50-100ms) compared to the other services because significant latency degrades call quality leading to poor user experience. Based on table 2 for our network model, 5G has the lowest delay but few users are admitted to it because it has a very high cost. 3G has the second lowest delay but no users are admitted to it by the algorithm because its bandwidth is too low to meet the QoS requirements for web browsing, since web browsing has moderate priority, the algorithm takes this into account. Therefore, it selects 4G for most users as it has a lower delay than WLAN. Even though both WLAN and 4G have similar bandwidth more users are admitted into 4G than WLAN since voice service doesn't have a high QoS requirement for bandwidth (32-64kbps) and packet loss rate. This shows that the algorithm can select the optimal network for multiple services by taking priority into consideration.

This differs from [7] as their algorithm was able to select UMTS(3G) for scenarios 1 and 2 because it had the lowest delay and jitter. This difference in performance occurs because our algorithm did not include jitter as one of the RAT selection criteria. Therefore, it chose 4G as the optimal RAT for this scenario. Even though 4G has the 3rd lowest delay in our network model, in a real-world scenario this may not necessarily be the optimal RAT for this scenario when 3G is available. Because 4G is the 2nd most expensive RAT as shown by table 2, this may be an issue for users who are sensitive about minimizing their costs. Nonetheless, the above result implies that if our algorithm included more criteria, it would be more accurate in selecting the appropriate RAT in a real-world scenario.

For scenario 2, voice service had the highest priority (5), video streaming had moderate priority (3) and web browsing had the lowest priority (1). As shown by figures 8 and 17, most MMTs were admitted into 4G(LTE) (about 858 users) followed by 5G (about 74 users) while only 68 users were admitted into WLAN, and no users were admitted into 3G (UMTS). As discussed above for scenario 1 since voice has the highest priority the algorithm selects the optimal RAT that satisfies its QoS requirement for low delay. However, since video streaming now has moderate priority, the algorithm takes this into account by having more users admitted into 5G than WLAN and 3G because video streaming requires very high bandwidth (512-5000kbps) as shown in table 10. Therefore, more users are admitted into 5G because it has the highest bandwidth in our network model. However, more users are admitted into 4G than 5G because 4G has a lower cost. Again, this differs from [7] because our algorithm did not include jitter as one of the selection criteria. Since low jitter is an important QoS requirement for both voice and video, if it was included in our algorithm the selection would've been more accurate.

For scenario 3, voice service had the lowest priority (1), video streaming had the highest priority (5) and web browsing had moderate priority (3). As shown by figures 9 and 17, most MMTs were admitted into 4G(LTE) (about 831 users) followed by WLAN (about 157 users) while only 12 users were admitted into 5G, and no users were admitted into 3G (UMTS). Based on table 10 video streaming requires very high bandwidth (512-5000kbps) and moderate delay (74-150ms) therefore the algorithm selects 4G for most users as opposed to WLAN and 5G because it has a lower delay than WLAN and lower cost than 5G, even though it has similar bandwidth and packet loss rates to WLAN. This differs from [7] because their algorithm selected WLAN for most users in this scenario, the reason for the difference in performance is that WLAN had the highest bandwidth in their network and was greater than LTE, whereas our network has similar bandwidths for WLAN and LTE(4G). It also implies that their algorithm strictly selects the optimal RAT for the service that has the highest priority in the group of services whereas our algorithm also takes the priorities of the other 2 services into account when making the selections. For this scenario, the selection of the optimal RAT is more accurate compared to the previous 2 scenarios as the optimal RAT chosen (4G) has the right balance of delay, cost, and bandwidth to meet the QoS requirements of video and web browsing.

For scenario 4, voice service had moderate priority (3), video streaming had the highest priority (5) and web browsing had the lowest priority (1). As shown by figures 10 and 17, almost all MMTs were equally admitted into 4G(LTE) (about 498 users) and WLAN (about 500 users) while only 2 users were admitted into 5G, and no users were admitted into 3G (UMTS). As video streaming has the highest priority and requires high bandwidth and moderate delay as stated in the scenario 3 discussion, the algorithm selects both 4G and WLAN over 5G as the optimal RATs in this scenario. Because they have similar bandwidth, are both cheaper than 5G and meet the QoS bandwidth requirements for both video and voice. However, since video has a greater priority than voice in this scenario there are slightly more users admitted into WLAN than 4G. This is because the delay QoS requirement for video is less strict than voice, so the algorithm selects WLAN which has a lower cost and slightly lower packet loss rate than 4G even though it has greater delay. The result of this scenario is in agreement with [7].

For scenario 5, voice service had the lowest priority (1), video streaming had moderate priority (3) and web browsing had the highest priority (5). As shown by figures 11 and 17, most MMTs were admitted into 4G(LTE) (about 838 users) while WLAN had admitted about 130 users, while only 32 users were admitted into 5G, and no users were admitted into 3G (UMTS). According to table 10, web browsing has the lowest QoS requirements for delay of the 3 services, has a moderate bandwidth requirement (128-1000kbps) and from the literature [8] and [13] has very strict requirements on packet loss rate because the accuracy of the data is important. This result differs from [7] as their algorithm selected WLAN for most MMTs in this scenario. The reason for the difference is that WLAN had a greater bandwidth than LTE in their simulation however their packet loss rates for all RATs were the same as ours. In the case of our network, the selection of 4G as the optimal RAT for this scenario may be a disadvantage in the real world because 4G has the highest packet loss rate, which may be an issue for users prioritizing web browsing even though the delay and bandwidth are acceptable.

For scenario 6, voice service had moderate priority (3), video streaming had the lowest priority (1), and web browsing had the highest priority (5). As shown by figures 12 and 17, most MMTs were admitted into 4G(LTE) (about 736 users) while WLAN had admitted about 252 users, while only 11 users were admitted into 5G, and 1 user was admitted into 3G (UMTS). This was the first scenario where 3G(UMTS) was selected for a user. Again, this differed from [7] because their algorithm included jitter as one of the selection criteria. However, since voice has moderate priority in this scenario our algorithm takes its low delay QoS requirement into account by admitting a user into 3G and prioritizes the low packet loss rate requirement of web browsing by admitting more users into WLAN than scenario 5.

For scenario 7, voice service had the highest priority (5), while video streaming and web browsing had the lowest priority (1). As shown by figures 13 and 17, most MMTs were admitted into WLAN(about 664 users) followed by (4G) LTE which had about 335 users, while only 1 user was admitted into 5G, and 0 users were admitted into 3G (UMTS). For this scenario, it was expected that the algorithm would select the optimal RAT that satisfies the QoS requirements of the highest priority service only and not take the other 2 services' QoS requirements into account because their priority is (1=very low). The results in this scenario differ from scenarios 1 and 2 because they had another service which had a priority greater than 1 influencing the algorithm's selection of the optimal RAT. For this scenario, the algorithm's selection for most users is arguably less accurate than scenarios 1 and 2 because WLAN has a greater delay than 4G and 5G so wouldn't be practical in the real world for users only prioritizing voice.

For scenario 8, voice service and web browsing had the lowest priority (1), while video streaming had the highest priority (5). As shown by figures 14 and 17, most MMTs were admitted into 4G(LTE) (about 729 users) followed by WLAN which had about 262 users, while only 9 users were admitted into 5G, and 0 users were admitted into 3G (UMTS). Similar to the discussion for scenario 7 above it was expected the algorithm would choose the best RAT strictly for video streaming. The results in this scenario are the same as in scenario 3 and 4 where other services had moderate priority. This implies that when video service has the most priority the algorithm accurately selects the optimal RAT for video irrespective of the other service priorities. Since video streaming has a high bandwidth requirement, moderate delay requirement and doesn't care too much about cost [8][13], 4G is a suitable RAT for this scenario because of its high bandwidth and lower delay than WLAN and lower cost than 5G. This would be suitable in a real-world scenario for users sensitive to costs.

For scenario 9, voice service and video streaming had the lowest priority (1), while web browsing had the highest priority (5). As shown by figures 15 and 17, most MMTs were admitted into 4G(LTE) (about 843 users) followed by WLAN which had about 128 users, while only 29 users were admitted into 5G, and 0 users were admitted into 3G (UMTS). The results in this scenario are the same as in scenarios 5 and 6. This implies that when web browsing has the highest priority the algorithm consistently selects the optimal RAT for web browsing irrespective of the other service priorities. Since web browsing has strict requirements on packet loss rate [8][13], 4G may not necessarily be the optimal RAT for this service because it has the highest packet loss rate in the network. However, the algorithm chose this because we used table 10's QoS requirement for packet loss rate, this was assumed to be the same for all services which isn't the case in reality. Therefore, a more accurate definition of each service's QoS requirements would improve the algorithm's accuracy.

For scenario 10, all 3 services had equal priority(5). As shown by figures 16 and 17, most MMTs were admitted into 4G(LTE) (about 647 users) followed by WLAN which had about 351 users, while only 2 users were admitted into 5G, and 0 users were admitted into 3G (UMTS). It was difficult to interpret the result because it wouldn't be possible for all services on the MMT to have the highest priority at the same time. One service would always have to have greater priority than another service to be realistic. Also, it was discovered that the weight calculation would derive the same weights whenever all 3 priorities were equal because the normalized vector would always be $\left[\frac{1}{3},\frac{1}{3},\frac{1}{3}\right]$.

On repeating the experiment for specific user preferences, as shown by figures 18-24, the number of users admitted into each RAT for each case followed a similar pattern to the case of random user-specified weights in figure 17 as the majority of the scenarios had 4G-LTE as the most selected RAT in all the cases. However, some key observations were that WLAN had more users admitted to it than 4G-LTE in scenario 7 (Pu=[5,1,1]) for all cases. Scenario 1 (Pu=[5,1,3]) had similar results for all cases except case 3(delay more important) and 5(voice more important). Scenario 4 (Pu=[3,5,1]) had similar results for all cases except case 3, and Scenario 10 (Pu=[5,5,5]) had similar results for all cases except case 3. Case 2(cost more important) had a significantly larger number of users admitted into 5G for all scenarios compared to the other cases.

The results imply that the algorithm does consider user preferences because the increased number of users admitted into 5G for case 2 showed that when users deemed cost as important to them, the algorithm selected the RAT which had the highest cost more frequently to account for this. The real-world expectation is that the algorithm would admit more users into the cheapest RAT (WLAN) if the cost was important to them. However, the 9-point weight scale used for user-specified weights didn't differentiate between cost and benefit criteria. Therefore, it wasn't clearly defined if low weights for cost implied that the user preferred cheaper costs or expensive costs. The output shows that the algorithm assumed that higher weights for a criterion imply the user prefers higher values of that criterion irrespective of it being a cost or benefit criterion. For example, higher weights for cost imply the user wants higher costs. Therefore, the behaviour of the algorithm is correct but the error in design resulted in the opposite output to what was expected. The same behaviour is observed for delay as all cases select WLAN over 4G-LTE for scenario 7, even though WLAN has a greater delay than 4G-LTE. Moreover, voice had the highest priority and required the lowest delay.

5.1.3 Experiment 3: Investigating threshold on number of handovers

According to figure 25, for a constant threshold of 1.3, as the number of users increased the number of handovers also increased gently. This was expected because we used randomly varying network attributes to simulate mobility (dynamic network conditions) therefore as more users move around the coverage area of the RATs the number of handovers would increase since more users would detect a better RAT than the one they are currently connected to, based on the outcome of equation (26). The results for this simulation differed from [7] as their designed algorithm had a gentler slope implying, they had a slower rate of increase in the number of handovers compared to our algorithm for the same threshold. Their algorithm produced about 250 handovers for 1000 users whereas our algorithm produced 390 handovers for 1000 users. The reason for the differences is that they used $\alpha = \beta = \gamma = \frac{1}{3}, P^u = [5,5,5]$ whereas we used $\alpha = 0.2,\ \beta = 0.5, \gamma = 0.3, P^u = [3,3,3]$. Our pseudocode handed over if the ratio of RAT scores was greater than the threshold whereas their pseudocode took the opposite case of the ratio being less than the threshold. It is also likely that the code implementations in MATLAB were different, but we were unable to compare our code to theirs to verify. However, the use of a threshold in our algorithm has effectively reduced the number of handovers to about 40% for 1000 users which is less than half the total number of users.

According to figure 26, as the threshold increased the number of handovers decreased from 1000 for a threshold of 0 to 0 for a threshold of 20. This was expected because according to equation (26) for very small thresholds (<1) there would always be a handover because the ratio of the best RAT's score to the current RAT's score is always greater than 1. As the threshold becomes greater the number of handovers would decrease as fewer cases of the ratio in (26) are greater than the threshold. Therefore, for very high thresholds there would be no handovers. The authors in [7] chose 1.3 as the threshold determined by experiment. However, they did not provide details of how they derived that value. Therefore, we were unable to determine an optimal threshold for our algorithm.

According to table 21 and figure 28, the lowest percentage of unnecessary handovers was 2% and the highest was 13.95% based on the definition of unnecessary handover in [7]. Unnecessary handovers remained under 20% implying that the threshold effectively minimized the ping-pong effect in our algorithm. For 100 users our algorithm results differed from those in [7] as the designed algorithm produced 10% whereas their algorithm produced a percentage of 3.1%.

6. Conclusions

In this section, conclusions are drawn from the results and discussions in sections 4 and 5 respectively.

In this report, a RAT selection algorithm for multiservice multimode terminals in next-generation wireless networks has been proposed. The algorithm considered 3 network selection factors namely user preferences, service characteristics and QoS requirements, and network attributes to select the optimal RAT for an MMT. The algorithm used entropy, FAHP, TOPSIS and utility functions to determine the optimal RAT for the group of services. Weight proportion parameters were introduced to vary the contribution of each network selection factor to the algorithm and a threshold was introduced to minimize the number of unnecessary handovers.

Several factors influenced the performance of the algorithm including the weight proportion parameters used, the choice and number of criteria, the priority of the group of services, the network attributes themselves and the simulation setup.

The choice of network attribute values (table 2) was collated from several sources and so may have been wrong or not reflective of a network in the real world. As for the simulation setup, there may have been errors in the code implementation of the algorithm due to incorrect interpretations of the equations used in the algorithm. Also, our simulation was very restricted because we only considered triple calls and didn't consider a variable number of services on an MMT or the level of agreement between the services.

Of all the factors listed above, the service priority had the biggest effect on the algorithm's performance because it had to satisfy the QoS requirements of the service that had the highest priority, while also not neglecting the QoS requirements of the lower priority services.

However, the small number of criteria used and the exclusion of important criteria such as jitter affected the accuracy of the algorithm's selections. The cost was assumed to be constant. Also, as discussed in section 5.1.2, The inaccurate definition of the user-specified weight scale affected performance. However as discussed in section 2 and [9], the optimal RAT determined by a group decision isn't necessarily the best for the individual services in the group, because the services have different QoS requirements, and the RATs all have different network attributes suited for different applications. There is a trade-off between meeting individual service QoS requirements when selecting the best RAT for multiple running services on the MMT. Therefore, we can argue that the algorithm selected the best RAT (4G-LTE) for the group of services on the MMT given the network conditions and QoS requirements of the services provided in the model. The selections would likely be more reflective of what happens in reality if a more accurate model was used in the network model design.

Despite the design flaws, we can conclude that the proposed algorithm was able to solve the problem of vertical handover decisions for multiple services in next-generation heterogeneous wireless networks thus satisfying the project requirements.

7. Recommendations

The following recommendations for future work were made based on the conclusions in section 6.

7.1.1 Improve network model design

Firstly, include more RAT selection criteria in the algorithm. Our investigation used 4 criteria whereas 5 is the most common number of criteria used in RAT selection algorithms across the literature. Increasing the number of criteria results in a more informed decision.

Secondly, include the most appropriate criteria for the services considered in the algorithm. For example, for an algorithm that selects the best RAT for voice, video streaming and web browsing, jitter is an essential criterion that must be included and can't be omitted. Also, the user-specified weight scale should be properly defined in terms of the meanings of the weights assigned to benefit criteria and cost criteria.

Lastly, the performance ratings/network attribute values chosen for the RATs used in the model should be accurate values that reflect real-world network attributes for such RATs.

7.1.2 Use a broader project scope

Our investigation used a very simple model that had many assumptions, to make the model more realistic in future, the scope of the project should be broadened to include the possibility of varying numbers of services on the MMT. E.g., single service, double service and or quadruple service RAT selection in addition to the triple service RAT selection that was implemented. In addition, novel services used in 5G such as Virtual Reality and cloud gaming should be considered in future models.

The algorithm should also be extended to include independent call decisions in addition to group call decisions. Such as handing over one or more calls from the group to one or more RATs simultaneously as opposed to one RAT at a time. A mechanism to determine the level of agreement(consensus mechanism) between decision-makers should also be introduced to achieve this.

In future, load balancing should be included in the simulation by limiting the network resources in the model. We assumed there were unlimited resources in all RATs so there was no limit to the number of users they could admit/handover. Also, the RATs that the calls are handed over should be tracked to give more insight into the optimal RAT for handover as we only determined the optimal RAT for selection.

7.1.3 Determine optimal parameters using machine learning

The threshold and weight proportion parameters were adapted from [7] and may not have necessarily been the optimal parameters for our network model. Therefore, in future machine learning techniques should be used to determine the optimal parameters and thresholds to use for the simulations.

8. List of References

[1] "Forecast number of mobile users worldwide 2020-2025 | Statista", statista.com, (Accessed 17 Aug. 2023)

[2] H. T. Yew, E. Supriyanto, M. H. Satria, and Y. W. Hau, “New vertical handover method to optimize utilization of wireless local area network in High-Speed Environment,” *PLOS ONE*, vol. 11, no. 11, p. e0165888, Nov. 2016, doi: 10.1371/journal.pone.0165888.

[3] E. Triantaphyllou, “Multi-Criteria Decision-Making Methods,” in Multi-criteria Decision-Making Methods: A Comparative Study, vol. 44, Applied Optimization, Boston, MA: Springer, 2000, pp. 5-21, doi: 10.1007/978-1-4757-3157-6_2

[4] E. Obayiuwana and O. E. Falowo, “Network selection in heterogeneous wireless networks using multi-criteria decision-making algorithms: a review,” *Wireless Networks*, vol. 23, no. 8, pp. 2617–2649, Jun. 2016, doi: 10.1007/s11276-016-1301-4.

[5] F. Anwar, M. H. Masud, B. U. Islam, and R. F. Olanrewaju, “Radio Access Technology (RAT) Selection Mechanism using TOPSIS Method in Heterogeneous Wireless Networks (HWN),” *Indonesian Journal of Electrical Engineering and Computer Science*, Nov. 2018, doi: 10.11591/ijeecs.v12.i2.pp852-864.

[6] J. D. Martínez-Morales, U. Pineda-Rico and E. Stevens-Navarro, “Performance comparison between MADM algorithms for vertical handoff in 4G networks,” 2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control, Tuxtla Gutierrez, Mexico, 2010, pp. 309-314, doi: 10.1109/ICEEE.2010.5608646

[7] H. Yu, Y. Ma, and J. Yu, “Network Selection Algorithm for Multiservice Multimode Terminals in Heterogeneous Wireless Networks,” *IEEE Access*, vol. 7, pp. 46240–46260, 2019, doi: https://doi.org/10.1109/access.2019.2908764.

[8] O. E. Falowo and H. A. Chan, “Dynamic RAT selection for multiple calls in heterogeneous wireless networks using group decision-making technique,” *Computer Networks*, vol. 56, no. 4, pp. 1390–1401, Mar. 2012, doi: 10.1016/j.comnet.2011.12.013.

[9] O. E. Falowo and O. A. Taiwo, “Network selection decisions for multiple calls based on consensus level,” *Journal of Network and Systems Management*, vol. 26, pp. 592-615, 2018, doi: 10.1007/s10922-017-9433-0

[10] O. E. Falowo and H. A. Chan, “Multiple-call handover decisions using fuzzy MCGDM in heterogeneous wireless networks,” in *2011-MILCOM 2011 Military Communications Conference*, pp. 1909-1914, IEEE, Nov. 2011

[11] O. A. Taiwo and O. E. Falowo , “Comparative analysis of algorithms for making multiple-sessions handover decisions in next generation wireless networks,” in *2013 Africon*, pp. 1-6, IEEE, Sep. 2013, doi: 10.1109/afrcon.2013.6757805

[12] U. Paul and O. E. Falowo, “Efficient RAT-selection for group calls using intuitionistic fuzzy TOPSIS in heterogeneous wireless networks,” in *2017 IEEE AFRICON*, pp. 365-370, IEEE, Sep. 2017, doi: 10.1109/afrcon.2017.8095510

[13] R. Luo, S. Zhao, and Q. Zhu, “Network selection algorithm based on group decision making for heterogeneous wireless networks,” in *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pp. 397-402, IEEE, May 2017, doi: 10.1109/iccsn.2017.8230143

[14] E. Obayiuwana and O. Falowo, “A new network selection algorithm for group calls over heterogeneous wireless networks with dynamic multi-criteria,” in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 491-494, IEEE, Jan. 2016, doi: 10.1109/ccnc.2016.7444829

[15] A. Sgora, N. Bouropoulou, M. Stamatelatos and A. Konidaris, "A Network Selection Algorithm for 5G Heterogeneous Environments," *2022 IEEE Conference on Standards for Communications and Networking (CSCN)*, Thessaloniki, Greece, 2022, pp. 48-52, doi: 10.1109/CSCN57023.2022.10051092.

[16] A. K. Yadav, H. Alam, V. K. Singh and A. Kumar, "Performance Analysis of Network Selection Technique for Handover Strategies in Heterogeneous 5G Network," *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, Sonbhadra, India, 2022, pp. 818-824, doi: 10.1109/AIC55036.2022.9848989.

[17] A. Zhu, M. Ma, S. Guo, S. Yu, and Y. Lin, "Adaptive Multi-Access Algorithm for Multi-Service Edge Users in 5G Ultra-Dense Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2807–2821, Mar. 2021, doi: https://doi.org/10.1109/tvt.2021.3060573.

[18] S. S, J. Augustine and S. K. Ghosh, "An analytic hierarchy process (AHP) based UAV selection mechanism for beyond 5G networks," *2022 IEEE 21st International Symposium on Network Computing and Applications (NCA)*, Boston, MA, USA, 2022, pp. 27-34, doi: 10.1109/NCA57778.2022.10013547.

[19] S. K. Borah and S. Bitragunta, "An Intelligent Link Selection Mechanism for Hybrid Classical-Quantum Communication Systems," in *IEEE Communications Letters*, vol. 26, no. 2, pp. 301-305, Feb. 2022, doi: 10.1109/LCOMM.2021.3129792.

[20] C. C. González, E. F. Pupo, L. Atzori and M. Murroni, "Dynamic Radio Access Selection and Slice Allocation for Differentiated Traffic Management on Future Mobile Networks," in *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 1965-1981, Sept. 2022, doi: 10.1109/TNSM.2022.3150978.

[21] C. -Q. Dai, J. Xu, J. Wu and Q. Chen, "Multi-objective Intelligent Handover in Satellite-Terrestrial Integrated Networks," *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, Seoul, Korea, Republic of, 2022, pp. 367-372, doi: 10.1109/ICCWorkshops53468.2022.9814653.

[22] X. Guo, M. H. Omar, K. M. Zaini, G. Liang, M. Lin and Z. Gan, "Multiattribute Access Selection Algorithm for Heterogeneous Wireless Networks Based on Fuzzy Network Attribute Values," in *IEEE Access*, vol. 10, pp. 74071-74081, 2022, doi: 10.1109/ACCESS.2022.3186149

[23] Z. Xu, Z. Zhang, S. Wang, Y. Yan and Q. Cheng, "Resource-Constraint Network Selection for IoT Under the Unknown and Dynamic Heterogeneous Wireless Environment," in *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12322-12337, 15 July15, 2023, doi: 10.1109/JIOT.2023.3245198

[24] **"Signal Transmission,"** [Online]. Available: https://freesvg.org/signal-transmission. Accessed: Sep. 25, 2023

[25] Y. Zhong, H. Wang, and H. Lv, "A cognitive wireless networks access selection algorithm based on MADM," *Ad Hoc Networks*, vol. 109, p. 102286, 2020, doi: https://doi.org/10.1016/j.adhoc.2020.102286

[26] R. Honarvar, A. Zolghadrasli, and M. Monemi, "Context-oriented performance evaluation of network selection algorithms in 5G heterogeneous networks," *Journal of Network and Computer Applications*, vol. 202, p. 103358, 2022

[27] H. W. Yu and B. Zhang, "A heterogeneous network selection algorithm based on network attribute and user preference," in *Ad Hoc Networks*, vol. 72, pp. 68-80, 2018

[28] G. Liang and H. Yu, "Network selection algorithm for heterogeneous wireless networks based on service characteristics and user preferences," in *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1-16, 2018

9. Appendices

Each appendix below consists of the MATLAB code used to simulate each experiment and portions of the experiments. All the MATLAB programs made use of the same functions defined in Appendix A. All MATLAB files are also available here.

9.1.1 Appendix A: Experiment 1 implementation

```
%Experiment 1 determining alpha,beta and gamma
%Author: Bina E Mukuyamba
%Date: 17/10/2023

%Updated network attributes for BW

%result: so far combination 1 gave highest mean gain
% alpha=0.2,beta=0.3,gamma=0.5

%NB the terms vectors and arrays will be used interchangably
%   treat the matrices as vectors of vectors

%1st experiment:network attributes were kept constant only user specified weights
were
%varied for each user so only WU will be different
%repeat of experiment used varied network attributes, therefore only Ws
%stayed constant, everything else changed

%comment out when doing the dynamic experiment
%Defining the RATs
%Default values from Table 3
%RAT-N = [bandwidth,cost,delay,loss rate] are our criteria
%3G
%RAT_1 =[1.350,3.5,30,6];
%%4G
%RAT_2=[2.400,4.5,60,13];
%%WLAN
%RAT_3=[4.500,0.5,80,10];
%%5G
%RAT_4=[1000,7,13,4];

%comment out when doing the static experiment
%Defining the RATs
%Dynamically changing values
%will be randomly selected from table 2 each time program is run
%NB cost is only criterion that will remain the same
%RAT-N = [bandwidth,cost,delay,loss rate]
%3G
RAT_1 =[(randi([7,20]))/10,3.5,randi([10,50]),randi([2,10])]; %divide by 10 to
generate random decimals
%4G
RAT_2=[(randi([8,1000]))/10,4.5,randi([40,80]),randi([6,20])];
%WLAN
RAT_3=[randi([1,100]),0.5,randi([70,90]),randi([4,15])];
%5G
RAT_4=[randi([100,1000]),7,randi([1,25]),(randi([1,80]))/10];

%MODULE 1: WEIGHTING
%Attribute matrix= Mt
%row_i of matrix = RAT_i attribute values
```

```
%Rown=RATn, colx=criteriax_value
Mt=[RAT_1;RAT_2;RAT_3;RAT_4];
%normalized net attribute matrix Mtbar normalizing using (2)
Mt_norm=zeros(4,4);
for j=1:4
    for i=1:4
        Mt_norm(i,j)=Mt(i,j)/sum(Mt(:,j));
    end
end
Mt_norm;

%ENTROPY calculation using (3)
Entropy_values=zeros(1,4);
k=-1/log(4); %constant
vecsum=0; %initializing
for j=1:4
    Mtvec=Mt_norm(:,j); %column vector
    for i=1:4
        vecsum=vecsum+sum(Mtvec(i)*log(Mtvec(i))); %intermediate sums
    end
    Entropy_values(j)=k*vecsum; %entropy value
    vecsum=0; %don't forget to reset the running total
end
Entropy_values;
%Objective weights calculation using (4)
ObjectiveWeights=zeros(1,4);

for j=1:4
    ObjectiveWeights(j)=(1-Entropy_values(j))/(4-sum(Entropy_values));
end
ObjectiveWeights;
%Subjective user preference weights calculation
%can vary the number of users to simulate for
%assume 10 pt weight scale [0,9] user specified weights are randomized
%create the users
numUsers=input("enter number of users to simulate"); %used 1000 for the report
users=allocateWeights(numUsers); % numUsers MMTs are created with random weights
for each service

%uncomment to check that each user has 3 sets of weights
%Assume 1st vector=voice, 2nd vector=voice, 3rd vector=web
%for i=1:numUsers
%disp(users{i})
%end

%usersthe_weights=users{1} %returns user 1's weights for all services
%voicethe_weights=users{1}{1} %returns user 1's weights for voice uncomment to
%bw_weight_for_voice=users{1}{1}(1) %returns user 1's weight for BW for voice

Pu=[3,3,3]; %user specified service priority vector, this will be the same for all
MMTs
Ps=Pu; %service determined priority=user specified priority

%normalizing weights and priority
NormedUweights=Normalize(users);

%Display normalized weights for each service specified by user
%for i=1:numUsers  %uncomment to check for correctness
%disp(NormedUweights{i})
```

```
%end

sp=sum(Pu);
Pu_norm=zeros(1,3);%normalized vector
for i=1:3
Pu_norm(i)=Pu(i)/sp;
end
Ps_norm=Pu_norm;%normalized priority vectors

%Wu user specified weight vector calculation
%each user will have unique Wu due to having unique weights per service
Wu_allUsers=calculate_Wu(NormedUweights,Pu_norm); %stores Wu for each user in a
cell

%display their Wu
%for i=1:numUsers
%disp(Wu_allUsers{i})
%end
%Wu_allUsers{1} %returns the aggregated user weights for user 1

%run FAHP_EAM.m with Pu=[3,3,3] then sub in result
Ws=[0.3300,0.3240,0.2097,0.1363]

%Comprehensive weight vector calculation
%Combines Wo(objective weights),Ws and Wu using weight proportion parameters
%alpha=a, beta=b, gamma=y
%W(comprehensive weights)=aWu+bWo+yWs
%commented out (not needed for this experiment)
%Wvec=cell(1,numUsers); %vector to store comprehensive weights for each user
%a=0.2;b=0.5;y=0.3; %varied by the experiment
%ObjectiveWeights;
%Ws;
%for i=1:numUsers
%    W=(a*Wu_allUsers{i})+(b*ObjectiveWeights)+(y*Ws); %using formula for each
user i
%    Wvec{i}=W;
%end
%%Wvec{1} %comprehensive weights W for user 1
%for i=1:numUsers  %uncomment to check for correctness
%disp(Wvec{i})
%end

%MODULE 2: UTILITY
Mt;
Mt(:,1)=Mt(:,1)*1000; %in kbps for utility

%utility is calculated by substituing the performance rating from Mt into
%the appropriate utility function and constants

u1=zeros(4,4); %utility value matrices for each service (voice)
u2=zeros(4,4); %video
u3=zeros(4,4); %web browsing
%voice utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u1(i,j)=f_x("voice",colvec(j));
```

```
    end
    if i==2
    u1(i,j)=h_x2("voice",colvec(j));
    end
    if i==3
        u1(i,j)=g_x("voice",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u1(i,j)=h_x("voice",colvec(j));
    end

    end

end
u1=transpose(u1); %get the elements the right way round
%video utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u2(i,j)=f_x("video",colvec(j));
    end
    if i==2
    u2(i,j)=h_x2("video",colvec(j));
    end
    if i==3
        u2(i,j)=g_x("video",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u2(i,j)=h_x("video",colvec(j));
    end

    end

end
u2=transpose(u2);
%web browsing utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u3(i,j)=f_x("web",colvec(j));
    end
    if i==2 %cost not security function
    u3(i,j)=h_x2("web",colvec(j));
    end
    if i==3
        u3(i,j)=g_x("web",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u3(i,j)=h_x("web",colvec(j));
    end

    end

end
u3=transpose(u3) ;
```

```
%Comprehensive utility value matrix calculation
U=zeros(4,4);
uvec={u1,u2,u3}; %using Matlab matrices
%disp(uvec{1})

%3 way loop i,j and g vary
tally=0;
for i=1:4
    for j=1:4
        for g=1:3
            usomething=uvec{g}; %u_ij for a specific service g
            tally=tally+usomething(i,j)*Ps_norm(g); %break down the sum into parts
        end
         U(i,j)=tally;
         tally=0;

    end
end
U ;%final comprehensive utility value matrix

%EXPERIMENT 1: Varying weight proportion parameters
%tested the same combination of parameters as table 4 in [7]
%function used find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
%Experiment was done twice using default network conditions/attribute
%values
%2nd time randomly varied network Attribute values (Mt) for each iteration
%user allocated weights were random and priority was constant
%gains-i corresponds to the calculated gains for combination i
%i=1...7
gains1=zeros(1,numUsers); %vectors to store gains for all users for a particular
parameter combination
gains2=zeros(1,numUsers); %gains for combination 2 from table 4 in [7]
gains3=zeros(1,numUsers);
gains4=zeros(1,numUsers);
gains5=zeros(1,numUsers);
gains6=zeros(1,numUsers);
gains7=zeros(1,numUsers); %and so on up to combination 7

for i=1:numUsers
    %there will be numUsers*7 gains in total, each user will have 7 gain
    %values, 1 for each combination
    WU=Wu_allUsers{i}; %user weight preference vector is unique for each user
    gains1(i)=find_gain(U,0.2,0.3,0.5,WU,Ws,ObjectiveWeights); %gain for ith user
for combination i

    gains2(i)=find_gain(U,0.3,0.2,0.5,WU,Ws,ObjectiveWeights);

    gains3(i)=find_gain(U,0.2,0.5,0.3,WU,Ws,ObjectiveWeights);

    gains4(i)=find_gain(U,0.3,0.5,0.2,WU,Ws,ObjectiveWeights);

    gains5(i)=find_gain(U,0.5,0.2,0.3,WU,Ws,ObjectiveWeights);

    gains6(i)=find_gain(U,0.5,0.3,0.2,WU,Ws,ObjectiveWeights);

    gains7(i)=find_gain(U,(1/3),(1/3),(1/3),WU,Ws,ObjectiveWeights);

end
%store the mean(average) gain of each combiantion into an array
```

```
%each position in the array corresponds to a particular scenario
%the combination that has the highest average gain is the one that will be
%used to calculate the comprehensive weights W in the later experiments
mean_gains=[mean(gains1),mean(gains2),mean(gains3),mean(gains4),mean(gains5),mean(
gains6),mean(gains7)]
highest_gain=0;
for g=1:length(mean_gains)
    if(mean_gains(g)>highest_gain)
        highest_gain=mean_gains(g);
        bestParamcombo=g;
    end
end
bestParamcombo
%The following are the utility functions for each
%criterion as defined in Table 3 in [7]
%each returns a single utility value u
%for a particular service and performance rating x
function u = f_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.25;b=48;
    u=1/(1.0+e^(-a*(x-b)));
    case "video"
    a=0.003;b=2000;
    u=1/(1.0+e^(-a*(x-b)));
    otherwise %web browsing
    a=0.01;b=564;
    u=1/(1.0+e^(-a*(x-b)));

end

end
function u = g_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.1;b=75;
    u=1-(1/(1.0+e^(-a*(x-b))));
    case "video"
    a=0.1;b=112.5;
    u=1-(1/(1.0+e^(-a*(x-b))));
    otherwise %web browsing
    a=0.03;b=375;
    u=1-(1/(1.0+e^(-a*(x-b))));

end

end
function u = h_x(service,x)
switch service
    case "voice"
    g=1/30.0;
    u=1-(g*x);
    case "video"
    g=1/30.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/30.0;
```

```
    u=1-(g*x);

end

end
%updated
function u = h_x2(service,x)
switch service
    case "voice"
    g=1/50.0;
    u=1-(g*x);
    case "video"
    g=1/50.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/50.0;
    u=1-(g*x);

end

end

%function to assign weights
%Function returns a 1x3 cell which consists of vectors of user defined weights
%for each service
% each element of the cell corresponds to a different service
%assume w{1}=voice weights for user i
function w = allocateWeights(N) %N = number of users you want

w=cell(1,N); %creates 1xN cell where each row = 1, column= N 1x3 sets of user
weights for user i
the_weights=[1,2,3,4,5,6,7,8,9];
for i =1:length(w) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=randi([weights(1),weights(9)]);
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
    end
    w{i}=users_i_Weights; %each user-i will have 3 sets of randomized weights
end

end

%returns normalized weights for a cell of user weight vectors
%returns a 1x3 cell consisting of vectors of normalized user weights for each
%service for each user
function Norm = Normalize(userCell)
userCellSize=size(userCell);
Norm=cell(1,userCellSize(2)); %should be the same size as input cell
for i =1:length(Norm) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=userCell{i}{j}(k)/sum(userCell{i}{j});%normalize using equation
```

```
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
     end
     Norm{i}=users_i_Weights; %each user-i will have 3 sets of normalized weights
 end

 end

 %Function that calculates Wu for N users
 %inputs normalized weights (cell) for all users, normalized user specified
priority vector
 %returns a  1xN cell of Wu vectors for all users
 function WuVec = calculate_Wu(normalizedWeights,Puvec)
 normSize=size(normalizedWeights);
 WuVec=cell(1,normSize(2)); %store Wu vector for all users
 for i=1:normSize(2) %find weight user weight vector for each user i
 %wunormvec=the weight vectors for each service for user i placed in one
 %vector for easier calculation
 wunormvec=[normalizedWeights{i}{1};normalizedWeights{i}{2};normalizedWeights{i}{3}
]; %each row is the normalized weight vector for each service
 Wu=zeros(1,4); %Wu user specified weight vector for group of services
 total=0;
 for j=1:4
     colvec=wunormvec(:,j); %break down the rieman sum into steps
     for g=1:3 %loop through services
         total=total+(Puvec(g)*colvec(g)); %sum of Pu_norm * uij^g
     end
     Wu(j)=total;
     total=0;
 end
 WuVec{i}=Wu; %assign calculated Wu vector to user i
 end
 end

 %this function uses TOPSIS to calculate the scores of each RAT
 %for all users
 %returns a 1XN cell of vectors of RAT scores for each user
 function all_RAT_scores_per_user = TOPSIS(Decision_matrices)
 num=size(Decision_matrices);
 all_RAT_scores_per_user=cell(1,num(2));
 for u = 1:num(2) %calculate score for all u users
     D=Decision_matrices{u}; %each user has unique D
     %TOPSIS code
     Dplus=zeros(1,4);
     Dminus=zeros(1,4);
     %store the ideal solutions D+ and D-
     for i=1:4
     colvec=D(:,i);
         if i<=2
         Dplus(i)=max(colvec);
         Dminus(i)=min(colvec);
         end
         if i>2
         Dplus(i)=min(colvec);
         Dminus(i)=max(colvec);
         end

     end
```

```
    %calculate euclidean distances Si+ and Si-
    %for each RAT to ideal solution
    Siplus=zeros(1,4);
    Siminus=zeros(1,4);%vectors to store each Si value
    tally1=0;
    tally2=0;
    for i=1:4
        for j=1:4
        tally1=tally1+(Dplus(j)-D(i,j))^2; %sum of squared diff
        tally2=tally2+(D(i,j)*Dminus(j))^2; %sum of products (22)
        end
        Siplus(i)=sqrt(tally1);
        Siminus(i)=sqrt(tally2);
        tally1=0;
        tally2=0;
    end
    SC=zeros(1,4); %array to store each RATs score
    for i=1:4
        SC(i)=Siminus(i)/(Siminus(i)+Siplus(i));
    end
    all_RAT_scores_per_user{u}=SC; %assign array of RAT scores for user i
end
end

%This function calculates the best RAT for N users
%input:a cell consisting of the vectors of each users RAT scores
%output: 1xN cell of the best RAT for each user
function best_RAT_per_user = UserRats(RATscores)
num=size(RATscores);
best_RAT_per_user=cell(1,num(2));
for k=1:num(2) %find best RAT for all users
    Best_RAT=0;
    UserRatScores=RATscores{k};
    for i=1:4
    score=UserRatScores(i);
    if score>=max(UserRatScores)
    Best_RAT=i;
    end
    end
    best_RAT_per_user{k}=Best_RAT;
end
end

%calculates the gain defined by (30) in [7]
%returns a single value (gain)
%inputs: U, a,b,y and Wu,Ws and Wo vectors
function T = find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
%a,b,y are varied to determine the best gain for our model
%using (30) from [7]
Wvec=(alpha*WU)+(beta*WO)+(gamma*WS);
gain=zeros(1,4);
total=0;
for i=1:4
    for j=1:4
        total=total+(Umat(i,j)*Wvec(j));
    end
    gain(i)=total;
    total=0;
end
```

```
T=sum(gain);

end
```

9.1.2 Appendix B: FAHP implementation

```
%Program to workout FAHP by extent analysis method
%Author Bina Mukuyamba
%Date:22/10/2023

%vary priority as needed
%Copied FCM from tables 6-9 in [7]
%Note order is different from tables in [7] due to order of criteria
%equal =(1,1,1) rather than (1,1,3)
%matrix=[BW,C,D,PLR]
%used cells for easier computation

%Fuzzy comparison matrices for each service
FCMvoice={[1,1,1],[1/8,1/6,1/4],[1/4,1/2,1],[1,2,4];[4,6,8],[1,1,1],[3,5,7],[5,7,9
];[1,2,4],[1/7,1/5,1/3],[1,1,1],[1,3,5];
    [1/4,1/2,1],[1/9,1/7,1/5],[1/5,1/3,1],[1,1,1];}
FCMvideo={[1,1,1],[4,6,8],[1,2,4],[4,6,8];[1/8,1/6,1/4],[1,1,1],[1/5,1/3,1],[1,1,1
];

[1/4,1/2,1],[1,3,5],[1,1,1],[1,3,5];[1/8,1/6,1/4],[1,1,1],[1/5,1/3,1],[1,1,1];}
FCMwebBrowsing={[1,1,1],[1,2,4],[3,5,7],[1/4,1/2,1];[1/4,1/2,1],[1,1,1],[2,4,6],[1
/5,1/3,1];

[1/7,1/5,1/3],[1/6,1/4,1/2],[1,1,1],[1/8,1/6,1/4];[1,2,4],[1,3,5],[4,6,8],[1,1,1];
}
N=4; %4 RATs
wSvoice=zeros(1,4);
wSvideo=zeros(1,4);
wSweb=zeros(1,4); %vectors to store calculated weights of criteria per service
Pu=[2,2,2] %user specified service priority vector, this will be the same for all
MMTs
Ps=Pu;
sp=sum(Pu);
Pu_norm=zeros(1,3);%normalized vector
for i=1:3
Pu_norm(i)=Pu(i)/sp;
end
Ps_norm=Pu_norm;

%computation for voice weights
rowsum=cell(1,4);
for i=1:4
    ai=FCMvoice(i,:); %work out sum of TFN for each row

rowsum{i}=[ai{1}(1)+ai{2}(1)+ai{3}(1)+ai{4}(1),ai{1}(2)+ai{2}(2)+ai{3}(2)+ai{4}(2)
,ai{1}(3)+ai{2}(3)+ai{3}(3)+ai{4}(3)]
end
rowsum; %stores sum of TFN(1x3 vector) for all 4 rows/criteria
colsum = zeros(1, 3);
% Iterate through the cell array and sum the elements
for i = 1:numel(rowsum)
   currentVector = rowsum{i}
```

```
    colsum = colsum + currentVector
end
colsum;
colsum=flip(colsum); %calculate the inverse by rearranging order of TFN
inverseTFN=[1/colsum(1),1/colsum(2),1/colsum(3)]; %take reciprocal to get inverse
%calculate Si values
Si_values=cell(1,4); %store Si values
for k=1:4
    Si_values{k}=rowsum{k}.*inverseTFN; %si=row i sum * inverse TFN
end
Si_values
S1=Si_values{1};
S2=Si_values{2};
S3=Si_values{3};
S4=Si_values{4};
%Calculate degree of possibility
%deviated from equation (10)
degreesOfpossibility=cell(4,3);
%S1 comparisions
%M2greaterM1(M1,M2) = solves for M2>M1
degreesOfpossibility{1,1}=M2greaterM1(S2,S1); % Solves S1>S2
degreesOfpossibility{1,2}=M2greaterM1(S3,S1); % Solves S1>S3
degreesOfpossibility{1,3}=M2greaterM1(S4,S1); %Solves S1>S4
%S2 comparisions
degreesOfpossibility{2,1}=M2greaterM1(S1,S2); % Solves S2>S1
degreesOfpossibility{2,2}=M2greaterM1(S3,S2); % Solves S2>S3
degreesOfpossibility{2,3}=M2greaterM1(S4,S2); % Solves S2>S4
%S3 comparisions
degreesOfpossibility{3,1}=M2greaterM1(S1,S3); % Solves S3>S1
degreesOfpossibility{3,2}=M2greaterM1(S2,S3); % Solves S3>S2
degreesOfpossibility{3,3}=M2greaterM1(S4,S3); % Solves S3>S4
%S4 comparisions
degreesOfpossibility{4,1}=M2greaterM1(S1,S4); % Solves S4>S1
degreesOfpossibility{4,2}=M2greaterM1(S2,S4); % Solves S4>S2
degreesOfpossibility{4,3}=M2greaterM1(S3,S4); % Solves S4>S3

degreesOfpossibility
%calculate degree of possibility for convex Si being greater than k TFNs
%find the minimum from each group of comparisons
minVs=zeros(1,4);
for o=1:4
    row=cell2mat(degreesOfpossibility(o,:));
    minVs(o)=min(row);
end
minVs
%Normalize the values in MinVs to obatin weight vector
tot=sum(minVs);
for x=1:4
    wSvoice(x)=minVs(x)/tot;
end
wSvoice
%Repeat the same process for video
%computation for video weights
rowsum=cell(1,4);
for i=1:4
    ai=FCMvideo(i,:); %work out sum of TFN for each row

rowsum{i}=[ai{1}(1)+ai{2}(1)+ai{3}(1)+ai{4}(1),ai{1}(2)+ai{2}(2)+ai{3}(2)+ai{4}(2)
,ai{1}(3)+ai{2}(3)+ai{3}(3)+ai{4}(3)];
```

```
end
rowsum; %stores sum of TFN(1x3 vector) for all 4 rows/criteria
colsum = zeros(1, 3);
%% Iterate through the cell array and sum the elements
for i = 1:numel(rowsum)
   currentVector = rowsum{i};
   colsum = colsum + currentVector;
end
colsum;
colsum=flip(colsum);
inverseTFN=[1/colsum(1),1/colsum(2),1/colsum(3)];
%calculate Si values
Si_values=cell(1,4);
for k=1:4
    Si_values{k}=rowsum{k}.*inverseTFN;
end
Si_values;
S1=Si_values{1};
S2=Si_values{2};
S3=Si_values{3};
S4=Si_values{4};
%Calculate degree of possibility
%deviated from equation (10)
degreesOfpossibility=cell(4,3);
%S1 comparisions
%M2greaterM1(M1,M2) = solves for M2>M1
degreesOfpossibility{1,1}=M2greaterM1(S2,S1); % Solves S1>S2
degreesOfpossibility{1,2}=M2greaterM1(S3,S1); % Solves S1>S3
degreesOfpossibility{1,3}=M2greaterM1(S4,S1); %Solves S1>S4
%S2 comparisions
degreesOfpossibility{2,1}=M2greaterM1(S1,S2); % Solves S2>S1
degreesOfpossibility{2,2}=M2greaterM1(S3,S2); % Solves S2>S3
degreesOfpossibility{2,3}=M2greaterM1(S4,S2); % Solves S2>S4
%S3 comparisions
degreesOfpossibility{3,1}=M2greaterM1(S1,S3); % Solves S3>S1
degreesOfpossibility{3,2}=M2greaterM1(S2,S3); % Solves S3>S2
degreesOfpossibility{3,3}=M2greaterM1(S4,S3); % Solves S3>S4
%S4 comparisions
degreesOfpossibility{4,1}=M2greaterM1(S1,S4); % Solves S4>S1
degreesOfpossibility{4,2}=M2greaterM1(S2,S4); % Solves S4>S2
degreesOfpossibility{4,3}=M2greaterM1(S3,S4); % Solves S4>S3

degreesOfpossibility;
%calculate degree of possibility for convex Si being greater than k TFNs
%find the minimum from each group of comparisons
minVs=zeros(1,4);
for o=1:4
    row=cell2mat(degreesOfpossibility(o,:));
    minVs(o)=min(row);
end
minVs;
%Normalize the values in MinVs to obatin weight vector
tot=sum(minVs);
for x=1:4
    wSvideo(x)=minVs(x)/tot;
end
wSvideo;
%repeat for web browsing
%computation for web browsing
```

```
rowsum=cell(1,4);
for i=1:4
    ai=FCMwebBrowsing(i,:); %work out sum of TFN for each row

rowsum{i}=[ai{1}(1)+ai{2}(1)+ai{3}(1)+ai{4}(1),ai{1}(2)+ai{2}(2)+ai{3}(2)+ai{4}(2)
,ai{1}(3)+ai{2}(3)+ai{3}(3)+ai{4}(3)];
end
rowsum; %stores sum of TFN(1x3 vector) for all 4 rows/criteria
colsum = zeros(1, 3);
%% Iterate through the cell array and sum the elements
for i = 1:numel(rowsum)
   currentVector = rowsum{i};
   colsum = colsum + currentVector;
end
colsum;
colsum=flip(colsum);
inverseTFN=[1/colsum(1),1/colsum(2),1/colsum(3)];
%calculate Si values
Si_values=cell(1,4);
for k=1:4
    Si_values{k}=rowsum{k}.*inverseTFN;
end
Si_values;
S1=Si_values{1};
S2=Si_values{2};
S3=Si_values{3};
S4=Si_values{4};
%Calculate degree of possibility
%deviated from equation (10)
degreesOfpossibility=cell(4,3);
%S1 comparisions
%M2greaterM1(M1,M2) = solves for M2>M1
degreesOfpossibility{1,1}=M2greaterM1(S2,S1); % Solves S1>S2
degreesOfpossibility{1,2}=M2greaterM1(S3,S1); % Solves S1>S3
degreesOfpossibility{1,3}=M2greaterM1(S4,S1); %Solves S1>S4
%S2 comparisions
degreesOfpossibility{2,1}=M2greaterM1(S1,S2); % Solves S2>S1
degreesOfpossibility{2,2}=M2greaterM1(S3,S2); % Solves S2>S3
degreesOfpossibility{2,3}=M2greaterM1(S4,S2); % Solves S2>S4
%S3 comparisions
degreesOfpossibility{3,1}=M2greaterM1(S1,S3); % Solves S3>S1
degreesOfpossibility{3,2}=M2greaterM1(S2,S3); % Solves S3>S2
degreesOfpossibility{3,3}=M2greaterM1(S4,S3); % Solves S3>S4
%S4 comparisions
degreesOfpossibility{4,1}=M2greaterM1(S1,S4); % Solves S4>S1
degreesOfpossibility{4,2}=M2greaterM1(S2,S4); % Solves S4>S2
degreesOfpossibility{4,3}=M2greaterM1(S3,S4); % Solves S4>S3

degreesOfpossibility;
%calculate degree of possibility for convex Si being greater than k TFNs
%find the minimum from each group of comparisons
minVs=zeros(1,4);
for o=1:4
    row=cell2mat(degreesOfpossibility(o,:));
    minVs(o)=min(row);
end
minVs;
%Normalize the values in MinVs to obatin weight vector
tot=sum(minVs);
```

```
for x=1:4
    wSweb(x)=minVs(x)/tot;
end
wSweb;
wsnormvec=[wSvoice;wSvideo;wSweb]; %each row is the normalized weight vector for
each service
Ws=zeros(1,4);
total=0;
for j=1:4
    colvec=wsnormvec(:,j); %break down the rieman sum into steps
    for g=1:3
        total=total+(Ps_norm(g)*colvec(g));
    end
    Ws(j)=total;
    total=0;
end
%Ws stores the overall weight of each criterion for multiservice i.e. each
%service's weight for criterion cj  is combined
Ws

%Function to calculate convex degree of possibility
%returns degree of possibility between 2 TFNs
%inputs 2 vectors where S1=(l1,m1,u1) and S2=(l2,m2,u2)
%output Vvalue= 0,1 or some value inbetween
function Vvalue = M2greaterM1(M1,M2)
l1=M1(1);m1=M1(2);u1=M1(3);
l2=M2(1);m2=M2(2);u2=M2(3);
if (m2>=m1)
    Vvalue=1;
else
if (l1>=u2)
        Vvalue=0;
else
Vvalue=(l1-u2)/((m2-u2)-(m1-l1));
end
end
end
```

9.1.3 Appendix C: Experiment 1 for all combinations

```
%RAT selection algorithm for a single user
%Author: Bina E Mukuyamba
%Date: 17/10/2023
%Experiment 1:Part 2
%Gain calculator for 1 user
%Run 1000 times to simulate 1000 users gain calculation
%NB each user will have different network conditions
%evaluates all possible combinations to 1dp

%Update 20/10/2023
%Updated range of values for bandwidth

%0.1,0.8,0.1 was the best combination for all users in both fixed and
%dynamic network conditions

combinations=cell(1,1000); %cell to store the combination that gave highest gain
for each iteration
```

```
%Run the entire program 1000 times to be equivalent to
%Running every possible combination for 1000 users

for iterations=1:1000
%comment out when doing the dynamic network experiment
%Defining the RATs
%updated security with cost, bandwidth are in Mbps
%RAT-N = [bandwidth,cost,delay,loss rate] are our criteria
%Static conditions
%3G
RAT_1 =[1.350,3.5,30,6];
%4G
RAT_2=[2.400,4.5,60,13];
%WLAN
RAT_3=[4.500,0.5,80,10];
%5G
RAT_4=[1000,7,13,4];

%comment out when doing the static experiment
%Defining the RATs
%Dynamically changing values
%will be randomly selected from table 2 each time program is run
%NB cost is only criterion that will remain the same
%RAT-N = [bandwidth,cost,delay,loss rate]
%UPDATED
%3G
%RAT_1 =[(randi([7,20]))/10,3.5,randi([10,50]),randi([2,10])]; %divide by 10 to
%%4G
%RAT_2=[(randi([8,1000]))/10,4.5,randi([40,80]),randi([6,20])];
%%WLAN
%RAT_3=[randi([1,100]),0.5,randi([70,90]),randi([4,15])];
%%5G
%RAT_4=[randi([100,1000]),7,randi([1,25]),(randi([1,80]))/10];

%MODULE 1: WEIGHTING
%Attribute matrix= Mt
%row_i of matrix = RAT_i attribute values
%Rown=RATn, colx=criteriax_value
Mt=[RAT_1;RAT_2;RAT_3;RAT_4];
%normalized net attribute matrix Mtbar normalizing using (2)
Mt_norm=zeros(4,4);
for j=1:4
    for i=1:4
        Mt_norm(i,j)=Mt(i,j)/sum(Mt(:,j));
    end
end

%ENTROPY calculation using (3)
Entropy_values=zeros(1,4);
k=-1/log(4); %constant
vecsum=0; %initializing
for j=1:4
    Mtvec=Mt_norm(:,j); %column vector
    for i=1:4
        vecsum=vecsum+sum(Mtvec(i)*log(Mtvec(i))); %intermediate sums
    end
    Entropy_values(j)=k*vecsum; %entropy value
    vecsum=0; %don't forget to reset the running total
end
```

```
sum(Entropy_values);
%Objective weights calculation using (4)
ObjectiveWeights=zeros(1,4);

for j=1:4
    ObjectiveWeights(j)=(1-Entropy_values(j))/(4-sum(Entropy_values));
end
%Subjective user preference weights calculation
%assume 9 pt weight scale [1,9] user specified weights are randomized
%user specified weights are random for each criterion
the_weights=[1,2,3,4,5,6,7,8,9];

wu1=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
1(voice)
wu2=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
2(video)
wu3=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
3(web browsing)

Pu=[3,3,3]; %user specified service priority vector
Ps=Pu; %service determined priority=user specified priority, will be manually
changed

%normalizing weights and priority
wu1norm=zeros(1,4); %normalized weight vectors
wu2norm=zeros(1,4);
wu3norm=zeros(1,4);
s1=sum(wu1);
s2=sum(wu2);
s3=sum(wu3);
for i=1:4
    wu1norm(i)=wu1(i)/s1;
    wu2norm(i)=wu2(i)/s2;
    wu3norm(i)=wu3(i)/s3;
end
sp=sum(Pu);
Pu_norm=zeros(1,3);%normalized vector
for i=1:3
Pu_norm(i)=Pu(i)/sp;
end
Ps_norm=Pu_norm;%normalized prority vectors
%Wu user specified weight vector calculation
wunormvec=[wu1norm;wu2norm;wu3norm]; %each row is the normalized weight vector for
each service
Wu=zeros(1,4);
total=0;
for j=1:4
    colvec=wunormvec(:,j); %break down the rieman sum into steps
    for g=1:3
        total=total+(Pu_norm(g)*colvec(g));
    end
    Wu(j)=total;
    total=0;
end
Wu;
```

```
%extent analysis results
%run FAHP_EAM.m with Pu=[3,3,3]
Ws=[0.3300,0.3240,0.2097,0.1363];
%Comprehensive weight vector calculation
%Combines Wo(objective weights),Ws and Wu using weight proportion parameters
%alpha=a, beta=b, gamma=y
%W(comprehensive weights)=aWu+bWo+yWs
a=0.2;b=0.5;y=0.3;
%a=1;b=0;y=0; %user pref only
%a=0;b=1;y=0; %network conditions only
%a=0;b=0;y=1; %service charc only
Wu;
ObjectiveWeights;
Ws;
W=(a*Wu)+(b*ObjectiveWeights)+(y*Ws);
W;
%MODULE 2: UTILITY
Mt(:,1)=Mt(:,1)*1000; %in kbps for utility

u1=zeros(4,4); %utility value matrices for each service (voice)
u2=zeros(4,4); %video
u3=zeros(4,4); %web browsing
%voice utility
%updated utility function for cost for all services
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u1(i,j)=f_x("voice",colvec(j));
    end
    if i==2
    u1(i,j)=h_x2("voice",colvec(j));
    end
    if i==3
        u1(i,j)=g_x("voice",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u1(i,j)=h_x("voice",colvec(j));
    end

    end

end
u1=transpose(u1); %get the elements the right way round
%video utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u2(i,j)=f_x("video",colvec(j));
    end
    if i==2
    u2(i,j)=h_x2("video",colvec(j));
    end
    if i==3
        u2(i,j)=g_x("video",colvec(j)); %use appropriate formula per col
    end
```

```
    if i==4
        u2(i,j)=h_x("video",colvec(j));
    end

    end

end
u2=transpose(u2);
%web browsing utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u3(i,j)=f_x("web",colvec(j));
    end
    if i==2 %cost not security function
    u3(i,j)=h_x2("web",colvec(j));
    end
    if i==3
        u3(i,j)=g_x("web",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u3(i,j)=h_x("web",colvec(j));
    end

    end

end
u3=transpose(u3);
%Comprehensive utility value matrix calculation
U=zeros(4,4);
uvec={u1,u2,u3}; %using Matlab matrices
%disp(uvec{1})

%3 way loop i,j and g vary
tally=0;
for i=1:4
    for j=1:4
        for g=1:3
            usomething=uvec{g}; %u_ij for a specific service g
            tally=tally+usomething(i,j)*Ps_norm(g); %break down the sum into parts
        end
         U(i,j)=tally;
         tally=0;

    end
end
U; %final comprehensive utility value matrix

%EXPERIMENT 1: GAIN CALCULATION
%find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
%assumed the case of 1dp for simplicity
%loop through every possible combination of weight proportion parameters to
%find the highest gain for a single MMT
%under static or dynamic network conditions
%11^3 possible combinations= 1331 iterations
values=0:0.1:1;
max_gain=0;
```

```
a_gain=0;
combination_tally=0;
for n=1:length(values)
    for m=1:length(values)
        for o=1:length(values)
            %generate random decimals between 0 and 1
            num1=values(n);num2=values(m);num3=values(o);
            combination_tally=combination_tally+1;%total number of combinations
            if ((num1+num2+num3)==1 && (num1~=1 && num2~=1 && num3~=1) && (num1~=0
&& num2~=0 && num3~=0)) %only find gain if the 3 numbers add up to 1 and none of
them are 1
            %num1,num2,num3 %show them
            a_gain=find_gain(U,num1,num2,num3,Wu,Ws,ObjectiveWeights); %find gain
            if a_gain > max_gain
                max_gain=a_gain; %find the combination of parameters that gives
highest gain
                combinations{iterations}=[num1,num2,num3]; %store the combination
with highest gain
            end
            end
        end
    end
end
combination_tally; %should be 1331 to show that every combination has been
considered
end %end of the topmost for loop (1000)
combinations %see the obtained combinations
function u = f_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.25;b=48;
    u=1/(1.0+e^(-a*(x-b)));
    case "video"
    a=0.003;b=2000;
    u=1/(1.0+e^(-a*(x-b)));
    otherwise %web browsing
    a=0.01;b=564;
    u=1/(1.0+e^(-a*(x-b)));

end

end
function u = g_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.1;b=75;
    u=1-(1/(1.0+e^(-a*(x-b))));
    case "video"
    a=0.1;b=112.5;
    u=1-(1/(1.0+e^(-a*(x-b))));
    otherwise %web browsing
    a=0.03;b=375;
    u=1-(1/(1.0+e^(-a*(x-b))));

end

end
```

```
function u = h_x(service,x)
switch service
    case "voice"
    g=1/30.0;
    u=1-(g*x);
    case "video"
    g=1/30.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/30.0;
    u=1-(g*x);

end
end
%update
function u = h_x2(service,x)
switch service
    case "voice"
    g=1/50.0;
    u=1-(g*x);
    case "video"
    g=1/50.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/50.0;
    u=1-(g*x);

end

end

%function to assign weights
function w = allocateWeights(N) %N = number of users you want
%weights=zeros(N,3); %w is a vector of 3 weight vectors for each user
%assume each element of the cell corresponds to a different service
%assume w{1}=voice weights for user i
w=cell(1,N); %creates 1xN cell where each row = 1, column= N 1x3 sets of user
weights for user i
%basically a 1D array of 1x3 columns
the_weights=[1,2,3,4,5,6,7,8,9];
for i =1:length(w) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=randi([weights(1),weights(9)]);
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
    end
    w{i}=users_i_Weights; %each user-i will have 3 sets of randomized weights
end

end
function Norm = Normalize(userCell) %returns normalized weights for a cell of user
weight vectors
userCellSize=size(userCell);
Norm=cell(1,userCellSize(2)); %should be the same size as input cell
for i =1:length(Norm) %for N users
```

```
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=userCell{i}{j}(k)/sum(userCell{i}{j});
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
    end
    Norm{i}=users_i_Weights; %each user-i will have 3 sets of normalized weights
end

end

function T = find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
%a,b,y are varied to determine the best gain for our model
%using (30) from [7]
Wvec=(alpha*WU)+(beta*WO)+(gamma*WS);
gain=zeros(1,4);
total=0;
for i=1:4
    for j=1:4
        total=total+(Umat(i,j)*Wvec(j));
    end
    gain(i)=total;
    total=0;
end
T=sum(gain);

end
```

9.1.4 Appendix D: Experiment 2 implementation

```
%RAT selection algorithm
%Experiment 2:Varying priority
%Author: Bina E Mukuyamba
%Date: 24/10/2023

%Varying service priority alternate version
%each user detects different network conditions
%To simulate a dynamic network environment
%Ws FAHP weights are calculated in another program for each scenario then
%substituted into this program
%can make multiple of these to store results for each scenario

%Run_experiment(P1,P2,P3,FAHP_weights,NumUsers)
%P1,P2,P3 = priorities for each service
%FAHP_weights = service determined weights Ws
%NumUsers=Number of users to simulate, 1000 for experiment
%Ws_i values found by running FAHP_EAM.m using appropriate [P1,P2,P3]
%values then paste results in appropriate variable

Ws_1=[0.1980,0.4837,0.1820,0.1363]; %Ws for scenario 1 [5,1,3]
Ws_2=[0.2614,0.4274,0.2657,0.0454]; %Ws for scenario 2 [5,3,1]
Ws_3=[0.4620,0.1643,0.2374,0.1363]; %Ws for scenario 3 [1,5,3]
Ws_4=[0.3934,0.2677,0.2934,0.0454]; %Ws for scenario 4 [3,5,1]
Ws_5=[0.3985,0.2206,0.1536,0.2272]; %Ws for scenario 5 [1,3,5]
```

```
Ws_6=[0.2665,0.3803,0.1259,0.2272]; %Ws for scenario 6 [3,1,5]
Ws_7=[0.1581,0.5495,0.2340,0.0584]; %Ws for scenario 7 [5,1,1]
Ws_8=[0.4975,0.1389,0.3052,0.0584]; %Ws for scenario 8 [1,5,1]
Ws_9=[0.3344,0.2836,0.0899,0.2922]; %Ws for scenario 9 [1,1,5]
Ws_10=[0.3300,0.3240,0.2097,0.1363]; %Ws for scenario 10 [5,5,5]

NumUsers=input("Enter number of users to simulate")
mydata1=Run_experiment(5,1,3,Ws_1,NumUsers); %scenario1
mydata2=Run_experiment(1,5,3,Ws_2,NumUsers);
mydata3=Run_experiment(1,5,3,Ws_3,NumUsers);
mydata4=Run_experiment(3,5,1,Ws_4,NumUsers);
mydata5=Run_experiment(1,3,5,Ws_5,NumUsers);
mydata6=Run_experiment(3,1,5,Ws_6,NumUsers);
mydata7=Run_experiment(5,1,1,Ws_7,NumUsers);
mydata8=Run_experiment(1,5,1,Ws_8,NumUsers);
mydata9=Run_experiment(1,1,5,Ws_9,NumUsers);
mydata10=Run_experiment(5,5,5,Ws_10,NumUsers); %scenario 10

%plot results for all scenarios
Big_Data=[mydata1;mydata2;mydata3;mydata4;mydata5;mydata6;mydata7;mydata8;mydata9;
mydata10];
x=1:10;
bar(x,Big_Data);
xlabel("Scenario")
legend("3G","4G","WLAN","5G")
ylabel("No of users(MMTs) admitted")
title("Number of MMTs admitted into each RAT per scenario")

%plot results scenario by scenario
mybar=bar(mydata1);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 1: voice=5 video=1 web browsing=3")

mybar2=bar(mydata2);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 2: voice=5 video=3 web browsing=1")

mybar3=bar(mydata3);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 3: voice=1 video=5 web browsing=3")

mybar4=bar(mydata4);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 4: voice=3 video=5 web browsing=1")

mybar5=bar(mydata5);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 5: voice=1 video=3 web browsing=5")

mybar6=bar(mydata6);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 6: voice=3 video=1 web browsing=5")
mybar7=bar(mydata7);
```

```
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 7: voice=5 video=1 web browsing=1")

mybar8=bar(mydata8);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 8: voice=1 video=5 web browsing=1")

mybar9=bar(mydata9);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 9: voice=1 video=1 web browsing=5")

mybar10=bar(mydata10);
xlabel("RAT-N");
ylabel("No of users(MMTs) admitted");
title("Scenario 10: voice=5 video=5 web browsing=5")

%NB updated utility functions to include cost
%Utility functions as defined in table 3 in [7]
function u = f_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.25;b=48;
    u=1/(1.0+e^(-a*(x-b)));
    case "video"
    a=0.003;b=2000;
    u=1/(1.0+e^(-a*(x-b)));
    otherwise %web browsing
    a=0.01;b=564;
    u=1/(1.0+e^(-a*(x-b)));

end

%Not used in this program
%Function to calculate convex degree of possibility
%returns degree of possibility between 2 TFNs
%inputs 2 vectors where S1=(l1,m1,u1) and S2=(l2,m2,u2)
%output Vvalue= 0,1 or some value inbetween
function Vvalue = M2greaterM1(M1,M2)
l1=M1(1);m1=M1(2);u1=M1(3);
l2=M2(1);m2=M2(2);u2=M2(3);
if (m2>=m1)
    Vvalue=1;
else
if (l1>=u2)
        Vvalue=0;
else
Vvalue=(l1-u2)/((m2-u2)-(m1-l1));
end
end
end
end

function u = g_x(service,x)
e=exp(1);
switch service
```

```
    case "voice"
    a=0.1;b=75;
    u=1-(1/(1.0+e^(-a*(x-b))));
    case "video"
    a=0.1;b=112.5;
    u=1-(1/(1.0+e^(-a*(x-b))));
    otherwise %web browsing
    a=0.03;b=375;
    u=1-(1/(1.0+e^(-a*(x-b))));

end

end
function u = h_x(service,x)
switch service
    case "voice"
    g=1/30.0;
    u=1-(g*x);
    case "video"
    g=1/30.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/30.0;
    u=1-(g*x);

end
end
%updated for cost
function u = h_x2(service,x)
switch service
    case "voice"
    g=1/50.0;
    u=1-(g*x);
    case "video"
    g=1/50.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/50.0;
    u=1-(g*x);

end

end

%function to assign weights
function w = allocateWeights(N) %N = number of users you want
%weights=zeros(N,3); %w is a vector of 3 weight vectors for each user
%assume each element of the cell corresponds to a different service
%assume w{1}=voice weights for user i
w=cell(1,N); %creates 1xN cell where each row = 1, column= N 1x3 sets of user
weights for user i
%basically a 1D array of 1x3 columns
the_weights=[1,2,3,4,5,6,7,8,9];
for i =1:length(w) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=randi([weights(1),weights(9)]);
```

```
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
     end
     w{i}=users_i_Weights; %each user-i will have 3 sets of randomized weights
 end

 end

 %returns normalized weights for a cell of user weight vectors
 function Norm = Normalize(userCell)
 userCellSize=size(userCell);
 Norm=cell(1,userCellSize(2)); %should be the same size as input cell
 for i =1:length(Norm) %for N users
     for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=userCell{i}{j}(k)/sum(userCell{i}{j});
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
     end
     Norm{i}=users_i_Weights; %each user-i will have 3 sets of normalized weights
 end

 end
 %Not used in this program
 function T = find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
 %a,b,y are varied to determine the best gain for our model
 %using (30) from [7]
 Wvec=(alpha*WU)+(beta*WO)+(gamma*WS);
 gain=zeros(1,4);
 total=0;
 for i=1:4
     for j=1:4
         total=total+(Umat(i,j)*Wvec(j));
     end
     gain(i)=total;
     total=0;
 end
 T=sum(gain);

 end

 %Function to run experiment 2
 %Implements all the above functions to calculate weights and unity
 %returns an array which stores the number of users in each RAT after
 %running the selections for 1000 users
 %inputs P1,P2,P3 = service priorities
 %FAHP_Weights = Ws for a particular priority vector [P1,P2,P3]
 %Num_Users = Number of users to simulate
 %Output The_RATS = array to store number of users per RAT
 function The_RATS = Run_experiment(P1,P2,P3,FAHP_weights,NumUsers)
 The_RATS=zeros(1,4); %array to store number of users in each RAT
 for iterations=1:NumUsers %no of users to simulate

 %Defining the RATs
 %Dynamically changing values
```

```
%net attr will be randomly selected from table 2 for each iteration/user
%NB cost is only criterion that will remain the same
%RAT-N = [bandwidth,cost,delay,loss rate]
%3G
RAT_1 =[(randi([7,20]))/10,3.5,randi([10,50]),randi([2,10])]; %divide by 10 to
generate random decimals
%4G
RAT_2=[(randi([8,1000]))/10,4.5,randi([40,80]),randi([6,20])];
%WLAN
RAT_3=[randi([1,100]),0.5,randi([70,90]),randi([4,15])];
%5G
RAT_4=[randi([100,1000]),7,randi([1,25]),(randi([1,80]))/10];

%MODULE 1: WEIGHTING
%Attribute matrix= Mt
%row_i of matrix = RAT_i attribute values
%Rown=RATn, colx=criteriax_value
Mt=[RAT_1;RAT_2;RAT_3;RAT_4];
%normalized net attribute matrix Mtbar normalizing using (2)
Mt_norm=zeros(4,4);
for j=1:4
    for i=1:4
        Mt_norm(i,j)=Mt(i,j)/sum(Mt(:,j));
    end
end

%ENTROPY calculation using (3)
Entropy_values=zeros(1,4);
k=-1/log(4); %constant
vecsum=0; %initializing
for j=1:4
    Mtvec=Mt_norm(:,j); %column vector
    for i=1:4
        vecsum=vecsum+sum(Mtvec(i)*log(Mtvec(i))); %intermediate sums
    end
    Entropy_values(j)=k*vecsum; %entropy value
    vecsum=0; %don't forget to reset the running total
end
sum(Entropy_values);
%Objective weights calculation using (4)
ObjectiveWeights=zeros(1,4);

for j=1:4
    ObjectiveWeights(j)=(1-Entropy_values(j))/(4-sum(Entropy_values));
end
%Subjective user preference weights calculation
%assume 9 pt weight scale [1,9] user specified weights are randomized
the_weights=[1,2,3,4,5,6,7,8,9];

wu1=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
1(voice)
wu2=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
2(video)
wu3=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
3(web browsing)
```

```
%Now a variable passed into function
Pu=[P1,P2,P3]; %user specified service priority vector
Ps=Pu; %service determined priority=user specified priority, will be manually
changed

%normalizing weights and priority
wu1norm=zeros(1,4); %normalized weight vectors
wu2norm=zeros(1,4);
wu3norm=zeros(1,4);
s1=sum(wu1);
s2=sum(wu2);
s3=sum(wu3);
for i=1:4
    wu1norm(i)=wu1(i)/s1;
    wu2norm(i)=wu2(i)/s2;
    wu3norm(i)=wu3(i)/s3;
end
sp=sum(Pu);
Pu_norm=zeros(1,3);%normalized vector
for i=1:3
Pu_norm(i)=Pu(i)/sp;
end
Ps_norm=Pu_norm;%normalized prority vectors
%Wu user specified weight vector calculation
wunormvec=[wu1norm;wu2norm;wu3norm]; %each row is the normalized weight vector for
each service
Wu=zeros(1,4);
total=0;
for j=1:4
    colvec=wunormvec(:,j); %break down the rieman sum into steps
    for g=1:3
        total=total+(Pu_norm(g)*colvec(g));
    end
    Wu(j)=total;
    total=0;
end
Wu;
Ws=FAHP_weights;
%Comprehensive weight vector calculation
%Combines Wo(objective weights),Ws and Wu using weight proportion parameters
%alpha=a, beta=b, gamma=y
%W(comprehensive weights)=aWu+bWo+yWs
a=0.2;b=0.5;y=0.3; %from Experiment 1
Wu;
ObjectiveWeights;
Ws;
W=(a*Wu)+(b*ObjectiveWeights)+(y*Ws);
W;

%MODULE 2: UTILITY
Mt(:,1)=Mt(:,1)*1000; %in kbps for utility
u1=zeros(4,4); %utility value matrices for each service (voice)
u2=zeros(4,4); %video
u3=zeros(4,4); %web browsing

%voice utility
%updated utility function for cost for all services
for i=1:4
    colvec=Mt(:,i);%colum vector
```

```
    for j=1:4
    %colvec(j);
    if i==1
    u1(i,j)=f_x("voice",colvec(j));
    end
    if i==2
    u1(i,j)=h_x2("voice",colvec(j));
    end
    if i==3
        u1(i,j)=g_x("voice",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u1(i,j)=h_x("voice",colvec(j));
    end

    end

end
u1=transpose(u1); %get the elements the right way round

%video utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u2(i,j)=f_x("video",colvec(j));
    end
    if i==2
    u2(i,j)=h_x2("video",colvec(j));
    end
    if i==3
        u2(i,j)=g_x("video",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u2(i,j)=h_x("video",colvec(j));
    end

    end

end
u2=transpose(u2);
%web browsing utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u3(i,j)=f_x("web",colvec(j));
    end
    if i==2 %cost not security function
    u3(i,j)=h_x2("web",colvec(j));
    end
    if i==3
        u3(i,j)=g_x("web",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u3(i,j)=h_x("web",colvec(j));
    end
```

```
    end

end
u3=transpose(u3);
%Comprehensive utility value matrix calculation
U=zeros(4,4);
uvec={u1,u2,u3}; %using Matlab matrices
%3 way loop i,j and g vary
tally=0;
for i=1:4
    for j=1:4
        for g=1:3
            usomething=uvec{g}; %u_ij for a specific service g
            tally=tally+usomething(i,j)*Ps_norm(g); %break down the sum into parts
        end
         U(i,j)=tally;
         tally=0;

    end
end
U; %final comprehensive utility value matrix

%MODULE 3: NETWORK RANKING AND SELECTION
%Create normalized decison matrix (NDM)
NDM=U;
%create weighted NDM by multiplying each row with W
D=zeros(4,4);
for i=1:4
    for j=1:4
        D(i,j)=NDM(i,j)*W(j);
    end
end
D;
%TOPSIS code
Dplus=zeros(1,4);
Dminus=zeros(1,4);
%store the ideal solutions D+ and D-
for i=1:4
    colvec=D(:,i);
        if i==1%update
        Dplus(i)=max(colvec);
        Dminus(i)=min(colvec);
        end
        if i>1 %update
        Dplus(i)=min(colvec);
        Dminus(i)=max(colvec);
        end

end

%calculate euclidean distances Si+ and Si-
%for each RAT to ideal solution
Siplus=zeros(1,4);
Siminus=zeros(1,4);%vectors to store each Si value
tally1=0;
tally2=0;
for i=1:4
    for j=1:4
```

```
        %Fixed TOPSIS
    tally1=tally1+(D(i,j)-Dplus(j))^2; %sum of squared diff
    tally2=tally2+(D(i,j)-Dminus(j))^2; %sum of products (22)
    end
    Siplus(i)=sqrt(tally1);
    Siminus(i)=sqrt(tally2);
    tally1=0;
    tally2=0;
end
%Use (23) to calculate the score of each RAT
SC=zeros(1,4); %array to store each RATs score
for i=1:4
    SC(i)=Siminus(i)/(Siminus(i)+Siplus(i));
end

%The best RAT is the one with the highest score
Best_RAT=0;
for i=1:4
score=SC(i);
if score==max(SC)
    Best_RAT=i; %assign RAT number of best RAT to variable
end
end
%count number of users in RAT 1
if (Best_RAT==1)
 The_RATS(1)=The_RATS(1)+1;
end
%count number of users in RAT 2
if (Best_RAT==2)
 The_RATS(2)=The_RATS(2)+1;
end
%count number of users in RAT 3
if (Best_RAT==3)
 The_RATS(3)=The_RATS(3)+1;
end
%count number of users in RAT 4
if (Best_RAT==4)
 The_RATS(4)=The_RATS(4)+1;
end

end %end of a single selection event for one user

The_RATS;

end
```

9.1.5 Appendix E: Experiment 2 for specific user assigned weights

The code was exactly the same as appendix D but the run_experiment method included the following snippet.

```
%EXPERIMENT 2 PART 2
%the weights of each criterion are biased to reflect importance to the user
%criterion deemed important has weight [7,9] while the rest are given [1,3]
%For the first part this is done for each criterion in all services
%For the second part this is done for all criteria for one service
%Uncomment a particular case to simulate it
```

```
 %Subjective user preference weights calculation
 %assume 9 pt weight scale [1,9] user specified weights are randomized
 the_weights=[1,2,3,4,5,6,7,8,9];

 %Part 1
 %case 1: Bandwidth more important to users
 wu1=[randi([weights(7),weights(9)]),randi([weights(1),weights(3)]),randi([weights(
1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for service
1(voice)
 wu2=[randi([weights(7),weights(9)]),randi([weights(1),weights(3)]),randi([weights(
1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for service
2(video)
 wu3=[randi([weights(7),weights(9)]),randi([weights(1),weights(3)]),randi([weights(
1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for service
3(web browsing)

 %case 2: Cost more important to users
 %wu1=[randi([weights(1),weights(3)]),randi([weights(7),weights(9)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 1(voice)
 %wu2=[randi([weights(1),weights(3)]),randi([weights(7),weights(9)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 2(video)
 %wu3=[randi([weights(1),weights(3)]),randi([weights(7),weights(9)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 3(web browsing)

 %case 3: Delay more important to users
 %wu1=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(7),weights(9)]),randi([weights(1),weights(3)])]; %user defined weights for
service 1(voice)
 %wu2=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(7),weights(9)]),randi([weights(1),weights(3)])]; %user defined weights for
service 2(video)
 %wu3=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(7),weights(9)]),randi([weights(1),weights(3)])]; %user defined weights for
service 3(web browsing)

 %Case 4:Loss rate more important to users
 %wu1=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(7),weights(9)])]; %user defined weights for
service 1(voice)
 %wu2=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(7),weights(9)])]; %user defined weights for
service 2(video)
 %wu3=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(7),weights(9)])]; %user defined weights for
service 3(web browsing)

 %Part 2
 %Case 5: Voice more important to user
 %wu1=[randi([weights(7),weights(9)]),randi([weights(7),weights(9)]),randi([weights
(7),weights(9)]),randi([weights(7),weights(9)])]; %user defined weights for
service 1(voice)
 %wu2=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 2(video)
```

```
 %wu3=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 3(web browsing)

 %Case 6: Video more important to user
 %wu1=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 1(voice)
 %wu2=[randi([weights(7),weights(9)]),randi([weights(7),weights(9)]),randi([weights
(7),weights(9)]),randi([weights(7),weights(9)])]; %user defined weights for
service 2(video)
 %wu3=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 3(web browsing)

 %Case 7: Web browsing more important to user
 %wu1=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 1(voice)
 %wu2=[randi([weights(1),weights(3)]),randi([weights(1),weights(3)]),randi([weights
(1),weights(3)]),randi([weights(1),weights(3)])]; %user defined weights for
service 2(video)
 %wu3=[randi([weights(7),weights(9)]),randi([weights(7),weights(9)]),randi([weights
(7),weights(9)]),randi([weights(7),weights(9)])]; %user defined weights for
service 3(web browsing)
```

9.1.6 Appendix F: Experiment 3 implementation

```
 %RAT selection algorithm
 %Author: Bina E Mukuyamba
 %Date: 17/10/2023

 %Updates
 %Fixed TOPSIS equations and bandwidth values
 %Added extent analysis FAHP

 %Experiement 3 part 1: Vary number of MMTs for a fixed threshold
 %Experiment 3 part 2: Vary the threshold for a fixed no of MMTs
 %Plots were hardcoded by running code several times

 %NB each user will have different random network conditions based on table
 %2

 %Each MMT is assigned a variable called current_RAT
 %which represents the current RAT it is connected to when the simulation
 %begins(at time 0)
 %current_RAT=1 implies the MMT is connected to RAT-1 initially
 %likewise current_RAT=2,3 or 4 means it is connected to RAT 2,3 or 4 respectively
 %The simulation runs and calculates the optimal RAT for the MMT
 %Then the threshold condition is evaluated to determine whether to handoff
 %or not

 %Experiment 3.1 vary NumUsers keep threshold constant
 No_of_handovers=0; %zero handovers initially

 NumUsers=input("Enter number of users to simulate")
 for iterations=1:NumUsers %no of users to simulate
```

```
%Defining the RATs
%Dynamically changing values
%will be randomly selected from table 2 each time program is run
%NB cost is only criterion that will remain the same
%RAT-N = [bandwidth,cost,delay,loss rate]
%3G
RAT_1 =[(randi([7,20]))/10,3.5,randi([10,50]),randi([2,10])]; %divide by 10 to
generate random decimals
%4G
RAT_2=[(randi([8,1000]))/10,4.5,randi([40,80]),randi([6,20])];
%WLAN
RAT_3=[randi([1,100]),0.5,randi([70,90]),randi([4,15])];
%5G
RAT_4=[randi([100,1000]),7,randi([1,25]),(randi([1,80]))/10];

current_RAT=randi([1,4]); %initial RAT connected to MMT will be random each
iteration
%disp(["initial RAT is:",num2str(current_RAT)])  %check initial RAT

%Experiment 3.2 vary this, keep no of users constant
threshold=0; %threshold for handover from [7]


%MODULE 1: WEIGHTING
%Attribute matrix= Mt
%row_i of matrix = RAT_i attribute values
%Rown=RATn, colx=criteriax_value
Mt=[RAT_1;RAT_2;RAT_3;RAT_4];
%normalized net attribute matrix Mtbar normalizing using (2)
Mt_norm=zeros(4,4);
for j=1:4
    for i=1:4
        Mt_norm(i,j)=Mt(i,j)/sum(Mt(:,j));
    end
end

%ENTROPY calculation using (3)
Entropy_values=zeros(1,4);
k=-1/log(4); %constant
vecsum=0; %initializing
for j=1:4
    Mtvec=Mt_norm(:,j); %column vector
    for i=1:4
        vecsum=vecsum+sum(Mtvec(i)*log(Mtvec(i))); %intermediate sums
    end
    Entropy_values(j)=k*vecsum; %entropy value
    vecsum=0; %don't forget to reset the running total
end
sum(Entropy_values);
%Objective weights calculation using (4)
ObjectiveWeights=zeros(1,4);

for j=1:4
    ObjectiveWeights(j)=(1-Entropy_values(j))/(4-sum(Entropy_values));
end
%Subjective user preference weights calculation
%assume 9 pt weight scale [1,9] user specified weights are randomized
the_weights=[1,2,3,4,5,6,7,8,9];
```

```
 wu1=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
1(voice)
 wu2=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
2(video)
 wu3=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
3(web browsing)

 Pu=[3,3,3]; %user specified service priority vector
 Ps=Pu; %service determined priority=user specified priority, will be manually
changed

 %normalizing weights and priority
 wu1norm=zeros(1,4); %normalized weight vectors
 wu2norm=zeros(1,4);
 wu3norm=zeros(1,4);
 s1=sum(wu1);
 s2=sum(wu2);
 s3=sum(wu3);
 for i=1:4
     wu1norm(i)=wu1(i)/s1;
     wu2norm(i)=wu2(i)/s2;
     wu3norm(i)=wu3(i)/s3;
 end
 sp=sum(Pu);
 Pu_norm=zeros(1,3);%normalized vector
 for i=1:3
 Pu_norm(i)=Pu(i)/sp;
 end
 Ps_norm=Pu_norm;%normalized prority vectors
 %Wu user specified weight vector calculation
 wunormvec=[wu1norm;wu2norm;wu3norm]; %each row is the normalized weight vector for
each service
 Wu=zeros(1,4);
 total=0;
 for j=1:4
     colvec=wunormvec(:,j); %break down the rieman sum into steps
     for g=1:3
         total=total+(Pu_norm(g)*colvec(g));
     end
     Wu(j)=total;
     total=0;
 end
 Wu;
 %run FAHP_EAM.m with P=[3,3,3] to get these weights
 Ws=[0.3300,0.3240,0.2097,0.1363];

 %Comprehensive weight vector calculation
 %Combines Wo(objective weights),Ws and Wu using weight proportion parameters
 %alpha=a, beta=b, gamma=y
 %W(comprehensive weights)=aWu+bWo+yWs
 a=0.2;b=0.5;y=0.3; %from Exp1
 %a=0.1,b=0.8,c=0.1;
 Wu;
 ObjectiveWeights;
 Ws;
```

```
W=(a*Wu)+(b*ObjectiveWeights)+(y*Ws);
W;

%MODULE 2: UTILITY
Mt(:,1)=Mt(:,1)*1000; %in kbps for utility
u1=zeros(4,4); %utility value matrices for each service (voice)
u2=zeros(4,4); %video
u3=zeros(4,4); %web browsing

%voice utility
%updated utility function for cost for all services
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u1(i,j)=f_x("voice",colvec(j));
    end
    if i==2
    u1(i,j)=h_x2("voice",colvec(j));
    end
    if i==3
        u1(i,j)=g_x("voice",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u1(i,j)=h_x("voice",colvec(j));
    end

    end

end
u1=transpose(u1); %get the elements the right way round
%video utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u2(i,j)=f_x("video",colvec(j));
    end
    if i==2
    u2(i,j)=h_x2("video",colvec(j));
    end
    if i==3
        u2(i,j)=g_x("video",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u2(i,j)=h_x("video",colvec(j));
    end

    end

end
u2=transpose(u2);
%web browsing utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
```

```
    if i==1
    u3(i,j)=f_x("web",colvec(j));
    end
    if i==2 %cost not security function
    u3(i,j)=h_x2("web",colvec(j));
    end
    if i==3
        u3(i,j)=g_x("web",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u3(i,j)=h_x("web",colvec(j));
    end

    end

end
u3=transpose(u3);
%Comprehensive utility value matrix calculation
U=zeros(4,4);
uvec={u1,u2,u3}; %using Matlab matrices
%disp(uvec{1})

%3 way loop i,j and g vary
tally=0;
for i=1:4
    for j=1:4
        for g=1:3
            usomething=uvec{g}; %u_ij for a specific service g
            tally=tally+usomething(i,j)*Ps_norm(g); %break down the sum into parts
        end
         U(i,j)=tally;
         tally=0;

    end
end
U; %final comprehensive utility value matrix

%MODULE 3: NETWORK RANKING AND SELECTION
%Create normalized decison matrix (NDM)
NDM=U;
%create weighted NDM by multiplying each row with W
D=zeros(4,4);
for i=1:4
    for j=1:4
        D(i,j)=NDM(i,j)*W(j);
    end
end
D;
%TOPSIS code
Dplus=zeros(1,4);
Dminus=zeros(1,4);
%store the ideal solutions D+ and D-
for i=1:4
    colvec=D(:,i);
        if i==1%update
        Dplus(i)=max(colvec);
        Dminus(i)=min(colvec);
        end
        if i>1 %update
```

```
        Dplus(i)=min(colvec);
        Dminus(i)=max(colvec);
        end

end
%calculate euclidean distances Si+ and Si-
%for each RAT to ideal solution
Siplus=zeros(1,4);
Siminus=zeros(1,4);%vectors to store each Si value
tally1=0;
tally2=0;
for i=1:4
    for j=1:4
        %Fixed TOPSIS
    tally1=tally1+(D(i,j)-Dplus(j))^2;
    tally2=tally2+(D(i,j)-Dminus(j))^2;
    end
    Siplus(i)=sqrt(tally1);
    Siminus(i)=sqrt(tally2);
    tally1=0;
    tally2=0;
end

% calculate the score of each RAT
SC=zeros(1,4); %array to store each RATs score
for i=1:4
    SC(i)=Siminus(i)/(Siminus(i)+Siplus(i));
end

%The best RAT is the one with the highest score
Best_RAT=0;
for i=1:4
score=SC(i);
if score==max(SC)
    Best_RAT=i;
end
end

%EXPERIMENT 3:Determining number of Handovers
%Assumed current_RAT will be directly mapped to the index position of the
%corresponding RAT score in the score array
%i.e. SC(current_RAT)=score for the current RAT
if (current_RAT==Best_RAT) %if MMT connected to best already do nothing
    current_RAT=Best_RAT;
end
if (max(SC)/SC(current_RAT)>threshold) %using equation (26)
        current_RAT=Best_RAT; %handover calls to the best RAT
        No_of_handovers=No_of_handovers+1; %increment number of handovers
else
    current_RAT=current_RAT; %maintain current connection
end
%observe if RAT changes
%disp(["new RAT is:",num2str(current_RAT)]);
end %end of a single selection event

No_of_handovers %show number of handovers
%plotting the data
%These values were manually recorded after running above experiment
%for each case
```

```
%HandoverArray=[0,35,77,125,162,199,229,274,298,344,387]; %array to store the number of handovers for varying number of users
%NumUserArray=[0,100,200,300,400,500,600,700,800,900,1000]; %Array to store the different number of users used in experiment
%plot(NumUserArray, HandoverArray)
%xlabel('Number of users');
%ylabel('Number of handovers');
%title('Handover against number of users for fixed threshold (\sigma)=1.3');
%grid on;

HandoverArray2=[1000,1000,1000,1000,762,609,587,470,381,337,314,285,277,266,252,248,247,228,250,242,217,137,44,0]; %array to store the number of handovers for varying number of users
Thresholds=[0.03,0.3,0.6,0.9,1,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2,2.1,2.4,2.7,3,4,7,10,15,20]; %Array to store the different number of users used in experiment
numel(HandoverArray2);
numel(Thresholds);
plot(Thresholds, HandoverArray2)
xlabel('Threshold (\sigma)');
ylabel('Number of handovers');
title('number of handovers against threshold for 1000 users');
grid on;


%NB updated utility functions to include cost
function u = f_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.25;b=48;
    u=1/(1.0+e^(-a*(x-b)));
    case "video"
    a=0.003;b=2000;
    u=1/(1.0+e^(-a*(x-b)));
    otherwise %web browsing
    a=0.01;b=564;
    u=1/(1.0+e^(-a*(x-b)));

end

%Function to calculate convex degree of possibility
%returns degree of possibility between 2 TFNs
%inputs 2 vectors where S1=(l1,m1,u1) and S2=(l2,m2,u2)
%output Vvalue= 0,1 or some value inbetween
function Vvalue = M2greaterM1(M1,M2)
l1=M1(1);m1=M1(2);u1=M1(3);
l2=M2(1);m2=M2(2);u2=M2(3);
if (m2>=m1)
    Vvalue=1;
else
if (l1>=u2)
        Vvalue=0;
else
Vvalue=(l1-u2)/((m2-u2)-(m1-l1));
end
end
end
end
function u = g_x(service,x)
```

```
e=exp(1);
switch service
    case "voice"
    a=0.1;b=75;
    u=1-(1/(1.0+e^(-a*(x-b))));
    case "video"
    a=0.1;b=112.5;
    u=1-(1/(1.0+e^(-a*(x-b))));
    otherwise %web browsing
    a=0.03;b=375;
    u=1-(1/(1.0+e^(-a*(x-b))));

end

end
function u = h_x(service,x)
switch service
    case "voice"
    g=1/30.0;
    u=1-(g*x);
    case "video"
    g=1/30.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/30.0;
    u=1-(g*x);

end
end
%update
function u = h_x2(service,x)
switch service
    case "voice"
    g=1/50.0;
    u=1-(g*x);
    case "video"
    g=1/50.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/50.0;
    u=1-(g*x);

end

end

%function to assign weights
function w = allocateWeights(N) %N = number of users you want
%weights=zeros(N,3); %w is a vector of 3 weight vectors for each user
%assume each element of the cell corresponds to a different service
%assume w{1}=voice weights for user i
w=cell(1,N); %creates 1xN cell where each row = 1, column= N 1x3 sets of user
weights for user i
%basically a 1D array of 1x3 columns
the_weights=[1,2,3,4,5,6,7,8,9];
for i =1:length(w) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
```

```
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=randi([weights(1),weights(9)]);
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
        end
        w{i}=users_i_Weights; %each user-i will have 3 sets of randomized weights
    end

    end
    function Norm = Normalize(userCell) %returns normalized weights for a cell of user
weight vectors
    userCellSize=size(userCell);
    Norm=cell(1,userCellSize(2)); %should be the same size as input cell
    for i =1:length(Norm) %for N users
        for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=userCell{i}{j}(k)/sum(userCell{i}{j});
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
        end
        Norm{i}=users_i_Weights; %each user-i will have 3 sets of normalized weights
    end

    end

    function T = find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
    %a,b,y are varied to determine the best gain for our model
    %using (30) from [7]
    Wvec=(alpha*WU)+(beta*WO)+(gamma*WS);
    gain=zeros(1,4);
    total=0;
    for i=1:4
        for j=1:4
            total=total+(Umat(i,j)*Wvec(j));
        end
        gain(i)=total;
        total=0;
    end
    T=sum(gain);

    end
```

9.1.7 Appendix G: Experiment 3 implementation of unnecessary handover

```
%RAT selection algorithm
%Author: Bina E Mukuyamba
%Date: 17/10/2023

%Experiement 3 part 3: Calculating the number of unnecessary Handovers
%threshold=1.3 and number of users kept constant=1000
```

```
%UPDATE
%fixed TOPSIS equation and bandwidth values
%Added Ws weights by extent analyis

%NB each user/iteration will have different random network conditions based on
table
%2
%Each iteration of the loop represents the handover/selection for a
%differnt user in the network

%Variables
%last_RAT = RAT MMT connected to in prior iteration
%current_RAT = RAT MMT connected to when the current iteration starts
%No_of_handovers = Total number of handovers
%unnecessary_handovers = number of unnecessary handovers
%t1= keep track of the current iteration number
%t2 = keeps track of the previous iteration to determine if handover
%happens in successive events

%current_RAT=0 means the MMT is not connected to any RAT initially
%current_RAT=1 implies the MMT is connected to RAT-1 initially
%last_RAT=1 means MMT was connected to RAT 1 in previous
%iteration/selection event
%likewise current_RAT=2,3 or 4 means it is connected to RAT 2,3 or 4 respectively
%The simulation runs and calculates the optimal RAT for the MMT
%Then the threshold condition is evaluated to determine whether to handoff
%or not
%Unnecessary handovers are counted if they happen

No_of_handovers=0; %zero handovers initially
unnecessary_handovers=0; %initialize variables
current_RAT=0;
last_RAT=0;
t1=0;
t2=0;

NumUsers=input("Enter number of users to simulate")
for iterations=1:NumUsers %no of users to simulate

%Defining the RATs
%Dynamically changing values
%will be randomly selected from table 2 each time program is run
%NB cost is only criterion that will remain the same
%RAT-N = [bandwidth,cost,delay,loss rate]
%3G
RAT_1 =[(randi([7,20]))/10,3.5,randi([10,50]),randi([2,10])]; %divide by 10 to
generate random decimals
%4G
RAT_2=[(randi([8,1000]))/10,4.5,randi([40,80]),randi([6,20])];
%WLAN
RAT_3=[randi([1,100]),0.5,randi([70,90]),randi([4,15])];
%5G
RAT_4=[randi([100,1000]),7,randi([1,25]),(randi([1,80]))/10];

%current_RAT=randi([0,4]); %initial RAT connected to MMT will be random each
iteration
%disp(["initial RAT is:",num2str(current_RAT)])  %check initial RAT
%disp(["last RAT is:",num2str(last_RAT)])
```

```
threshold=1.3; %threshold for handover from [7]

%MODULE 1: WEIGHTING
%Attribute matrix= Mt
%row_i of matrix = RAT_i attribute values
%Rown=RATn, colx=criteriax_value
Mt=[RAT_1;RAT_2;RAT_3;RAT_4];
%normalized net attribute matrix Mtbar normalizing using (2)
Mt_norm=zeros(4,4);
for j=1:4
    for i=1:4
        Mt_norm(i,j)=Mt(i,j)/sum(Mt(:,j));
    end
end

%ENTROPY calculation using (3)
Entropy_values=zeros(1,4);
k=-1/log(4); %constant
vecsum=0; %initializing
for j=1:4
    Mtvec=Mt_norm(:,j); %column vector
    for i=1:4
        vecsum=vecsum+sum(Mtvec(i)*log(Mtvec(i))); %intermediate sums
    end
    Entropy_values(j)=k*vecsum; %entropy value
    vecsum=0; %don't forget to reset the running total
end
sum(Entropy_values);
%Objective weights calculation using (4)
ObjectiveWeights=zeros(1,4);

for j=1:4
    ObjectiveWeights(j)=(1-Entropy_values(j))/(4-sum(Entropy_values));
end
%Subjective user preference weights calculation
%assume 9 pt weight scale [1,9] user specified weights are randomized
the_weights=[1,2,3,4,5,6,7,8,9];

wu1=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
1(voice)
wu2=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
2(video)
wu3=[randi([weights(1),weights(9)]),randi([weights(1),weights(9)]),randi([weights(
1),weights(9)]),randi([weights(1),weights(9)])]; %user defined weights for service
3(web browsing)

Pu=[5,5,5]; %user specified service priority vector
Ps=Pu; %service determined priority=user specified priority, will be manually
changed

%normalizing weights and priority
wu1norm=zeros(1,4); %normalized weight vectors
wu2norm=zeros(1,4);
wu3norm=zeros(1,4);
s1=sum(wu1);
s2=sum(wu2);
s3=sum(wu3);
```

```
for i=1:4
    wu1norm(i)=wu1(i)/s1;
    wu2norm(i)=wu2(i)/s2;
    wu3norm(i)=wu3(i)/s3;
end
sp=sum(Pu);
Pu_norm=zeros(1,3);%normalized vector
for i=1:3
Pu_norm(i)=Pu(i)/sp;
end
Ps_norm=Pu_norm;%normalized prority vectors
%Wu user specified weight vector calculation
wunormvec=[wu1norm;wu2norm;wu3norm]; %each row is the normalized weight vector for
each service
Wu=zeros(1,4);
total=0;
for j=1:4
    colvec=wunormvec(:,j); %break down the rieman sum into steps
    for g=1:3
        total=total+(Pu_norm(g)*colvec(g));
    end
    Wu(j)=total;
    total=0;
end
Wu;
%run FAHP_EAM.m with P=[3,3,3] to get these weights
Ws=[0.3300,0.3240,0.2097,0.1363];


%Comprehensive weight vector calculation
%Combines Wo(objective weights),Ws and Wu using weight proportion parameters
%alpha=a, beta=b, gamma=y
%W(comprehensive weights)=aWu+bWo+yWs
%a=0.2;b=0.5;y=0.3; %from exp1
%comparing with [7]
a=1/3;b=1/3;y=1/3;
Wu;
ObjectiveWeights;
Ws;
W=(a*Wu)+(b*ObjectiveWeights)+(y*Ws);
W;

%MODULE 2: UTILITY
Mt(:,1)=Mt(:,1)*1000; %in kbps for utility
u1=zeros(4,4); %utility value matrices for each service (voice)
u2=zeros(4,4); %video
u3=zeros(4,4); %web browsing
%voice utility
%updated utility function for cost for all services
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u1(i,j)=f_x("voice",colvec(j));
    end
    if i==2
    u1(i,j)=h_x2("voice",colvec(j));
    end
```

```
    if i==3
        u1(i,j)=g_x("voice",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u1(i,j)=h_x("voice",colvec(j));
    end

    end

end
u1=transpose(u1); %get the elements the right way round

%video utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u2(i,j)=f_x("video",colvec(j));
    end
    if i==2
    u2(i,j)=h_x2("video",colvec(j));
    end
    if i==3
        u2(i,j)=g_x("video",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u2(i,j)=h_x("video",colvec(j));
    end

    end

end
u2=transpose(u2);
%web browsing utility
for i=1:4
    colvec=Mt(:,i);%colum vector
    for j=1:4
    %colvec(j);
    if i==1
    u3(i,j)=f_x("web",colvec(j));
    end
    if i==2 %cost not security function
    u3(i,j)=h_x2("web",colvec(j));
    end
    if i==3
        u3(i,j)=g_x("web",colvec(j)); %use appropriate formula per col
    end
    if i==4
        u3(i,j)=h_x("web",colvec(j));
    end

    end

end
u3=transpose(u3);
%Comprehensive utility value matrix calculation
U=zeros(4,4);
uvec={u1,u2,u3}; %using Matlab matrices
```

```
%3 way loop i,j and g vary
tally=0;
for i=1:4
    for j=1:4
        for g=1:3
            usomething=uvec{g}; %u_ij for a specific service g
            tally=tally+usomething(i,j)*Ps_norm(g); %break down the sum into parts
        end
         U(i,j)=tally;
         tally=0;

    end
end
U; %final comprehensive utility value matrix

%MODULE 3: NETWORK RANKING AND SELECTION
%Create normalized decison matrix (NDM)
NDM=U;
%create weighted NDM by multiplying each row with W
D=zeros(4,4);
for i=1:4
    for j=1:4
        D(i,j)=NDM(i,j)*W(j);
    end
end
D;
%TOPSIS code
Dplus=zeros(1,4);
Dminus=zeros(1,4);
%store the ideal solutions D+ and D-
for i=1:4
    colvec=D(:,i);
        if i==1%update
        Dplus(i)=max(colvec);
        Dminus(i)=min(colvec);
        end
        if i>1 %update
        Dplus(i)=min(colvec);
        Dminus(i)=max(colvec);
        end

end
%calculate euclidean distances Si+ and Si-
%for each RAT to ideal solution
Siplus=zeros(1,4);
Siminus=zeros(1,4);%vectors to store each Si value
tally1=0;
tally2=0;
for i=1:4
    for j=1:4
        %Fixed TOPSIS
    tally1=tally1+(D(i,j)-Dplus(j))^2; %sum of squared diff
    tally2=tally2+(D(i,j)-Dminus(j))^2; %sum of products (22)
    end
    Siplus(i)=sqrt(tally1);
    Siminus(i)=sqrt(tally2);
    tally1=0;
    tally2=0;
```

```
end

% calculate the score of each RAT
SC=zeros(1,4); %array to store each RATs score
for i=1:4
    SC(i)=Siminus(i)/(Siminus(i)+Siplus(i));
end
%SC
%The best RAT is the one with the highest score
Best_RAT=0;
for i=1:4
score=SC(i);
if score==max(SC)
    Best_RAT=i;
end
end

%EXPERIMENT 3.3:Determining number of unnecessary Handovers
%Assumed current_RAT will be directly mapped to the index position of the
%corresponding RAT score in the score array
%i.e. SC(current_RAT)=score for the current RAT
if (iterations==1) %if first selection event assume MMT was not connected
initially so select best RAT
    last_RAT=current_RAT;
    current_RAT=Best_RAT;
    t2=t1;
    t1=1;
%not the first iteration/selection event
else
    if(current_RAT==Best_RAT || (current_RAT~=Best_RAT &&
(max(SC)/SC(current_RAT)<=threshold)))
        Best_RAT=current_RAT; %maintain current connection if below threshold


    else %if meets handover condition
        if(last_RAT~=0 && last_RAT==Best_RAT && (t1-t2)==2) %if last connected
network same as best RAT in current iteration
            unecessary_handovers=unecessary_handovers+1;
        end
        No_of_handovers=No_of_handovers+1;
        last_RAT=current_RAT;
        current_RAT=Best_RAT;
        t2=t1;
        t1=iterations;
    end

end

end

%observe if RAT changes
%disp(["new RAT is:",num2str(current_RAT)]);
%disp(["last RAT is:",num2str(last_RAT)]);
%disp(["best RAT is:",num2str(Best_RAT)]);
%end of a single selection event
No_of_handovers %show number of handovers
unecessary_handovers
pecentage_handovers=(unecessary_handovers/No_of_handovers)*100
%plotting the results
```

```
categories = {'total', 'unnecessary', '%'};
values = [No_of_handovers, unnecessary_handovers, pecentage_handovers];
bar(categories, values);
xlabel('Handovers');
ylabel('Number of handovers and percentages');
title('number of unnecessary handovers for 1000 users where \sigma=1.3');

%plotting data
ExecutionNumber=[1,2,3,4,5,6,7,8,9,10]; %execution no
percentageHo=[5.7143,2,9.0909,13.9535,4.7619,7.8947,4.5455,2.0408,2.4390,8.3333];
% %of unnecessary HO
plot(ExecutionNumber, percentageHo)
xlabel('Execution number');
ylabel('% of unnecessary handovers');
title('percentage of unnecessary handovers against execution number');
grid on;
%NB updated utility functions to include cost
function u = f_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.25;b=48;
    u=1/(1.0+e^(-a*(x-b)));
    case "video"
    a=0.003;b=2000;
    u=1/(1.0+e^(-a*(x-b)));
    otherwise %web browsing
    a=0.01;b=564;
    u=1/(1.0+e^(-a*(x-b)));

end

end
function u = g_x(service,x)
e=exp(1);
switch service
    case "voice"
    a=0.1;b=75;
    u=1-(1/(1.0+e^(-a*(x-b))));
    case "video"
    a=0.1;b=112.5;
    u=1-(1/(1.0+e^(-a*(x-b))));
    otherwise %web browsing
    a=0.03;b=375;
    u=1-(1/(1.0+e^(-a*(x-b))));

end

end
function u = h_x(service,x)
switch service
    case "voice"
    g=1/30.0;
    u=1-(g*x);
    case "video"
    g=1/30.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/30.0;
```

```
    u=1-(g*x);

end
end
%update
function u = h_x2(service,x)
switch service
    case "voice"
    g=1/50.0;
    u=1-(g*x);
    case "video"
    g=1/50.0;
    u=1-(g*x);
    otherwise %web browsing
    g=1/50.0;
    u=1-(g*x);

end

end

%function to assign weights
function w = allocateWeights(N) %N = number of users you want
%weights=zeros(N,3); %w is a vector of 3 weight vectors for each user
%assume each element of the cell corresponds to a different service
%assume w{1}=voice weights for user i
w=cell(1,N); %creates 1xN cell where each row = 1, column= N 1x3 sets of user
weights for user i
%basically a 1D array of 1x3 columns
the_weights=[1,2,3,4,5,6,7,8,9];
for i =1:length(w) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=randi([weights(1),weights(9)]);
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
    end
    w{i}=users_i_Weights; %each user-i will have 3 sets of randomized weights
end

end
function Norm = Normalize(userCell) %returns normalized weights for a cell of user
weight vectors
userCellSize=size(userCell);
Norm=cell(1,userCellSize(2)); %should be the same size as input cell
for i =1:length(Norm) %for N users
    for j = 1:3 % loop through weight vector for each service (x3 per user)
            arr=zeros(1,4);
            for k=1:4 %loop through weight for each criterion (x4 per weight
vector)
            arr(k)=userCell{i}{j}(k)/sum(userCell{i}{j});
            end
            users_i_Weights{j}=arr; %assign each completed vector to set of
vectors
    end
    Norm{i}=users_i_Weights; %each user-i will have 3 sets of normalized weights
```

```
end

end

function T = find_gain(Umat,alpha,beta,gamma,WU,WS,WO)
%a,b,y are varied to determine the best gain for our model
%using (30) from [7]
Wvec=(alpha*WU)+(beta*WO)+(gamma*WS);
gain=zeros(1,4);
total=0;
for i=1:4
    for j=1:4
        total=total+(Umat(i,j)*Wvec(j));
    end
    gain(i)=total;
    total=0;
end
T=sum(gain);

end
```

9.1.8 Appendix H: Ethics outcome letter

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

PRE-SCREENING QUESTIONNAIRE OUTCOME LETTER

STU-EBE-2023-PSQ000520

2023/08/05

Dear Bina Mukuyamba,

Your Ethics pre-screening questionnaire (PSQ) has been evaluated by your departmental ethics representative. Based on the information supplied in your PSQ, it has been determined that you do not need to make a full ethics application for the research project in question.

You may proceed with your research project titled:

RAT Selection Algorithm for Multiservice Multimode Terminals in Next Generation Wireless Networks

Please note that should aspect(s) of your current project change, you should submit a new PSQ in order to determine whether the changed aspects increase the ethical risks of your project. It may be the case that project changes could require a full ethics application and review process.

Regards,

Faculty Research Ethics Committee

10. EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer (Zulpha.Geyer@uct.ac.za; Chem Eng Building, Ph 021 650 4791).Students must include a copy of the completed form with the final year project when it is submitted for examination.

Name of Principal Researcher/Student: Bina Mukuyamba **Department:** ELECTRICAL ENGINEERING

If a Student: YES **Degree:** BSc Electrical & computer engineering **Supervisor:** Olabisi Falowo

If a Research Contract indicate source of funding/sponsorship: N/A

Research Project Title: RAT Selection Algorithm for Multiservice Multimode Terminals in Next Generation Wireless Networks

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	NO
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	YES	NO
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.	YES	NO
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	NO

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate.

I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

	Full name and signature	**Date**
Principal Researcher/Student:	**Bina Eric Mukuyamba**	30 October 2023

This application is approved by:

Supervisor (if applicable):	**Olabisi Falowo**	30 October 2023
HOD (or delegated nominee): Final authority for all assessments with NO to all questions and for all undergraduate research.	**Janine Buxey**	30 October 2023
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		

ADDENDUM 1:

Please append a copy of the research proposal here, as well as any interview schedules or questionnaires:

Student proposed?	*Y/N* **N**	*If Y, student name*
ID:	**OF-7**	
SUPERVISOR:	Olabisi Falowo	
TITLE:	RAT Selection Algorithm for Multiservice Multimode Terminals in Next Generation Networks	
DESCRIPTION:	Most multimode terminals simultaneously support multiple services (such as voice, video streaming, and web browsing). However, most of the network selection algorithms proposed for vertical handoff in the literature are only suitable for mobile terminals running a single service. It is possible for multiples services from a multimode terminal to be handed over to a single or multiple RATs. The objective of this project is to develop a network selection algorithm for multiservice multimode terminals in heterogeneous wireless networks. The algorithm will consider user preferences, network attributes, and service characteristics in making vertical handoff decisions. These attributes will be combined to make vertical handoff decisions using the fuzzy analytic hierarchy process.	
DELIVERABLES:	A review of vertical handoff algorithms, implemented vertical handoff algorithm, simulation results, and report.	
SKILLS/REQUIREMENTS:	MATLAB or any other programming language, EEE4121F.	
GA1: Problem solving: *Identify, formulate, analyse and solve complex* engineering problems creatively and innovatively*	The student is expected to design and implement an algorithm for making vertical handover decisions for multiple services in heterogeneous wireless networks.	
GA 4**: Investigations, experiments and analysis: *Demonstrate competence to design and conduct investigations and experiments.*	The student is expected to investigate the performance of RAT selection algorithm.	
EXTRA INFORMATION:	For a student interested in pursuing a master's degree, the project can be expanded to an MSc dissertation.	
BROAD Research Area:	Wireless Networks	
ETHICS	The project does not require the use of human subject or animal.	
Project suitable for ME/ ECE/EE/All?	EE/ECE students who have taken EEE4121F course.	

ADDENDUM 2: To be completed if you answered YES to Question 2:

It is assumed that you have read the UCT Code for Research involving Human Subjects (available at http://web.uct.ac.za/depts/educate/download/uctcodeforresearchinvolvinghumansubjects.pdf) in order to be able to answer the questions in this addendum.

2.1 Does the research discriminate against participation by individuals, or differentiate between participants, on the grounds of gender, race or ethnic group, age range, religion, income, handicap, illness or any similar classification?	YES	NO
2.2 Does the research require the participation of socially or physically vulnerable people (children, aged, disabled, etc) or legally restricted groups?	YES	NO
2.3 Will you not be able to secure the informed consent of all participants in the research? (In the case of children, will you not be able to obtain the consent of their guardians or parents?)	YES	NO
2.4 Will any confidential data be collected or will identifiable records of individuals be kept?	YES	NO
2.5 In reporting on this research is there any possibility that you will not be able to keep the identities of the individuals involved anonymous?	YES	NO
2.6 Are there any foreseeable risks of physical, psychological or social harm to participants that might occur in the course of the research?	YES	NO
2.7 Does the research include making payments or giving gifts to any participants?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

ADDENDUM 3: To be completed if you answered YES to Question 3:

3.1 Is the community expected to make decisions for, during or based on the research?	YES	NO
3.2 At the end of the research will any economic or social process be terminated or left unsupported, or equipment or facilities used in the research be recovered from the participants or community?	YES	NO
3.3 Will any service be provided at a level below the generally accepted standards?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

ADDENDUM 4: To be completed if you answered YES to Question 4

4.1 Is there any existing or potential conflict of interest between a research sponsor, academic supervisor, other researchers or participants?	YES	NO
4.2 Will information that reveals the identity of participants be supplied to a research sponsor, other than with the permission of the individuals?	YES	NO
4.3 Does the proposed research potentially conflict with the research of any other individual or group within the University?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues: