A User Facilitated Autonomous Load Balancing Framework for UCN

vorgelegt von
Mürsel Yildiz, M.Sc.
aus Elazig - Türkei

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung der akademischen Grades:
Doktor der Ingenieurwissenschaften
Dr.-Ing.
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. rer. nat. Volker Markl
Gutachter: Prof. Dr.-Ing. habil. Dr. h. c. Sahin Albayrak
Gutachter: Prof. Dr. Alessandro Bogliolo
Gutachter: Prof. Dr. Gözde Bozdağ Akar

Tag der wissenschaftlichen Aussprache: 25. November 2013

Berlin, 2013
D 83
Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt.

__________________________  __________________________
Ort, Datum                      Mürsel Yildiz
Abstract

The Internet providers and end-users are attracted by the particular conceptualization of user centric networking (UCN), where the end-user, as an Internet stakeholder, plays the micro network provider role aside from being the consumer [1]. An attractive candidate technology for the deployment of this fresh approach is the IEEE 802.11 standard. The worldwide availability, efficiency and low-cost for broadband access technologies are the main attractive features of the WLAN, which renders this novelty feasible. The main motivation of participating the end-user to the network infrastructure, not only as a consumer but also as a producer of the content, is to expand the coverage for Internet access by means of wireless fidelity (Wi-Fi) [2]. The deployment of this wireless social community-like approach necessitates additional improvements in network functionalities for feasibility and community persistency. Certain core enabling factors of the UCN are i) cooperation incentivized bandwidth sharing of subscribed broadband access, ii) security and trust management, and finally iii) the optimization in the resource allocation aspects.

Participation of the end-user to the Internet as a $\mu$-provider attracts the attention of Internet providers, and the researchers as well, due to new open issues for this immensely dynamic telecommunications landscape. One of these open issues is the question how one can achieve a homogenous quality of service (QoS) distribution among UCN-providers. It should be underlined that ensuring a certain level of QoS plays a significant role for the persistence of the UCN environment as congestion regions would demotivate the UCN-stakeholders. Such unsolicited states can be eliminated by means of load balancing among UCN providers. The first difficulty, however, in realizing this objective is the variety of user entities in terms of hardware or software capabilities, which participate to the UCN with a provider role. Additionally the dynamic characteristics of UCN domains, the necessity of network deployment with an autonomic manner, fairness in terms of gained incentive units and the selfishness of UCN providers have great impacts on the realization of a balanced load.

The basic emphasis of this thesis is to develop an end user facilitated autonomous
load balancing framework for the UCN. To this end, different load balancing dynamics are covered and validated by means of real implementation and testbed experiments. We relax our approaches mainly to two classes of decision making perspectives, namely, the end-user side and the provider-side. The end-user side is the dynamic for the facilitation of the load balancing in UCN. The decision is restricted to the access point (AP) selection stage, then after which the end-user is loyal to the decisions driven by the provider. The proposed AP selection method is a network performance history based approach. To this end we answer the question, how we can equip the end-user entities with the ability of characterizing UCN subnetworks in terms of the network performances and selecting more suitable ones. For the characterization ability we also operate the UCN-clients with an active role by means of performing network performance related measurements and even forwarding them to the providers. The forwarded measurements are then utilized by the providers for the provider side decision-making. Additionally we extend our answer with the functionality of taking the short-term provider operation into account while selecting a specific UCN subnetwork. The short-term operation of providers stem mainly from either the mobility behavior of the stakeholders or the user preferences. Finally on the end-user side, we discuss the mobility awareness while selecting the UCN-provider.

On the provider side, we initially introduce our approach on how the load balancing can be performed among providers under the presence of selfish UCN nodes. Our solution issues the distributiveness, fairness and selfishness of the stakeholders. To this end, we introduce our QoS-based crediting method as an incentive mechanism, which creates driving forces for the load balancing. That is to say, the selfish APs are motivated for sharing their load with available APs based on the introduced principle that an AP may gain more credits while serving in an underloaded state, i.e. with better QoS. These decisions are taken in a distributed manner by means of collaborations among providers. The fairness is realized by the natural dynamic of the proposed QoS based crediting mechanism. This principle addresses the diversity of providers in terms of their hardware or software capabilities. In other words the load is balanced in a manner that the better equipped providers serve more clients than the ones serving with worse QoS. We additionally took the provider owners’ preferences into account while preventing excessive resource consumption from the associated UCN-clients and hence risking the owners’ network experience. Finally, we discuss on an exceptional case, i.e. the environmental conditions, where the collocutor APs are not capable of balancing the load in a collaborative way and hence the decision should be taken in a central manner.
A higher level mechanism on the provider side is introduced for the autonomic behavior in load balancing. We initially focus on a more comprehensive model for the autonomous behavior in resource allocation and strategy improvement in wireless networks. This model has mainly two motivations. The first motivation of the proposed model is the improvement for the elimination of redundant resource utilization for specific network objectives while reducing the network response time. The second motivation is to introduce the ability to the network for matching network strategies with different environmental conditions in terms of impact rates on the environment, i.e. the success rates of the taken strategies. Having validated this framework, we adjust the model parameters in order to focus on a single network objective, i.e. the load balancing. We define different characteristics of environments and equip the providers with different strategies for the load balancing. We validate our approach via real implementation and observe how the strategies of the providers change with the changing environmental conditions. The experiment results draw our attention to the significance of the network adaptation capability.

Finally we comment on our test environment and discuss the implementation aspects. We introduce the user centric wireless test-bed, which interprets the user as a key component of the network control and operation. Throughout the thesis, various approaches are studied for the load balancing problem stipulating frequent collaborations among members in the UCN. Thus we additionally state the risk for the UCN persistency due to the excessive resource consumptions by UCN-clients, which may arise due to frequent collaborations. In order to handle this problem, we introduce our methodology for limiting the resource utilization to a certain extent for the UCN environment in an efficient and sufficient way. We present our $\mu$-Kernel based OS virtualization technique for the providers, which limits excessive resource utilization by others while decreasing the complexities in decision-making.
Abstrakt


großen Einfluss auf die Erreichung einer ausgewogenen Netzbelastung.


Schließlich wird die Testumgebung eingeführt und die verschiedenen Aspekte der Implementierung, bzw. die Umsetzbarkeit diskutiert. Der benutzerzentrische Wireless Prüfstand, der den Benutzer als eine wichtige Komponente der Netzwerksteuerung und -bedienung interpretiert, wird eingeführt. In dieser Dissertation werden verschiedene Ansätze für das Lastenausgleichsproblem untersucht, die häufige Kooperationen zwischen den Akteuren in der UCN vorsehen. Deshalb haben wir zusätzlich das Risiko für die UCN Dauerhaftigkeit angegeben, das aufgrund des übermäßigen Ressourcenverbrauchs (wegen häufigen Kooperationen) entstehen kann. Um mit diesem Problem umzugehen, führen wir eine Methodik zur Begrenzung der Ressourcennutzung ein. Wir präsentieren unsere µ-Kernel basierende OS-Virtualisierung Technik für die Anbieter, die eine übermäßige Ressourcenauslastung begrenzt. Mit dieser Herangehensweise zielen wir auf die Verringerung der Komplexität bei der Entscheidungsfindung ab.
Acknowledgements

I would like to express my deepest thanks to my advisor Professor Şahin Albayrak for his guidance and providing me a perfect research environment. I have learned a lot from his comments and the way he handles the difficulties. I do believe that I still have a lot to be learnt from him...

My deepest thanks and warmest feelings go to my family for their guidance and support on my learning adventure throughout my life. I still remember my Father’s words on my aspiration in science and engineering, which I believe worth to be mentioned here: "...Science is a long-term marathon, one can not pass it by chance, on the contrary it requires endeavoring regularly, with patience and a well-disciplined manner...". I had to remember these words in each day during my studies. Additionally, I can not deny how much it is worth to me having continuous and tireless support of my Mother during my challenging days. Without her encourage, I believe that I definitely could not have the chance to complete this thesis.

Finally, many thanks go to my DAI Labor colleagues for their supportive discussions and fruitful comments, which I interpret as immensely valuable feedbacks.

Berlin, December 4, 2013

The research leading to these results has received funding from the EU IST Seventh Framework Programm ([FP7/2007- 2013]) under grant agreement number 257418, project ULOOP User-centric Wireless Local Loop.
# Contents

1 Introduction 3  
   1.1 Briefly the UCN Dynamics and Relevant Issues . . . . . . . . . . . . 4  
      1.1.1 Core Functionalities and Related Descriptions . . . . . . . . 6  
      1.1.2 Challenges from the Load Balancing Perspective . . . . . . . 9  
   1.2 Our Load Balancing Approach . . . . . . . . . . . . . . . . . . . . 10  
   1.3 Contributions and Thesis Organization . . . . . . . . . . . . . . . 11  
      1.3.1 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . 11  
      1.3.2 Thesis Organization . . . . . . . . . . . . . . . . . . . . . . 15  

2 End User Cooperation for the Facilitation of Load Balancing in UCN 19  
   2.1 AP Selection in UCN: A Network Performance History Based Approach 19  
      2.1.1 State of the Art . . . . . . . . . . . . . . . . . . . . . . . . 20  
      2.1.2 Evaluation of the Literature from the UCN Perspective . . . 23  
      2.1.3 Our Approach . . . . . . . . . . . . . . . . . . . . . . . . . 24  
      2.1.4 Technical Details of the Proposed Model . . . . . . . . . . . . 25  
      2.1.5 Experiments and Test Results . . . . . . . . . . . . . . . . . 43  
      2.1.6 The Evaluation of the Proposed AP Selection Framework . . . 48  
   2.2 Relative Transiency Problem and Mobility Behavior Modeling for  
      UCN-clients . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50  
      2.2.1 State of the Art . . . . . . . . . . . . . . . . . . . . . . . . 51  
      2.2.2 Evaluation of the Literature from the UCN Perspective . . . 52  
      2.2.3 Our Approach . . . . . . . . . . . . . . . . . . . . . . . . . 53  
      2.2.4 Technical Details of the Proposed Model . . . . . . . . . . . . 55  
      2.2.5 Experiments and Test Results . . . . . . . . . . . . . . . . . 61  
      2.2.6 Evaluation of the Proposed Framework . . . . . . . . . . . . 64  
   2.3 Relative Transiency and Mobility Awareness for the AP Selection  
      Problem in UCN . . . . . . . . . . . . . . . . . . . . . . . . . . . 66  
   2.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69  

xiii
5 Test Environment and Implementation Aspects 159
  5.1 User Centric Wireless Testbed 159
    5.1.1 Literature Review on Wireless Testbeds 160
    5.1.2 Evaluation of the Literature from the UCN Perspective and Our Approach 162
    5.1.3 Testbed Architecture 163
    5.1.4 Testbed Components 164
    5.1.5 Sample Experiment and Results 167
  5.2 Microkernel Based OS Virtualization as A Trusted Cooperation Platform for UCN-APs 168
    5.2.1 Technical Details 171
    5.2.2 Exploring Network Performance of the Proposed Solution 172
  5.3 Conclusion 174

6 Conclusions 175
List of Figures

1.1 ULOOP Project Overview ........................................ 5
1.2 An Overview of the UCN Communication Framework .......... 9
1.3 An Overview of the Contributions in this Thesis ............ 12

2.1 Timing Diagram on Client Entities for the Network Performance Tests 26
2.2 Transitions between States of a Markov Chain ................ 26
2.3 Objective Function and Component Surfaces ................... 30
2.4 Critical Points of Objective Function for Constant Queues ........ 30
2.5 EM Algorithm Flow Chart ....................................... 33
2.6 Experiment Results on the Proposed EM Algorithm ............ 37
2.7 Transitions between Belief States based on pdf Parameters .... 40
2.8 Steady-State Probability Distribution for ”Unattractive” State with Changing Variance and Transition Parameter .......... 42
2.9 Test Bed .......................................................... 44
2.10 Estimations on the pdf of Time Durations for Different Performance Indicators ................................................. 46
2.11 Steady State Probabilities of Critical States and AP Attractiveness based on different TC Percentages ....................... 47
2.12 UCN-APs attractivenesses among UCN-Clients ................. 48
2.13 UCN-APs attractivenesses for a specific UCN-Client preferring voice traffic .................................................... 49
2.14 Encountered Access Points and Signal Levels on the corresponding Route ......................................................... 54
2.15 Access Point Groups and UCN-Node Mobility .................. 55
2.16 Three State Appearance Frequency with State Parameters ..... 56
2.17 Three State Appearance Duration with State Parameters ..... 58
2.18 Mobility Behavior Learning and Prediction ..................... 62
2.19 AP Group Formation ............................................... 63
2.20 Next AP Group Prediction ......................................... 63
2.21 Transiency Prediction Results .................................... 64
List of Figures

2.22 Example Model Objects ........................................... 64
2.23 Regular Chain Formation and Intersection Points .................. 65
2.24 Mobility Awareness in AP Selection Decision ...................... 67
2.25 Flow Chart for the Mobility Aware AP Selection Improvement ..... 68
2.26 A Summary for the Proposed AP Selection Model ................. 69
3.1 An MDP Model Representation in the Literature [3] ............... 76
3.2 A POMDP Model Representation in the Literature [3] ............. 78
3.3 Figure representing the topology and last-hop performance measurements ........................................... 85
3.4 Sigmoidal Cost Function Component .................................. 86
3.5 Estimation states of different communities .......................... 87
3.6 Arrivals of Cooperation Request ...................................... 88
3.7 POMDP policies of stakeholders .................................... 95
3.8 UCN-AP load balancing software components ................. 99
3.9 UCN-client side software components ............................. 100
3.10 Evaluation outcome ............................................... 101
3.11 A Load Balancing Scenario by means of broadcasted bid exchange 102
3.12 A Load Balancing Scenario by means of multiple interactions among APs ........................................... 103
3.13 Partial Observation of Manageable UCN-AP ....................... 105
3.14 State Transition Probabilities for No Action ...................... 108
3.15 State Transition Probabilities for Attack Response .............. 108
3.16 State Transition Probabilities for Load Balancing ............... 109
3.18 System Block Diagram ............................................ 112
3.19 Testbed Block Diagram ............................................ 113
3.20 Anomaly Scenario User Experience w.r.t Time .................... 114
3.21 Anomaly Scenario Observed Data Rate w.r.t Time ............... 114
3.22 Congestion Scenario User Experiences w.r.t Time ............... 115
3.23 Congestion Scenario Observed Data Rate w.r.t Time ............ 115
3.24 Critical System Failure Scenario User Experiences w.r.t Time . 116
3.25 Critical System Failure Scenario Observed Data Rate w.r.t Time 116
4.1 Resource allocation based on the environmental conditions ....... 125
4.2 Sample Network Ontology ........................................... 127
4.3 State Transition Diagram of the Environment ...................... 128
4.4 An example count of arrival rate vs time .......................... 130
List of Figures

4.5 Positions of the Proposed Mechanisms in the Generic Control Loop . 132
4.6 Basic Control Loop States............................................. 133
4.7 Access Point Block Diagram ........................................ 139
4.8 Organizational Threads’ Flow Charts ................................. 140
4.9 Example Information Models (IM) for an Observation and Action . 141
4.10 Dynamic Resource Allocation Thread Flow Charts .................. 142
4.11 Object Prioritization based on Event Arrivals ..................... 144
4.12 Success Rates and Instances of Network Actions vs. System Time . 144
4.13 Information Model for the Network Ontology ........................ 146
4.14 Object Prioritization based on Event Arrivals and Client Experience 147
4.15 Success Rates and Instances of QoS Objective Actions vs. System Time 147
4.16 Object Prioritization based on Event Arrivals and Client Experience 148
4.17 Success Rates and Instances of Network Actions vs. System Time . 148
4.18 Load Balancing Policy in Ontological Form ........................ 152
4.19 Experiment Results: Impact Rates of Various Algorithms in a very
dynamic UCN Environment ............................................. 155

5.1 Testbed Architecture .................................................. 164
5.2 Alix Board .............................................................. 165
5.3 NEO-Industrial PC IPC Embedded Computer ......................... 166
5.4 Borrower User QoE over time. ....................................... 168
5.5 Donor Throughput over time. ......................................... 169
5.6 End User Collaborations in Different Telecommunication Landscapes 170
5.7 A UCN Scenario ........................................................ 171
5.8 Access Point Entity with Virtualization Solution .................... 172
5.9 Network Performance Comparison of Trusted Cooperation Platform
with a Full-featured AP ................................................. 173
List of Tables

2.1 Summary of Parameter Set for the Active Measurement Queue . . . 28
2.2 Dominant Critical Observation vs. Related Traffic Categories . . . . 40

3.1 The state-transition probability function for requestee POMDP model 95
3.2 The observation probability function for requestee POMDP model . 96
3.3 The reward function for requestee POMDP model . . . . . . . . . 97
3.4 The POMDP based policy for the requestee strategy . . . . . . . . 97
3.5 The state-transition probability function for requester POMDP model 97
3.6 The observation probability function for the requester . . . . . . . . 98
3.7 The reward function for requester POMDP model . . . . . . . . . 98
3.8 The POMDP based policy for the requester strategy . . . . . . . . 98
3.9 The POMDP based policy for controller UCN-AP . . . . . . . . . 110

4.1 Summary of Parameter Set for the Control Loop Dynamics . . . . 134
## Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
<tr>
<td>AAC</td>
<td>Available Admission Capacity</td>
</tr>
<tr>
<td>AC</td>
<td>Admission Control</td>
</tr>
<tr>
<td>AMD</td>
<td>Advanced Micro Devices</td>
</tr>
<tr>
<td>AME</td>
<td>Autonomic Management Element</td>
</tr>
<tr>
<td>ANA</td>
<td>Autonomic Network Architecture</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>ARASO</td>
<td>Autonomous Resource Allocation and Strategy Optimization</td>
</tr>
<tr>
<td>BMAP</td>
<td>Batched Markov Arrival Processes</td>
</tr>
<tr>
<td>BSSID</td>
<td>Basic Service Set Identification</td>
</tr>
<tr>
<td>BSS</td>
<td>Basic Service Set</td>
</tr>
<tr>
<td>CAC</td>
<td>Call Admission Control</td>
</tr>
<tr>
<td>CAPEX</td>
<td>Capital expenditures</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CTMC</td>
<td>Continuos Time Markov Chain</td>
</tr>
<tr>
<td>CTMP</td>
<td>Continuos Time Markov Processes</td>
</tr>
<tr>
<td>CUSUM</td>
<td>Cumulative Sum</td>
</tr>
<tr>
<td>D-ITG</td>
<td>Distributed Internet Traffic Generator</td>
</tr>
<tr>
<td>DAI</td>
<td>Distributed Artificial Intelligence</td>
</tr>
<tr>
<td>DC-SNM</td>
<td>Distributed Cognitive cycle for System and Network Management</td>
</tr>
</tbody>
</table>
List of Tables

DDoS  Distributed Denial of Service

DME  Decision Making Engine

DoS  Denial of Service

E-UTRAN  Evolved Universal Terrestrial Radio Access Network

EFIPSANS  Exposing The Features In IP Version Six Protocols that can be Exploited/Extended for the Purposes of Designing/Building Autonomic Networks and Services

EM  Expectation Maximization

ETSI  European Telecommunications Standards Institute

EU  European Union

FI  Future Internet

FOCALE  Foundation Observation Comparison Action Learn Reason

FP7  Seventh Framework Programme

GANA  Generic Autonomous Network Architecture

GPS  Global Positioning System

GSM  Global System for Mobile Communications

HDD  Hard disk drive

HRFA  High Rate First Association

IBM  International Business Machines

ICT  Information and Communication Technologies

ID  Identity

IEEE  Institute of Electrical and Electronics Engineers

IMS  IP Multimedia Subsystem

IPC  Inter Process Communication

IP  Internet Protocol

xxiv
List of Tables

**JGCAC** Joint Group Call Admission Control

**LAN** Local Area Network

**MAC** Medium Access Control

**MANET** Mobile Ad Hoc Network

**MAP** Markov Arrival Processes

**MAS** Multi-Agent System

**MB** Mega Byte

**MC** Markov Chain

**MDP** Markov Decision Processes

**ME** Managed Element

**MHB** Mobility History Base

**miniPCI** mini Peripheral Component Interconnect

**MIT** Massachusetts Institute of Technology

**MMAP** Marked Markov Arrival Processes

**MMPP** Markov Modulated Poisson Process

**NA** Network Architecture

**NDCM** Network Domain Cognitive Manager

**NECM** Network Element Cognitive Manager

**ORBIT** Open-Access Research Testbed for Next-Generation Wireless Networks

**OS** Operating System

**PCI** Peripheral Component Interconnect

**PCMCIA** Personal Computer Memory Card International Association

**PC** Personal Computer

**pdf** Probability Distribution Function

**POMDP** Partially Observable Markov Decision Processes
List of Tables

QoE Quality of Experience
QoS Quality of Service
RoQ Reduction of Quality
RSSI Received Signal Strength Indicator
SBO Send Both Oriented State
SDRAM Synchronous dynamic random-access memory
Self-Net Self-Management of Cognitive Future InterNET Elements
SLO Send Load Oriented State
SNMP Simple Network Management Protocol
SNR Signal-to-Noise Ratio
SOCRATES Self-Optimisation and self-ConfiguRATion in wirelEss networkS
SON Self Organizing Networks
SRO Send Revenue Oriented State
STA Station
SUT System Under Test
TCP Transmission Control Protocol
UCN-AP User Centric Networking Access Point
UCN User Centric Networking
UCW Testbed User Centric Wireless Testbed
UE User Entity
ULOOP User Centric Wireless Local Loop
URC Unit Resource Cost
VoIP Voice Over IP
WAP Wireless Access Point
WART Wide-Area Radio Testbed

xxvi
Wi-Fi  Wireless Fidelity

WIMNET  Wireless Infrastructure Mesh Network

WLAN  Wireless Local Area Network

WSN  Wireless Sensor Network

WS  Wireless Station

XML  Extensible Markup Language
1 Introduction

The last decade has been a true success story, where rapid innovations in hardware and software technologies raise the tendency for a specific novelty, i.e. all-in-one attitude. This approach, in turn, opens the door for the smart device era. The elegant features of smart devices, such as portability, computation power and the high performant networking capabilities, enable the participation of these devices to the collaboration based applications. Although this facility entail the long term availability and low cost interoperability, it draws network providers’ attention to the utilization of end-user device capacity for network control and optimization related activities. These advances in telecommunications literature prepare a convenient ambiance for the concept of end-user centricity, which brings about the opportunity for the end-user to play an active role rather than a passive one.

The evolution of the Internet towards multidisciplinary services and applications improves the significance of end-user, i.e the client, by means of various business models. In parallel to this progress, the user centricity becomes a comprehensive notion for various networking scenarios and research fields. Alongside the role of the end-user in control or management related decision making, many studies focus on interpreting the well-equipped end-user device as a key component of the Network. For instance, empowering the end-user as a new network entity with a provider role, is a promising approach as an opportunity to extend the coverage of the Internet access.

The Internet providers and end-users are attracted by the particular conceptualization of user centric networking (UCN), where the end-user, as an Internet stakeholder, plays the micro network provider role aside from being the consumer [1]. An attractive candidate technology for the deployment of this fresh approach is the IEEE Wireless LAN (WLAN) standard. The worldwide availability, efficiency and low-cost for broadband access technologies are the main attractive features of the WLAN, which renders this novelty feasible. The main motivation of participating the end-user to the network infrastructure, not only as a consumer but also as a producer of the content, is to expand the coverage for Internet access by means of wireless fidelity (Wi-Fi) [2]. The deployment of this wireless social community-like
1 Introduction

One of the main dynamics for the UCN formation is the necessity of the organization among stakeholders in an autonomous manner, which raises the complexity in the previously mentioned core UCN enabling factors. Considering the resource allocation aspects, for instance, one would agree on the fact that the attractiveness, and hence the persistency of the UCN is highly correlated with the degree of optimization in resource allocation. This is mainly due to the nature of the UCN telecommunication landscape, where i) members with the client role (UCN-client) prefer a service quality over an acceptable threshold, ii) members with provider role (UCN-provider) are allowed to join to the network with resource limited devices and finally iii) the coalescence of UCN-providers and clients are fully stochastic. Additionally, the necessity of the autonomous deployment in highly dynamic UCN environments increases the risks for the potential formation of congestion regions among UCN subnetworks. Such congestion regions, in turn, bring about extensive fluctuations in the network quality in UCN environment. This can be eliminated by means of load balancing among UCN-providers, which helps for distributing the network quality in a homogenous manner. In this thesis, we focus on this specific branch of the resource allocation aspects, i.e. the load balancing problem in UCN.

In the proceeding section, we briefly introduce the UCN and relevant issues in order to provide a better understanding of the content and the motivation of this thesis (Section 1.1). Based on the introduced dynamics of UCN, we issue various challenges from the load balancing perspective (Section 1.1.2). In Section 1.2, we introduce our load balancing approach as a solution to the addressed challenges. Finally we comment on the contributions of this thesis and provide the thesis organization in Section 1.3.

1.1 Briefly the UCN Dynamics and Relevant Issues

The UCN is a highly stochastic and dynamic environment due to fully uncertain coalescences of various stakeholders. The main focus remains in the autonomic deployment at local-loops by means of feasible business models, adequate incentives and various network core functionalities’ evolution. To this end, the user centric wireless local loop (ULOOP) [2], an EU FP7 project funded within ICT Call 5,
1.1 Briefly the UCN Dynamics and Relevant Issues

is proposed for the research, analysis and conceptualization of the corresponding enabler dynamics. The main motivation of this large scale attempt is to explore the user provided networks and come up with suitable application frameworks, user-centric business models, and low cost wireless local-loop architectures [4]. Apart from the research oriented ULOOP project, different wireless network communities have already been proposed in the industry [5, 6, 7].

Before we introduce the UCN dynamics, we firstly describe different roles of UCN stakeholders. These stakeholders are composed of UCN members, which we refer as UCN-nodes throughout the thesis. The UCN-nodes may play mainly two different roles, namely, the UCN-client and UCN-access point (UCN-AP). A UCN-node is allowed to maintain different roles from time to time. Finally a UCN-community is a set of UCN-nodes, sharing similar interests.

![Figure 1.1: ULOOP Project Overview](image)

An overview of the ULOOP functionalities and the boundaries are provided in Figure 1.1, which is taken from [8]. The basic principle of this telecommunication landscape is to form a community on Wi-Fi to expand the subscribed broadband access bandwidth, where end-users share the Internet access in exchange for incentive units. Each UCN stakeholder has a personal identity to be utilized while participating to the UCN community with various Wi-Fi capable devices either as a UCN-AP
1 Introduction

or a UCN-client. Once a UCN-client associates with a UCN-AP, an additional negotiation phase starts for the trust and incentive management. The negotiation parameters can be, however not restricted to, the distributed trust metrics of stakeholders, expected resource utilizations and unit resource costs. A fulfillment of the reciprocally guaranteed negotiation parameters has an impact on the trust and reputation metrics of both stakeholders. During the network operation, the UCN-APs are in collaboration for the self-organization functionalities. Besides, there exists no restriction for the participation of mobile UCN-APs or the UCN members with the motivation of short-term transient AP functionalities.

In this section, a very brief introduction is provided for the UCN functionalities referring to ULOOP project. More detailed information can be found in the references [9, 10, 11, 12]. It should be highlighted that certain UCN dynamics are later revisited in this thesis in order to take the readers attention to the specific relevant issues. Next, we briefly comment on the core functionalities and related descriptions.

1.1.1 Core Functionalities and Related Descriptions

For the realization of the UCN community, referring to the Figure 1.1, the following core functionalities and descriptions are the point of interest for the UCN research community.

1.1.1.1 Cooperation Incentives and Trust Management

Providing cooperation incentives to participating users is one of the core enabling components of UCN, which should be complemented by trust management and crediting mechanisms. Fairness and the adequateness of the cooperation incentives for the motivation of UCN members is a hot topic for the UCN research community. This functionality is critically significant for the continuity of this wireless community.

The roaming habits and the end user’s willingness to share constructs the main framework for the cooperation incentives mechanism in UCN. This principle boosts the coalescences of UCN communities, i.e. the cooperative network models. This is due to various motivations such as willingness to share multimedia data, having the advantage of utilizing various network services, low-cost Internet access, etc. The bilateral interaction among UCN nodes necessitates the cooperation, hence rewarding processes or in a contrary case the retaliation against non-cooperative nodes. To this end, the requester, i.e. a UCN-node willing to access Internet over an available UCN-AP, and the requestee, i.e. the UCN-node willing to cooperate and give access to requesters, are defined in ULOOP. The requestee may operate in
1.1 Briefly the UCN Dynamics and Relevant Issues

Volunteer mode if the incentives are only based on reputation and trust parameters or in retailer mode if the monetization or virtual crediting system is used as a basis for incentive mechanism. In parallel to the base for the incentive mechanism, the requestee may earn credits or be reputed well after being cooperative, which may be spent utilized when working at requester mode.

The trust management is another vital mechanism for the UCN as this paradigm plays a great role for permanent willingness of stakeholders for cooperation. The definition of harmless behavior in the UCN, and a control over it, draws the boundaries of the privacy and security related suspenses. As the fully stochastic UCN coalescences may necessitate the intensive interactions among several strangers, the trust management is of vital significance in addressing the governance of the social behavior of the members.

The scalability of these mechanisms are the challenging issues. Addressing a central approach may raise the complexity from the point of scalability, whereas a distributed manner brings about the trustworthy in terms of dissemination among members. Yet another relevant issue is the feasibility of the proposed solutions.

1.1.1.2 Resource Allocation

The very first tackle concerning the resource allocation aspects is the scalability. Although the scalability is a core optimization item for various network architectures, it becomes more difficult to organize scalable UCN deployment, as additional challenges and complexity arise due to fully stochastic behavior in the formation, member coalescences and personal preferences. This stipulates self-organization among participants in order to achieve well-acceptable formation and roles of UCN nodes in a scalable manner for i) autonomously expanded capillarity, ii) robustness, iii) throughput maximization and finally iv) acceptable utilities either from a client or a provider perspective.

Alongside the stochastic formation of the wireless community, the fairness is another aspect to be handled in UCN, which is entailed by cooperation incentives, trust and the diversity of the network devices participating to the community. The members are motivated for the cooperation by means of the classification and prioritization of the cooperative and trusted ones. Additionally, more incentives and trust parameters are granted for the members serving with a better network performance during a UCN-AP mode. In other words, the social community dynamics and the necessity of continuity have great impacts on various resource allocation aspects.

As the dynamics of the resource allocation change when considering the UCN settings, various aspects should be revisited for a feasible and robust network oper-
1 Introduction

... Consideration. Considering haphazard formation of a community, the amount of available UCN-APs and clients may not be compatible from time to time. This complexity may stem from various reasons such as the environment characteristics, traffic behavior of the clients, mobility patterns, the transiency of operation modes based on user preferences, etc. Additionally the classical network challenges such as high interference, dynamic congestion cases and fluctuations in the formation of UCN members are potential environmental settings. Thus these aspects should be kept in mind when handling a resource allocation related improvements.

1.1.1.3 Mobility Management

Based on the similar reasonings related to the environmental characteristics, one may expect highly dynamic mobility patterns of the UCN-nodes. From the mobility management point of view, the UCN resembles, to a certain extent, the mobile ad hoc networks (MANET). In this context, the mobility anchor points and/or the coordination points may disappear from time to time, because of not only the mobility related characteristics, but also the user preferences. This behavior is referred as transiency for the rest of this thesis.

Considering the challenges for the mobility management aspects, the UCN faces similar tackles related to MANET. The additional complexity may arise from the motivation of the UCN members for longer-term AP modes due to aforementioned transiency problem.

1.1.1.4 UCN Communication Framework for the Self-Organization of UCN-nodes

For the discovery of the UCN nodes and related operation modes, a specialized communication framework is of vital importance. The haphazard formation of UCN community and the realization of collaborations are yet other motivations for the necessity of a communication platform or methodology.

Considering the self-organization, autonomic formation during network operation, resource allocation and mobility management aspects, the communication frequency of the UCN members’ interactions is high for the robustness and the feasibility issues. As illustrated in Figure 1.2, potential communications may take place with a form of full-duplex i) UCN-AP to UCN-AP ii) UCN-AP to non-resident UCN-clients and iii) UCN-AP to resident UCN-clients depending on the network scenario.

When it comes to shedding some light on the implementation and feasibility of such a communication framework; the beacon frames attract the researchers for the transmission of non-standard information. That is to say, the beacon frames
1.1 Briefly the UCN Dynamics and Relevant Issues

Figure 1.2: An Overview of the UCN Communication Framework

can be utilized for the application specific information transmission from an AP to the potential clients without association [13]. Another possibility is to introduce a certain amount of central UCN servers, through which i) the communication among UCN-nodes may flow, ii) several UCN members may receive live information on the deployment of various UCN coalescences and finally iii) the UCN-nodes may share cooperation incentives and trust management based information for the community persistency. In this thesis, the presence of a communication model is assumed, which realizes the interactions as given in Figure 1.2.

Having introduced different aspects and settings, we, next, present the potential challenges from the load balancing perspective based on the above-mentioned UCN enabling factors.

1.1.2 Challenges from the Load Balancing Perspective

Considering the core enabling functionalities, the system complexity drastically increases in the UCN environment due to the diversity of end-user devices, subscribed to UCN community, in terms of software or hardware capabilities. The dynamic environment and diversity of user entities (UE) joining to the UCN as micro-providers or as clients, are certain factors forming several challenges for the building-blocks of UCN. Additionally, the interests of stakeholders differ broadly with the changing UCN roles as a consumer or a provider.

Among these network dynamics, frequency and the duration of congestion instances play a great role for the persistency of the instant and future UCN commu-
1 Introduction

nities. This is due to the natural fact that the stakeholders would start leaving the community in case of unacceptable service quality. An expected increase in the density and uncoordinated operation of UCN-APs, combined with constantly growing traffic demands, would start hurting the users’ quality of experience (QoE) without any doubt. Thus a tolerable certain level of service quality should be guaranteed throughout the UCN environment.

In addition to congestion avoidance, fairness is another aspect, which is directly related with the service quality and the incentive parameters. The very changeable Wi-Fi capabilities of the subscribed UCN-APs result in a fluctuation in the service quality among various UCN subnetworks. Hence a fixed unit resource cost may not be feasible in different UCN sub-coalescences. In other words, certain mechanisms should be deployed in the UCN to keep the quality of service (QoS) measurements over an acceptable threshold. This motivation should additionally address i) conserving fairness in incentive mechanisms ii) improving revenue of a stakeholder operating with the provider mode and finally iii) preventing selfish nodes to collect all potential UCN-clients while operating in provider mode.

Another significant factor is the immensely dynamic and uncertain characteristic of the operation environment. A single strategy solution addressing aforementioned challenges may not be efficient as a specific action/algorithm may operate based on pre-assumed environmental conditions. Considering the individual mobility pattern of the UCN-APs, which doubles the impact of an assumption for a highly dynamic environment, an intended tactic may not be sufficient and hence put a risk on the persistency of the UCN.

The user centric approach cater for a specialized network platform, where the decision making entities may appear either on the content server or the consumer side with the communication skills for further collaborations. In other words, UCN is cut for the cooperations between end-user and the network. This is a potential advantage for the facilitation of not only the load balancing problem but also many other potential network related improvements by means of end-user collaborations. A potential contribution of the UCN-client for the facilitation of load balancing problem is smarter AP selection methodology. Additionally, the UCN-clients can provide live feedbacks for the quality of network services as a trigger for load balancing activities.

1.2 Our Load Balancing Approach

Addressing the previously described challenges, we propose an autonomous load balancing scheme for UCN-APs in order to provide a homogenous QoS among UCN-
subnetworks and tackle the dynamic behavior of the UCN environment. We propose a smart system with the capability of adapting to the changing environmental conditions. With the adaptation, we mean the ability to select more suitable and effective strategies against unbalanced load. To this end, the UCN-APs are equipped with intelligent methods, which are capable of characterizing the operation environment and monitoring the impact rates of their own available strategies against congestion cases.

As one of the potential strategies against unbalanced load, we study how the UCN-clients can be shared among UCN-APs during live time by means of forcing a certain number of clients for a migration to more suitable UCN-APs. This is performed in a fair manner taking the Wi-Fi and computation skills of the UCN-APs into account. Considering the basic assumption of node selfishness, a suitable incentive mechanism for load balancing is introduced to the UCN in order to prevent immense revenue oriented UCN-AP operation. That is to say, the providers are motivated for client sharing with an objective of keeping QoS over an acceptable threshold. It goes without saying that this mechanism is autonomic without human intervention. We additionally investigate the situations where manageable stakeholders join to the UCN environment with an access unit role, which are incapable of collaboration skills.

We propose the facilitation of load balancing by means of end user collaboration, where the end user plays an active role by i) providing live network performance related measurements, ii) characterizing UCN subnetworks and finally iii) taking the short-term UCN-AP functionalities into account while selecting a specific UCN subnetwork. This motivation led us to study the mobility awareness and a mechanism for detecting the short-term AP-client interactions. We equip the UCN-clients with the corresponding learning methods in order to realize above-mentioned objectives.

In this thesis, we validate our approaches by means or real implementation and testbed experiments. To this end, we additionally study the user centricity in testbed foundations. Next section provides a summary for the contributions, which constructs our approach.

1.3 Contributions and Thesis Organization

1.3.1 Contributions

An overview of the focus points in this thesis is given in Figure 1.3. Referring to this Figure, we summarize our primary contributions as follows:

- **User centric wireless testbed (1)** We start with an appropriate working plat-
1 Introduction

Figure 1.3: An Overview of the Contributions in this Thesis

form for the validation of the contributions in this thesis. In this approach, we set up a user centric wireless testbed (UCW), where the end-user participates to the network as a key component for the network control and operation. The testbed offers programmable entities in both core and access network edges, enabling researchers to implement cognitive and cooperative decision mechanisms for enhancing the end-to-end service experience. Using this testbed provided us with the opportunity to investigate feasibility aspects and improve our approaches. Section 5.1 presents the corresponding technical details.

- **User facilitated central load balancing (2)** Having established the experiment platform, we focus on a semi-distributed and user collaborated load balancing approach. In this approach, we discuss a load balancing strategy in UCN for the circumstances, where among \( \mu \)-providers only a certain number of APs
are capable of taking decisions for load balancing. In other words, the certain APs are not able to collaborate, nevertheless are assumed to be manageable and support UCN communication framework. In this work, we introduce an integrated solution to congestion avoidance and congestion-like anomaly mitigation problems. The proposed solution implements a Partially Observable Markov Decision Process (POMDP) as the semi-distributed control system. This contribution is introduced in Section 3.3.

1.3 Contributions and Thesis Organization

• Cooperation Incentives based load balancing (3) In this approach we extend previously introduced load balancing strategy for the UCN-AP pools addressing the distributed manner in load balancing. This solution focuses on the network scenarios, where the UCN-APs are capable of collaborating with each other. As the collaboration stipulates the additional challenge for motivation, the proposed method handles with the presence of selfish members. Moreover, the proposed solution bases on the previously discussed setting that the QoS provided by UCN-APs differs due to the hardware or software capabilities. As these capabilities span over a wide range, it brings about the adversity for the incentive mechanisms of UCN-AP cooperations. We propose a QoS-based crediting mechanism, providing incentives for the UCN-APs to share the load, i.e. the clients. We target thereby ensuring a homogenous level of QoS distribution throughout the UCN community. We discuss a utility-based decision making framework for the incentivized & cooperative load balancing. The main objective of this approach is to prevent the potential harms caused by selfish nodes. The details of this contribution are presented in Section 3.2.

• The autonomous control framework (4) After completing stable and feasible strategies for load balancing and looking deeply into the load balancing literature, we observed by various approaches the high dependency on the environmental conditions. This motivated us for concentrate on an autonom behavior in balancing the load, which addresses the improvements in strategies and adaptation to the changing environmental conditions. In this approach, we make use of a deductive method and focus on a more comprehensive problem. We propose an autonomous control framework, where the focus remains on autonomous resource utilization and strategy optimization in wireless networks. The control framework aims at an immensely dynamic, time varying, and less predictable network environments, which depict similar characteristics of UCN. The first motivation of this contribution is to eliminate redundant resource utilization for specific network objectives while reducing the network
response time. To this end, the decision making engines of the control framework characterize the network environment in terms of correlations with specific network objectives and improves the resource utilization accordingly. The second motivation is to match network strategies with different environmental conditions in terms of impact rates in healing critical cases. We develop a POMDP based control loop to improve the network strategies against alarm conditions (ref. to Section 4.1). The proposed control framework is kept modular, where the parameters of the solution can be modified addressing different problems. Finally the framework parameters are adjusted in addressing the load balancing by means of adapting to the changing environmental conditions. The details of this contribution is provided in Section 4.2.

- **AP selection in UCN (6) - (7)** For the facilitation of load balancing, we focus mainly on the UCN-client side impact on a balanced load. In these approaches, we deal mainly with the AP selection problem. The AP selection has considerable impact on the network operation, influencing the adaptive allocation of limited network resources, seamless connectivity and hence the network performance from the end user perspective. In UCN, information on the hardware or software capabilities, available network resources, instant AP load, etc., i.e. the network performance indicators are not available for the end user. Additionally, the presence of mobile APs or the tentative provider-modes raise the complexity in AP selection. Such instances put a great risk on an already balanced load and create fluctuations in the re-formation of load and AP match. This additionally brings about connectivity related difficulties (ref. to Section 2.2). Among various aspects of provider selection in UCN, we focus on a network performance history based AP selection approach. This is to help for a balanced load, where we propose UCN dynamics and stakeholders’ behavior aware strategy. The details of this contribution is provided in Section 2.1.

- **Mobility behavior modeling in UCN (5)** Having the network performance history based AP selection approach, we focus additionally on the mobility behavior of the UCN-clients. The mobility patterns are used as additional observation components in the previously discussed AP selection framework. The main motivation of this contribution is to reduce amount of load balancing attempts by the UCN-APs and help for the circumstances of relative transient interactions among specific UCN-APs and clients. In this study, we propose the mobility prediction approach by WLAN group formation. We focus on mobility behavior of UCN-clients and propose the WLAN group prediction
1.3 Contributions and Thesis Organization

Our approach is based on building the history of encountered WLAN groups, modeling an encounter instance of a new group as arrivals for the proposed model and finally adapting the prediction. This contribution is finally adapted the proposed AP selection framework. The details of this approach and the corresponding adaptation for AP selection are given in Section 2.2 and 2.3 respectively.

- Trusted cooperation platform for UCN-APs (8) It should be underlined that the previously discussed contributions are based on intensive collaborations among multiple stakeholders in UCN. Additionally, extensive resource utilization of the network resources are allowed to the strangers in the UCN due the nature of the social communities. The previously stated motivation of keeping the QoS indicators over acceptable thresholds concerns naturally the owner of the UCN-AP. In this part we focus on a $\mu$-Kernel based operating system (OS) virtualization as a trusted cooperation platform for the UCN-nodes, which is adapted for the aforementioned UCW testbed. Our main motivation is to come up with a solution addressing various UCN relevant issues such as i) preventing extensive resource utilization of the UCN-APs by others, ii) preventing malicious activities on the private OS instances due to UCN settings, and finally iii) providing a trusted platform for the multiple collaborations in UCN. We evaluate the network performances of the proposed solution, which is detailed in Section 5.2.

1.3.2 Thesis Organization

The organization of this thesis can be summarized as follows:

In Chapter 2 we introduce the user facilitation for the load balancing problem in UCN. This Chapter introduces the afore-mentioned AP selection method and transiency problem for the UCN-AP functionalities due to either mobility or user preferences. This chapter addresses the previously described contributions 5, 6 and 7.

The Chapter 3 is devoted for the cooperative load balancing approaches in UCN. This chapter provides details on the contributions 2 and 3.

We present the autonomous load balancing framework for the load balancing in Chapter 4. In this chapter we additionally introduce the autonomous framework for the resource allocation and strategy healing in wireless networks. The chapter includes the contribution 4.
1 Introduction

We introduce the proposed testbed and the trusted collaboration framework in Chapter 5, which provides technical details on the implementation and feasibility aspects. This chapter is composed of the contributions 1 and 8.

Finally we provide a summary and a general discussion part for the future work and potential extensions of the proposed methodologies in Chapter 6.
1.3 Contributions and Thesis Organization

Bibliography

In this thesis, we contributed for the following peer-reviewed papers as the outcome of this thesis. The contents of the these contributions are changed in accordance with the construction this thesis.


2 End User Cooperation for the Facilitation of Load Balancing in UCN

In this chapter, we comment on the end user cooperation for the load balancing problem in UCN by means of decision-making for AP selection. The chapter is comprised of three main sections. Section 2.1 provides an AP selection framework for the UCN-clients based on the network performance histories and client traffic behavior. Later in this chapter, the Section 2.2 details the connectivity related tackle in the UCN due to the participation of mobile APs or short-term AP functionalities as another potential network performance indicator. In this section we additionally comment on the mobility of the UCN-clients, which necessarily has a great impact on the AP selection. Finally, in Section 2.3 we briefly comment on the mobility awareness in AP selection.

2.1 AP Selection in UCN: A Network Performance History Based Approach

The AP selection has a considerable impact on the network operation, influencing the adaptive allocation of limited network resources and hence the quality of experiences (QoEs) for all clients. The very first difficulty for the AP selection is the unorganized deployment of the UCN-APs and the uneven distribution of the UCN-clients, i.e. the coalescences of UCN members with provider and consumer roles are fully stochastic. Thus one may agree on the puzzling circumstances, that the random distribution of the UCN consumers among randomly distributed providers, would cause an unbalanced load. This may, hence, result in unevenly experienced quality of services (QoS) among UCN-clients. As the uneven QoS distribution puts a risk on the continuity of this wireless social community, the AP selection plays a great role not only for a balanced load but also for the UCN persistency.

Although the AP selection problem is broadly discussed in the literature, due to the nature of the social communities, additional complexities should be taken into account while dealing with the AP selection problem. For instance, information on
instant AP loads, the hardware or software capabilities, available network resources, etc., i.e. network performance indicators are not available for the end user in the UCN. Additionally, the presence of mobile APs and user preferences in tentative AP modes raise the complexity in AP selection as the longevity of an AP functionality is not provided for the corresponding UCN members. The main motivation of the autonomous UCN organization with less human intervention [2], in addition to the aforementioned dynamics and the limited resources, brings about the difficulty in service guarantee. Besides it should be highlighted that the real time load information on a specific UCN-AP, which is disseminated by the UCN-AP itself, may not be realistic and trusted as a selfish node may not be willing to share this information in order not to risk own revenue. Hence the decision instances on the AP selection may rely only on partial information and past mutual experiences based information.

In this section we discuss AP selection problem in UCN from the end-user perspective with the consumer role. Among various concerns of provider selection in UCN, we focus on a network performance history based AP selection approach. We propose UCN dynamics and stakeholders’ preferences aware approach. To this end, we make use of learning methods for UCN-AP characterization in terms of network performance. The characterization of a UCN-AP takes the mutual past experiences between the UCN-client and a specific UCN-AP into account. Additionally we define the AP attractiveness model from the end-user perspective by means of end-user preferences in network traffic categories. The main motivation of our approach is to introduce user cooperation in load balancing among UCN-APs and prevent redundant credit losses due to participation of relatively low performant providers.

The rest of this section is organized as follows: Section 2.1.1 provides a literature review on the access point selection in wireless networks. In Section 2.1.2, we evaluate the state of the art solutions with a UCN settings perspective. Section 2.1.3 is devoted for the statement of our solution, the technical details of which are given in Section 2.1.4. Then after, the proceeding section, the Section 2.1.5, we comment on the verification and evaluation of the proposed model with the testbed experimentation.

2.1.1 State of the Art

Miyata et al. [14], endorse the common AP selection algorithm based on received signal strength indicator (RSSI) by arguing performance anomaly [15] problem. They propose an algorithm, which imposes the collaborative behavior of the new comers to the WLAN in terms of willingness to move to a network guided appropriate AP. On the contrary Nicholson et al. [16], claim that the AP selection based on RSSI
results in not a better throughput and bit-rate in the WLAN in comparison with the selection of APs in a randomized way. They propose a network management software (Virgil), which continuously scans APs around and associates instantaneously in order to perform performance tests.

The [17] focuses on an alpha-optimal load balancing approach by means of AP selection, where alpha refers to rate, throughput, delay and load equalizing terms. A distributed user association policy based on the optimal load vector mapping is iteratively applied, which globally adapts the spatial load. The coordination among APs for the interference avoidance is skipped in this study.

In [18], the delay in the actual beacon frames are assumed to be an indicator for the load of the AP and the contention on the wireless medium, which in turn gives the client the ability to estimate available bandwidth and affiliate for the APs accordingly. Similarly, [19] proposes an AP selection method based on the available bandwidth, which is estimated by passively monitoring the transmission channels and calculating the channel utilization total time in a period. A similar real-time measurement based AP selection technique [20], two different algorithms are proposed for the estimation of traffic loads among the APs during camp on time. In this study the load of APs are estimated by observing the delays between probe request and the probe response frames. An appropriate AP selection among multiple APs is proposed based on the load estimation and hence balancing the load.

Tajima et al. [21], analyzed AP selection problem in wireless infrastructure mesh networks WIMNET in order to decrease the total traffic in between APs due to routing activities by selecting appropriate APs on the routing path. A simple mathematical formulation is proposed and the authors claim that the proposed model guarantees less interferences in the WIMNET due to the less traffic between APs. Yilmaz et al. [22], evaluate different access point selection methodologies based on the utilized metrics and in terms of achieved bit-rates. They propose utilizing RSSI as the basic and the most efficient access selection methodology due to the unpredictable user behavior and spatial access point distributions. In [23] simple methodologies are proposed for i) the AP selection algorithm for Maximizing Local Throughput and ii) for AP selection for avoiding APs with larger transmission error probabilities, where the packet transmission error and the number of clients should be available as a metric for stations (STAs). [24] proposes a centralized call admission controller called HRFA (high-rate first association), which targets a load balancing mechanism from the client perspective.

In [25], the network selection is formulated as a non-collaborative game where each client tries to maximize her own utility based on the throughput provided by
End User Cooperation for the Facilitation of Load Balancing in UCN

APs and the fee paid for the Internet access. A no-regret learning mechanism is proposed in order to guarantee a convergence to equilibrium among the network. In this study, the throughput information is assumed to be available for the clients during association procedure. In one another game theoretical approach [26], it is claimed that the utilities of clients would be higher when changing their location at the airport and move to a less overloaded access point if a centralized prompting mechanism is available. In one another utility-based decision-making approach [27], the client entities are assumed to have the capability of simultaneously receiving and decoding multiple channels. With this assumption, the problem is formulated with convex system settings. Additionally it is assumed that the clients exhibit quasi-static mobility patterns, i.e. the clients prefer longer session times in the same physical place. A periodical iterative load-balancing algorithm is proposed, which is claimed that balances the network load among APs within 3-9 iterations. The load balancing problem is formulated based on the parameters i) the channel usage time between AP and an STA and ii) the capacity for a special STA. Finally the STAs traffic is distributed among APs based on the maximum sum utility with time allocations, which is iteratively calculated, and by means of multiple Wi-Fi interfaces.

As can be deduced from the literature review, the load metrics, which are proposed as the reasoning parameters for decision making, play a great role in AP selection. Various measurement taking methods, either passive or active, are discussed in order to define the load metrics. In addition to the afore-mentioned AP selection methodologies, we provide briefly additional load metrics defined for the WLAN APs.

Garcia et. al. [28] claimed that the number of associated clients may still be load indicator for the APs as the collision probabilities rise with the increasing client number. Furthermore the number of retransmission attempts for a successful packet transfer indicates the load level of an AP. Similarly, packet loss estimation is also proposed to be a metric, which is a different way of expressing the previous load metric. The packet level measurements, such as the measure of busy time, i.e., the channel occupation time or the AP buffer queue are utilized as load indicators in the literature. Secondly the Beacon frames include i) BSS average access delay (which is based on the channel load reports and is the average time for the delay from the ready time for a transmission until the actual frame start time) ii) BSS AC access delay and iii) the BSS load containing number of associated clients, channel utilization (the percentage of time, where the AP senses that the medium is busy) and finally the available admission capacity (AAC) which is the complementary
component for the channel utilization percentage. In [29], average latency, jitter
and the packet losses are defined as the load metrics, which are also proposed as the
QoS indicators for the WLAN. The authors of [30] make use of the total delay time
between probe requests and responses as the load metrics for the STAs during the
association phase.

2.1.2 Evaluation of the Literature from the UCN Perspective

The UCN community consists of end-user entities with various and dynamic Wi-Fi
capabilities, either in terms of the software or the hardware aspects. For instance, a
UCN member may join to the community with a well equipped AP providing higher
QoS or with a mobile smart phone, which may support AP functionality, however,
is not specialized for it. Additionally, the network performances may also vary
due to the alternations in the quality of subscribed broadband access technology
or the network operator. This, in turn, results in a wide variety in the provided
QoS among the sub-networks of UCN. Considering these settings for the UCN, the
definition of AP selection problem differs from the ones proposed in the literature.
Mainly identical APs in terms of hardware and software capabilities are taken into
account in the literature. One another proposed easiness among the AP selection
methodologies in the literature, i.e. the presence of a dissemination mechanism for
the AP load, may not be applicable for the UCN environment as selfish nodes may
not be willing to share this information. This is due to the potential revenue oriented
motivations of selfish UCN-APs.

Various studies focus on an AP selection mechanism by means of semi-distributed
frameworks. With the semi-distributed framework, we mean the presence of a central
unit facilitating the AP selection based on additional information exchange for the
available APs in terms of the load. This is, however, not applicable for the UCN
environment as the main cornerstone is to provide user entities with autonomic
formation ability in a distributed manner. That is to say, the UCN formation and
hence information exchange by means of a central unit is not applicable for the
UCN.

Additionally, the UCN-APs may collaborate for the self-organization aspects in
the UCN. In other words, they may attempt on balancing the load by means of
client migrations. The QoE of a client would be affected in a worse manner in such
a scenario between UCN-APs as the seamless handover may be not guaranteed.
Such scenarios are not yet discussed in the literature as they are not convenient
for many of the proposed solutions. In the UCN, however, this dynamic should
be taken into account in making decisions for the AP selections as a less number
of UCN-AP initiated load balancing attempts would prevent connectivity related tackles and additional traffic load due to relevant messaging. These extra messaging may include information on resource negotiations, client collaborations and client associations in a migration scenario. A less number of migration instances may improve the operational capabilities of UCN community as the additional network load would be eliminated.

One another issue worth mentioning is the assumption of identical traffic behavior among clients. Many solutions in the literature focus on a network environment, where clients have similar tendencies on the utilization of various traffic categories. We believe that the inclusion of traffic category tendency of a client would yield a better client-AP distributions in the UCN. More information on this can be found later in this chapter.

Finally, the decision-making in AP selection may base on partial information due to the aforementioned UCN-specific settings. Nevertheless, various studies in the literature assumes the availability of complete information for the AP selection problem in WLAN.

2.1.3 Our Approach

We focus on a distributed AP selection method, which is basically driven by the AP characterization in terms of network performance related parameters. As the proposed framework takes a set of performance indicators, we comment on an active measurement taking system for potential network performance indicators in WLAN. This system provides a base for the contributions. By means of intelligently aggregating the collected indicators, we target UCN-clients’ tendency for camping on relatively better UCN-APs in a distributed nature. This, we believe, yields better UCN experiences from the clients’ perspective. In our approach, the decision of AP selection in dynamic and uncertain UCN environments takes the client migration and connection refusal statistics into account in addition to the most commonly used QoS parameters (i.e., delay, packet loss, bit ratio, and Jitter). These parameters help infer the load and congestion frequency of UCN subnetworks, and also assist in future decision makings. This is to yield for specific UCN-AP and UCN-client sub-coalescences as the migration instances or the connection refusals are restricted to a specific UCN-AP and UCN-client pair. In other words, these instances are individual and hence differs among UCN-clients. In this manner, a well-characterized UCN-AP for specific UCN-clients can be unattractive for others. This prevents potential overloads on specific UCN-APs, hence, decrease the number of client migration instances.
In order to provide the UCN-clients with the ability of characterizing the UCN-APs in terms of network performance related indicators, we propose using learning methods, which, in our case, are based on expectation maximization (EM) algorithm. Using this method, we obtain probability distribution functions (pdf) over various performance indicators. Focusing on the highly dynamic characteristics of the UCN, we propose a new method for the utilization of EM. In the literature of EM utilization, various attempts are based on the assumptions of a prior mixture-distribution. Our approach, however, sidesteps this assumption to a certain extend. To the best of our knowledge, our method is a novel approach in the literature of EM utilization. This method improves the pdf constructions for random variables in highly uncertain domains like UCN.

Finally, in conjunction with the traffic categories preferred by the client, we describe a joint attractiveness of a specific UCN-AP. The attractiveness term, then after, is utilized in AP selection. Having constructed the AP attractiveness, the impact of the precision, i.e. the certainty level of the believes is introduced while taking the decisions. More details for the proposed solution and construction of the model are provided in next sections.

Finally it is worth to mention that the proposed methods are validated by real implementation and testbed experiments. We believe that these approaches can be generalized for the company or campus wireless networks by easily manipulating the framework parameters.

2.1.4 Technical Details of the Proposed Model

The proposed AP selection method comprises of two main blocks, namely, i) the learning method characterizing the experienced UCN-APs in terms of network performance and ii) the decision-making phase based on the characterization parameters. The core enabler of the proposed method is the prediction on how likely the performance indicators alternate based on different UCN-AP operation states and environmental conditions. The proposed solution takes additionally the SNR measurements during the clients’ residency at a UCN-AP into account.

In the proceeding section, we initially comment on how we collect network performance indicators and then after discuss the proposed framework.

2.1.4.1 Network Performance Indicators used for the Proposed Framework

Our model relies on a set of passive and active measurements. The passive measurements correspond to events of connection refusals or migration instances (ref.
to Section 1.1), which are taken without frequently exploiting the network resources. However, active measurements are extracted by monotone network performance tests, which are carried out randomly and for random durations. Active measurements are basically the frequently measured QoS parameters.

The dimensions worth focusing in active measurements is their frequency, duration and coordination of clients. And to roughly provide a distributed coordination mechanism of client requests for the active measurement sessions, we make use of the well-known distributed coordination function (DCF) approach. We believe that this reduces the additional complexity in client coordinations in a scalable manner and reduce the potential congestions caused by too frequent re-coordination attempts.

In Figure 2.1, we give a pictorial representation of the measurement pattern.

![Figure 2.1: Timing Diagram on Client Entities for the Network Performance Tests](image)

It should be noted that the focus of the work is confined to collecting the measurements in the last mile of UCN environment and do not consider the QoS parameters values in backbone and core network. We answer two basic questions, when it comes to active measurement taking i.e., i) what should be network test duration? and ii) what should be the inter-arrival time? This is realized by randomly carrying out the network performance tests and random back-off durations, which are exponentially distributed with different mean values. We model such active measurement taking system as M/M/1/m. In order to provide a better understanding, we firstly summarize the M/M/1/m system parameters.

![Figure 2.2: Transitions between States of a Markov Chain](image)

**2.1.4.1.1 M/M/1 System with Finite Queue** The Figure 2.2 [31] represents the transition parameters of a continuous time Markov chain with m finite states, where \( \mu \) stands for the service time parameter and the \( \lambda \) is the arrival rate of the requests.
to the system. In other words, a transition from a lower state to a higher occurs with the parameter $\lambda$, whereas a transition from a higher state to a lower one occurs with the parameter $\mu$. Considering the steady-state conditions, where the probability distributions of the finite states $p_i$ are independent of time factor:

\begin{align*}
p_0 &= p_0(1 - \lambda) + p_1\mu \\
& \vdots \\
p_i &= p_{i-1}\lambda + p_i(1 - \mu - \lambda) + p_{i+1}\mu \\
& \vdots \\
p_m &= p_{m-1}\lambda + p_m(1 - \mu) \tag{2.1}
\end{align*}

Given the $\rho = \frac{\lambda}{\mu}$ as the service occupation rate; under the constraint $\sum_{i=0}^{m} p_i = 1$ for the substitution for the $p_m$

\begin{align*}
\sum_{i=0}^{m} p_i &= \sum_{i=0}^{m} (\frac{\lambda}{\mu})^i p_0 = \sum_{i=0}^{m} (\rho)^i p_0 = (\frac{\rho^{m+1} - 1}{\rho - 1})p_0 \\
\implies p_m &= \frac{\rho m+1 - \rho m}{\rho^{m+1} - 1} \tag{2.2}
\end{align*}

The given probability $p_m$ is defined as the probability of dropping a new request as there exists no available waiting room for the new comers, hence plays a great role in the design phase of a system. Knowing the steady-state probability distributions, one may define the expected number of requests in the system $R$ as:

\begin{align*}
R &= \frac{(m + 1)\rho^{m+1}}{\rho^{m+1} - 1} + \frac{\rho}{1 - \rho} \tag{2.3}
\end{align*}

Finally as another important design parameter, the mean sojourn time $S_j$, i.e., the expected time spent by clients in order to be completely served is:

\begin{align*}
S_j = \frac{R}{\lambda(1 - p_m)} \\
&= \frac{(m + 1)\rho^{m+1}}{\rho^{m+1} - 1} + \frac{\rho}{1 - \rho} \frac{1}{\lambda(1 - p_m)} \tag{2.4}
\end{align*}
parameter | description | value
--- | --- | ---
$N$ | Number of associated clients | -
$\tilde{N}$ | Set of associated clients | \{1, 2, …, $N$\}
n | Finite size of queue | -
$\lambda_i$ | Inter-arrival distribution parameter of $i \in \tilde{N}$ | -
$\lambda_{tot}$ | Total arrival parameter of proposed system | $\sum_{i=1}^{N} \lambda_i$
$\mu$ | Service time parameter of proposed system | -
$\rho$ | Service occupation rate | $\frac{\lambda_{tot}}{\mu}$
$p_n$ | Probability of dropping a new request | $\frac{(n+1)\rho^{n+1}}{\rho^{n+1} + 1 - \rho}$
$R$ | Expected number of requests in the system | $\frac{\lambda_{tot}(1-p_n)}{\rho(1 - p_n)}$
$S_j$ | Expected sojourn time per requester | -
$T$ | Expected number of requests being served | -

Table 2.1: Summary of Parameter Set for the Active Measurement Queue

Detailed information on the M/M/1/m systems and the dedication of the corresponding system parameters can be found in [32, 31, 33]. The summary of the utilized parameters and the mathematical model of the proposed system for the active measurement taking is provided in Table 2.1.

In the proceeding part, we detail the design objective for the proposed system and discuss on the derivations for the aforementioned network test durations and network test frequency related with the objective.

### 2.1.4.1.2 The Design Objective for the Active Measurement Taking System

The intensity of test requests mainly depends on the number of resident clients, which is related with the inter arrival time distributions of the UCN-client associations and disassociations. However, we initially restrict our interest for the network test requests on i) criticality of the instant resident client numbers, i.e., the maximum amount of resident clients, ii) the test durations and iii) inter arrival distribution parameters. Our main motivation is to take the limited UCN resources into account.

The main objective behind the stated motivation is to minimize the costs due to the active measurements from the network resource utilization point of view in a distributed manner. That is to say, the service time, i.e., the network test duration should be low, however, enough to capture the network capacity. Besides, each client should be allowed allocating the service port, i.e. the probability of dropping a request should be minimized. Finally, the implementation complexity and the sojourn time for a client should be decreased.

The discussion so far dictates that an intuitive optimization problem is maximizing the expected number of measurement requests constrained by congestion avoidance
2.1 AP Selection in UCN: A Network Performance History Based Approach

conditions. One may think of attaining this objective by minimizing the sojourn
time of a requester and increasing the expected number of requests being served.
Thus, we pack these controlling variables into a simplistic relationship and declare
the objective function for optimizing active measurements as:

\[ f(\cdot) = \frac{T}{S_j(\rho)} = \frac{T(n, \rho)}{S_j(\rho)(n, \rho)} \]

\[ \implies \rho(1 - \frac{\rho^{n+1} - \rho^n}{\rho^n + 1 - \rho^n}) = \frac{(\rho^{n+1} + 1 - \rho^n)}{\rho^n + 1 - \rho^n} \]

(2.5)

It should be underlined that the \( \lambda_{tot} \) in the definition of \( S_j \) is replaced with the
service occupation rate, i.e. the \( \rho \) in \( S_j^{(\rho)} \), as these terms are directly proportional
to each other. Let us now briefly highlight the constraints. As a constraint, for
guaranteeing the growth of the queue up to a finite number, we consider the first
boundary as \( 0 < \rho < 1 \). Secondly, we assume a practical constraint for the number of
potential UCN clients as 20, which implies that the boundary for the finite queue size
is \( 1 \leq n \leq 20 \). We assume that the objective function is compact and bounded set. A
closer look at the objective function (i.e., multivariate) and constraint function (i.e.,
inequality constraint) dictate that it is solvable using approaches like Lagrangian
Multiplier (Hint: which will be outcome of solving four variable four equations). In
this study, we avoid providing details of the solution, however, for ready reference,
In Figure 2.3, we show the behavior of different variables against the constrained
parameters. In Figure 2.4, we present the critical points and corresponding contour
for the aforementioned problem.

Based on the aforementioned setting and experimental runs, we choose the param-
ter set for the tuple \( (\rho, n) \) as \( (0.95, 1) \) and set the service time parameter, \( \mu \) as \( 1/5 \),
i.e. 5 sec test duration per client. This implies that for 20 clients, \( 0.95 = \frac{\lambda_{20}}{(1/5)} \),
\( \lambda / 105 \), i.e. each client requests network tests once per approximately 1.75 minutes
in average. For the network performance tests, we make use of D-ITG tool [34],
where the clients generate exponentially distributed 750 byte in average packet size
and 100 packets (also exponentially distributed) per second, which in total corre-
sponds to 375k in average for each test and, which we believe is a reasonable and
sufficient amount in order to capture the realistic network capacity.
2.1.4.2 Network Performance History Based UCN-AP Selection

Based on the past experiences of a UCN-client, i.e. the previously discussed network performance indicators, the UCN-AP is characterized. The main objective is to predict on various distributions of the rates for the critical observations. Due to the fact that the environmental characteristics play a great role in the quality of the provided network services, one should take the physical characteristics into account to capture the realistic parameters. In order to have a sense on the environmental
2.1 AP Selection in UCN: A Network Performance History Based Approach

conditions we make use of commonly utilized SNR values as a starting point.

The proposed framework comprises of two parts; the first part is the aggregation
of measurements, which is the construction of probability distributions on the rate
of critical observations and the second part is the decision over UCN-AP selection.
With this manner, we target the facilitation of the load balancing in UCN. In the
proceeding section, we detail our methodology for the characterization of the UCN-
APs, where we discuss the learning methodology that we propose.

2.1.4.2.1 UCN-AP Characterization in terms of Network Performances provided
in Different Environmental Conditions

The proposed network performance measurements considers Wi-Fi capabilities and AP load. The QoS parameters’ sensitiv-
ity to real-time and non-real-time traffic is considered to be the same as detailed in
literature. Our main motivation is to construct probability density functions (pdf)
for each performance indicator specific to the sensed SNR level. This stage is the
aggregation layer of the raw information, i.e. the observed parameters including
the passive and the active measurements. We mainly focus on the expected value
and the variance of either the duration or the number of critical / well indicators.
A probability distribution over the rate of critical observations is estimated using
Expectation Maximization (EM) algorithm over a mixture distribution. In order to
provide a better understanding for the model construction, we firstly comment very
briefly on the EM algorithm.

2.1.4.2.1.1 The EM Algorithm

The parameter estimation of the distribution for
a random variable becomes more complicated in case of uncompleted data set due to
hidden variables [35]. EM algorithm is an iterative optimization algorithm proposed
for estimating unknown parameters based on measured data [36]. The EM algorithm
relates the missing measurement data with a lower-bound posterior distribution and
optimizes the bound with a maximization step. In this part we briefly comment on
the mathematical formulation of the EM algorithm. Nevertheless more information
on the mathematical formulation and different applications of EM algorithm can be
found in [37, 35, 36, 38, 39, 40, 41].

The EM algorithm is a maximum likelihood (ML) estimation method for the in-
complete data and where the behavior of the observed data alternates based on
latent variables. The algorithm consists of two steps, namely the E-step (the expect-
tation step for the estimation of missing data) and the M-step (the maximization
step for the prior model parameters that are most likely based on the observed data).
Given a set of observed and incomplete data $X$, which is known to be resulted from

31
a parametrized data generator with parameter set of $\theta$, the main motivation of the EM algorithm is to maximize $P\{X|\theta\}$ by regulating the parameter set $\theta$ in an iterative way. The term iterative raises the question on the convergence of the proposed algorithm. Considering that the $\theta[k]$ is the estimated parameter set at the $k^{th}$ iteration, the iteration stops, i.e., the algorithm converges once the inequality holds for the next iteration $||\theta[k+1]−\theta[k]||<\epsilon$ for some pre-defined $\epsilon$ value. In other words, the iteration stops once the parameters tend to remain constant. Denoting the hidden variable set with $Z=\{z_1, \cdots, z_m\}$, the probability of observing the vector $X$ is;

$$P\{X|\theta\} = \sum_{i=1}^{m} P\{X|z_i, \theta\} P\{z_i|\theta\} \quad (2.6)$$

Representing the log-likelihood of 2.6 with $\mathcal{L} = \log P\{X|\theta\}$ for the next estimation and given that the $k^{th}$ estimated log-likelihood is $\mathcal{L}^{[k]}$, the main motivation is to maximize $\mathcal{L}−\mathcal{L}^{[k]}$. Using Bayes’ theorem and Jensen’s inequality [42] (as the log function is convex),

$$\mathcal{L}−\mathcal{L}^{[k]} = \log \left( \sum_{i=1}^{m} P\{X|z_i, \theta\} P\{z_i|\theta\} \right) - \log P\{X|\theta[k]\}$$

$$= \log \left( \sum_{i=1}^{m} P\{z_i|X, \theta[k]\} \frac{P\{X|z_i, \theta\} P\{z_i|\theta\}}{P\{z_i|X, \theta[k]\}} \right) - \log P\{X|\theta[k]\} \quad (2.7)$$

$$\geq \sum_{i=1}^{m} P\{z_i|X, \theta[k]\} \log \left( \frac{P\{X|z_i, \theta\} P\{z_i|\theta\}}{P\{z_i|X, \theta[k]\} P\{X|\theta[k]\}} \right)$$

The next iteration term $\theta[k+1]$ is selected in such a way that the last term in the Equation 2.7 is maximized.

$$\theta[k+1] = \arg\max_{\theta} \sum_{i=1}^{m} P\{z_i|X, \theta[k]\} \log \left( \frac{P\{X|z_i, \theta\} P\{z_i|\theta\}}{P\{z_i|X, \theta[k]\} P\{X|\theta[k]\}} \right) \quad (2.8)$$

Ignoring the constant terms in the Equation 2.8 and with a set of basic simplifications, we obtain:
2.1 AP Selection in UCN: A Network Performance History Based Approach

\[ \theta^{[k+1]} = \arg\max_{\theta} \sum_{i=1}^{m} P\{z_i | X, \theta^{[k]}\} \log (P\{X, z_i | \theta\}) \]

\[ = \arg\max_{\theta} [E_{Z|X,\theta^{[k]}}] \tag{2.9} \]

The term \( E_{Z|X,\theta^{[k]}} \) constructs the E-step and the maximization of this term with respect to the \( \theta \) is the second step, which is known to be the Maximization step. A very basic and a generalized flow chart of the EM algorithm is represented in Figure 2.5.

![Figure 2.5: EM Algorithm Flow Chart](image-url)
Due to the aforementioned highly stochastic settings of the UCN subnetworks, a classical approach on the definition of main network performance related characteristics may not catch the realistic system properties. In other words a prior assumption of a specific probability distribution model may bring about the convergence problem of an EM approach in defining the matter of subject. This is due to the fact that a monitored highly dynamic behavior would have a great impact on the pre-assumed distribution parameters. Based on this reasoning, we propose a new approach in defining the prior information on the distribution and which neatly sidesteps the necessity of an assumption on the probability distribution or even mixture characteristics. We discuss model construction with additional preliminary information in order to provide better understanding of the proposed framework. We start with basic mathematical background to motivate the reader for the model construction.

**Lemma 1.** For a set of coefficients $\{a_i, b_i\}_{i=1}^{N} \in R^+$ and $L \in R^+$ there exists at least one random variable $\chi$ with a continuous sample space $-L \leq \chi \leq L$, the likelihood of which can be defined by the following function $f_\chi(x)$.

$$f_\chi(x) = \begin{cases} \sum_{i=1}^{N} (a_i + b_i + a_i \cos(i\pi x/L) + b_i \sin(i\pi x/L)) / 2L[\sum_{i=1}^{N}(a_i + b_i)] & -L \leq x \leq L \\ 0 & \text{elsewhere} \end{cases}$$ (2.10)

**Proof.** $f_\chi(x) \geq 0, \forall x$ as $2L[\sum_{i=1}^{N}(a_i + b_i)] > 0$ and

$$\sum_{i=1}^{N} (a_i + b_i) \geq \sum_{i=1}^{N} (a_i \cos(i\pi x/L) + b_i \sin(i\pi x/L)), \forall x, i, L$$ (2.11)

Additionally, $\int_{-\infty}^{\infty} f_\chi(x) = 1$, thus the proposed function $f_\chi(x)$ defines a probability density function in the given sample space.

**Lemma 2.** Every probability density function can be approximately represented within the sample space $-L \leq \chi \leq L$ with the form given in Eq. 2.10 and with a sufficiently large number of sine and cosine components, i.e. $N$.

**Proof.** The Eq. 2.10 is the Fourier Series Expansion of a periodic function with the period of $L$ as $N \to \infty$. Hence an approximation on a probability density function is possible once a realistic assumption on the restriction for the sample space is possible depending on the application area.

As commonly practiced in various disciplines in the applications of Expectation Maximization (EM) algorithm, a prior information is assumed on the mixture distribution characteristics of the subject random variable [43, 44, 45, 46, 36, 38, 35]. In
2.1 AP Selection in UCN: A Network Performance History Based Approach

general, mixture of different probability densities with similar geometric shapes are
given as the prior data, where the latent variables are responsible for drawing the
next samples. The main motivation behind using EM algorithm is to estimate the
parameters of the mixture components and find out how likely the estimated densi-
ties are responsible for drawing future samples. As the UCN is a highly stochastic
environment, prior assumptions on the characteristics of mixture density may not
yield reliable estimation on the behavior of various types of UCN entities. Thus we
propose making use of Fourier Extension, sets of which may construct the mixture
components rather than constructing the model based on prior model assumption.

Given the Fourier coefficient set \( \Gamma = \{a_i, b_i\}_1^\infty \) and a limit \( L \) as the realistic limit for
the sample space, the probability of \( k \) critical observation on a single measurement
component of performance indicators is

\[
P\{\tilde{k} = k|\Gamma\} = \frac{1}{2L} \sum_{i=0}^{\infty} (a_i + b_i) \sum_{i=0}^{\infty} (a_i \cos(i\pi k/L + b_i \sin(i\pi k/L)) \quad (2.12)
\]

In this model, each sinusoidal component represents the probability densities, which
are responsible for the future draws of each sample and are latent variables. Consid-
erng the environmental states for the hidden variables of EM model is represented
by the set \( Z = \{z_i\}_0^\infty \), where the probability distribution of hidden variables are
given by the set \( \Gamma \). Having a set of critical observation rate \( K = < k_1, k_2, \ldots, k_n > \)
as the counter vector, given \( z_p \in Z \),

\[
Pr\{z_p | \tilde{k} = k_n, \Gamma\} =
\frac{a_p(e^{j\pi k_n/L} + e^{-j\pi k_n/L}) + b_p(e^{j\pi k_n/L} - e^{-j\pi k_n/L})}{\sum_{i=0}^{\infty} (a_i \cos(i\pi k_n/L + b_i \sin(i\pi k_n/L))}
\quad (2.13)
\]

It should be highlighted that we make use of Euler’s form and rewrite the sine
and cosine terms. This is to prevent potential misunderstandings due to phase
terms of the coefficients as negative terms may seem to appear at the first glance.
By analyzing the cosine and sine terms of \( z_p \) as sub-latent variables as \( z_p^c \) and \( z_p^s \)
respectively,

\[
Pr\{K, z_p^c | \Gamma\} = \prod_{i=1}^{n} a_p(e^{j\pi k_n/L} + e^{-j\pi k_n/L})
\quad (2.14)
\]
As commonly practiced in EM algorithm, taking the log of Eq 2.14 we get,

$$\log(Pr\{K, z_p \mid \Gamma\}) = \sum_{i=1}^{n} \log\left\{a_p\left(\frac{e^{j\pi k_n/L} + e^{-j\pi k_n/L}}{2}\right)\right\}$$

(2.15)

Hence the E-step is:

$$E_{Z\mid K; \Gamma} = \sum_{p=0}^{\infty} \sum_{i=1}^{n} T_{p,i} \log\left\{a_p\left(\frac{e^{j\pi k_n/L} + e^{-j\pi k_n/L}}{2}\right)\right\}$$

(2.16)

Using Jensen’s Inequality, i.e. as the log function is a convex function, a higher layer lower boundary for the $E_{Z\mid K; \Gamma}$ term can be constructed for the M-step as follows;

$$E_{Z\mid K; \Gamma} \geq \sum_{p=0}^{\infty} \sum_{i=1}^{n} T_{p,i} \log\left\{a_p\right\} \left(\frac{e^{j\pi k_n/L} + e^{-j\pi k_n/L}}{2}\right)$$

(2.17)

For the M-step, the objective is to find out the coefficients $\{a_i, b_i\}_{1}^{\infty}$, which maximizes the lower boundary $\tilde{E}_{Z\mid K; \Gamma}$ given in Eq. 2.17,

$$\frac{\partial E_{Z\mid K; \Gamma}}{\partial a_p} = \sum_{i=1}^{n} T_{p,i} \frac{1}{a_p} \cos(p\pi k_i/L)$$

$$\Rightarrow$$

$$a_p^{(t+1)} = \sum_{i=1}^{n} T_{p,i} \cos(p\pi k_i/L)$$

(2.18)

Using the Eq. 2.17 and due to symmetric property of the Eq. 2.16, we obtain the following equations for the $(t+1)$ iterations for both parameters as illustrated in Eq. 2.18.

Although a prior model assumption may hold for various natural phenomena, highly dynamic and stochastic domains may still not be easily modeled with prior distributions. The UCN environment is an unpredictable environment due to various
2.1 AP Selection in UCN: A Network Performance History Based Approach

reasonings such as the stochastic coalescences of the clients, variety of the utilized network entities either as consumer or provider and etc. As an advantage, the proposed method neatly sidesteps the obligation of using a prior model assumption when using the EM algorithm. Nevertheless, it should be highlighted that the proposed methods necessitates a track on infinitely many Fourier series components, which is not realistic for the practical implementations. Hence we propose limiting the number of components to an acceptable finite number depending on the practices.

As illustrated in Figure 2.6, this approach is evaluated by randomly drawing samples of various random variables with different probability distributions. It is observed that the variance plays a great role on the estimation for the proposed model. This is due to the fact that the approximation by means of finite Fourier series expansion, the estimated distribution spreads over x-axis which shows its impact on the normalization factor of the probability distribution. As a result the estimated distribution is suppressed around the mean value.

\[ f_X(x) \]

![Figure 2.6: Experiment Results on the Proposed EM Algorithm](image)

Figure 2.6: Experiment Results on the Proposed EM Algorithm
Though the suppression around the mean value, which is due to higher variances, is a drawback of the proposed methodology, we eliminate this negative effect in the decision making phase. A higher variance and the suppression indicates the level of uncertainty for monitored stochastic phenomenon. A higher uncertainty on the other hand plays the role of negative feedback term for the decision making phase, which is detailed in the proceeding part of this section.

Finally this stage forwards the variance and the expected value of the constructed distributions. Let the p.d.f of the random variable $\tilde{x}$, having the sample space within $[-L, L]$, is given as:

$$f_{\tilde{x}}(x) = \frac{1}{2L} \sum_{i=0}^{\infty} (a_i \cos(x \frac{i\pi}{L}) + b_i \sin(x \frac{i\pi}{L}))$$  \hspace{1cm} (2.19)

where the $\{a_i, b_i\}_{i=0}^{\infty} \in \mathbb{R}^+$. We are interested in calculating the expected value and the variance, i.e. the $E\{\tilde{x}\}$ and $\text{Var}\{\tilde{x}\}$ respectively. We start with the $E\{\tilde{x}\}$:

$$E\{\tilde{x}\} = \int_{-\infty}^{\infty} x f_{\tilde{x}}(x) dx = \int_{-L}^{L} x f_{\tilde{x}}(x) dx = \int_{-L}^{L} x \left[ \frac{1}{2L} \sum_{i=0}^{\infty} (a_i \cos(x \frac{i\pi}{L}) + b_i \sin(x \frac{i\pi}{L})) \right] dx$$  \hspace{1cm} (2.20)

where $C_L = 2L \sum_{i=0}^{\infty} (a_i + b_i)$. Clearly, the final form of the Eq. 2.25 is composed of two integration terms. For both terms, we make use of the method integration by parts.

$$\int_{-L}^{L} x a_i \cos(x \frac{i\pi}{L}) dx = [uv - \int v du]_{-L}^{L}$$  \hspace{1cm} (2.21)

where $u = x, du = dx$ and $v = \sin(x \frac{i\pi}{L})/(\frac{i\pi}{L})$.

$$\int_{-L}^{L} x a_i \cos(x \frac{i\pi}{L}) dx = \left[ \frac{x \sin(x \frac{i\pi}{L})}{(\frac{i\pi}{L})} + \frac{\cos(x \frac{i\pi}{L})}{(\frac{i\pi}{L})} \right]_{-L}^{L} = 0$$  \hspace{1cm} (2.22)

Following the similar procedure, we obtain the following equation for the second term,
2.1 AP Selection in UCN: A Network Performance History Based Approach

\[
\int_{-L}^{L} x(b_i \sin(x \frac{i\pi}{L})) dx = -L^2 \frac{2\cos(i\pi)}{\pi i} \tag{2.23}
\]

Finally, we obtain the expected value as:

\[
E\{\tilde{x}\} = \frac{L}{\sum_{i=0}^{\infty} (a_i + b_i)} \sum_{i=0}^{\infty} (-b_i \frac{\cos(i\pi)}{\pi i}) \tag{2.24}
\]

We write the variance as \(Var\{\tilde{x}\} = E\{\tilde{x}^2\} - E\{\tilde{x}\}.\) Having the \(E\{\tilde{x}\}\), we calculate the term \(E\{\tilde{x}^2\}\), which is

\[
\frac{1}{\sum_{i=0}^{\infty} (a_i + b_i)} \sum_{i=0}^{\infty} \int_{-L}^{L} x^2(a_i \cos(x \frac{i\pi}{L}) dx + \int_{-L}^{L} x^2 b_i \sin(x \frac{i\pi}{L}) dx \tag{2.25}
\]

Once again we make use of integration by parts for both terms and obtain,

\[
\int_{-L}^{L} x^2(a_i \cos(x \frac{i\pi}{L}) dx = \frac{x^2 \sin(x \frac{i\pi}{L})}{(\frac{i\pi}{L})^2} - 2[-x \cos(x \frac{i\pi}{L}) + \sin(x \frac{i\pi}{L}) \frac{-L^2}{(\frac{i\pi}{L})^2}] |_{-L}^{L} = L^4 \frac{4\cos(i\pi)}{\pi i^2} \tag{2.26}
\]

\[
\int_{-L}^{L} x^2(b_i \sin(x \frac{i\pi}{L}) dx = \frac{-x^2 \cos(x \frac{i\pi}{L})}{(\frac{i\pi}{L})^2} + 2[x \sin(x \frac{i\pi}{L}) \frac{-L^2}{(\frac{i\pi}{L})^2} + \cos(x \frac{i\pi}{L}) \frac{-L^2}{(\frac{i\pi}{L})^2}] |_{-L}^{L} = 0
\]

Thus the term \(E\{\tilde{x}^2\} = \sum_{i=0}^{\infty} (a_i + b_i) \sum_{i=0}^{\infty} (a_i \frac{4\cos(i\pi)}{\pi i^2}).\) Finally the variance is:

\[
Var\{\tilde{x}\} = \frac{L}{\sum_{i=0}^{\infty} (a_i + b_i)} \sum_{i=0}^{\infty} \left(2La_i \frac{4\cos(i\pi)}{\pi i^2} - b_i \frac{\cos(i\pi)}{\pi i}\right) \tag{2.27}
\]

2.1.4.3 Decision Making

In the decision making phase, the UCN-APs are classified either as attractive or unattractive from a UCN-client’s point of view and the main objective of this stage is to provide how likely a specific UCN-AP is attractive for the UCN-client. With the attractive, we mean that a UCN-AP provides acceptable performance depending on the traffic behavior of a client.
2 End User Cooperation for the Facilitation of Load Balancing in UCN

<table>
<thead>
<tr>
<th>Traffic Category</th>
<th>Related non-tolerable Critical Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>connection refusal (cr)</td>
</tr>
<tr>
<td>Best Effort</td>
<td>packet loss, migration instances, connection refusal</td>
</tr>
<tr>
<td>Video</td>
<td>delay time, migration instances, cr</td>
</tr>
<tr>
<td>Voice</td>
<td>delay time, jitter, packet loss, migration instances, cr</td>
</tr>
</tbody>
</table>

Table 2.2: Dominant Critical Observation vs. Related Traffic Categories

Various efficient monitoring systems are supported by different mobile platforms [47, 48] for the traffic behavior monitoring of client entities. A feasible integration of these tools into the proposed solution enables a deduction of the client tendency in terms of highly utilized traffic categories. In accordance with the AP classification term, the definition of attractiveness changes based on the substantially utilized access categories on a client entity. In other words, the dominance of the proposed network performance indicators differ in the decision-making stage. As shown in Table 2.2 we list the dominant critical performance indicators based on the access categories, which helps for the construction of the AP selection strategy. Consequently, we propose an AP selection strategy based on three main parameters, namely, the traffic behavior of client, the SNR values and finally the learning engine based client belief on the network performance of the UCN-AP, which mainly is a function of SNR values as proposed in the preceding subsection.

![Figure 2.7: Transitions between Belief States based on pdf Parameters](image)

Referring to Figure 2.7, we model the world of a specific UCN-AP in two belief-states, as the perception of the UCN-client. We define the states as $S_i$ and $S_j$ as the generic attractive and unattractive states for different access categories and for which the state transitions are modeled with exponential processes (continuous time Markov Chains). $\eta_{ij}$ and $\eta_{ji}$ stand for the transition parameters due to the critical and non-critical observations. Defining the random variables $\tilde{C}$ and $n\tilde{C}$, the probability distributions of which are obtained with the learning process described in Section 2.1.4.2.1, $\eta_{ij} = E\{\tilde{C}\}$ and $\eta_{ji} = E\{n\tilde{C}\}$. Additionally, the state transition parameter $\zeta$ is defined as a function of precisions of the monitored random variables,
2.1 AP Selection in UCN: A Network Performance History Based Approach

i.e. the $\tilde{C}$ and $n\tilde{C}$. As the precision describes how certain further draws of the random variable varies, this additional state transition parameter is incumbent upon increasing the randomization in AP selection decision in case of a lower precisions, i.e. higher variance. We model a weighted sum as the combination and assign the same weights so that the precisions of the both random variables contribute to the decision making in a fair manner. Thus we define a new random variable $\tilde{C}_c = \tilde{C} \ast 0.5 + n\tilde{C} \ast 0.5$. For this variable, we define a sigmoidal function of the corresponding precision of $\tilde{C}_c$ in order to i) keep the suppression of outcome with two pre-defined thresholds from top and down, which guarantees that the outcome is compatible with the $\eta_{ij}$ and $\eta_{ji}$ parameters and i) provide a graceful flexibility in multiple regions. Defining the precision as the reciprocal of the standard deviation,

$$\hat{P} = \frac{1}{\sqrt{Var{\tilde{C}_c}}} = \frac{2}{\sqrt{Var{\tilde{C}}} + Var{n\tilde{C}}}$$

(as these random variables are independent), the

$$\zeta = (\eta_{ij} + \eta_{ji})i + \frac{\kappa}{1 + e^{(A_P)(T_P - \hat{P})}}$$

where the lower and upper limits are $(\eta_{ij} + \eta_{ji})i$, $(\eta_{ij} + \eta_{ji})(\kappa + i)$. The $A_P$ defines how smoothly the $\zeta$ characteristic changes around threshold $T_P$.

The state transition parameters are interpreted as the parameters for the occurrence rate of special events, e.g., arrivals. The steady state probability distribution vector $\pi = (\pi_i \quad \pi_j)$ over the states $S_i$ and $S_j$, defines how likely is the UCN-AP is attractive or vice versa. The generator matrix

$$D = \begin{pmatrix} -(\zeta + \eta_{ij}) & (\zeta + \eta_{ij}) \\ (\zeta + \eta_{ji}) & -(\zeta + \eta_{ji}) \end{pmatrix}$$

defines the state transitions, where $\pi D = 0$ for the steady state. As the $\pi_i + \pi_j = 1$,

$$\pi_i = \frac{(\zeta + \eta_{ji})}{(\zeta + \eta_{ji}) + (\zeta + \eta_{ij})}$$
and

\[ \pi_j = \frac{(\zeta + \eta_{ij})}{(\zeta + \eta_{ji}) + (\zeta + \eta_{ij})} \]  

(2.32)

Under these circumstances, the open form for these probabilities are:

\[ \pi_i = \frac{(\eta_{ij} + \eta_{ji})(t + \frac{\kappa}{1 + e^{\tilde{C}_P(T_{\tilde{P}} - \tilde{P})}}) + \eta_{ji}}{2(\eta_{ij} + \eta_{ji})(t + \frac{\kappa}{1 + e^{\tilde{C}_P(T_{\tilde{P}} - \tilde{P})} + 1}) + \eta_{ij} + \eta_{ji}} \]

(2.33)

\[ \pi_j = 1 - \pi_i \]

in order to provide a better understanding of the proposed method for the belief-states distributions, we draw the \( \pi_i \) vs. variance for different \( \eta_{ij} + \eta_{ji} \) and constant \( \nu, \kappa \) values.

![Figure 2.8: Steady-State Probability Distribution for "Unattractive" State with Changing Variance and Transition Parameter](image)

As illustrated in Figure 2.8, the proposed approach normalizes the steady-state distribution in case of lower precisions, i.e., in case the UCN-AP network performance characteristic is highly dynamic. For the proposed active measurement taking system, each client has the opportunity in the range of 30 times per hour. The \( \tilde{C} \) and \( n\tilde{C} \) are the complementary random variables, i.e. \( \tilde{C} + n\tilde{C} \) is in the range of 30. Thus we choose \( T_{\tilde{P}} \approx 15 \) where the \( \zeta \) becomes comparable much higher than the \( \eta_{ij} \) and \( \eta_{ji} \) values and dominate the distribution. In other words for the higher variance values, the proposed solution dictates an uncertainty in the AP selection decision.

The set \( \Delta_{x/SNR} = \{\delta_{bg}, \delta_{be}, \delta_{vi}, \delta_{vo}\} \) defines the percentage of access categories. The \( \Pi_{x/SNR} = \{\pi_{ad}, \pi_{aj}, \pi_{pl}, \pi_{cr}, \pi_{mi}\} \) is the set of steady-state probability distribu-
2.1 AP Selection in UCN: A Network Performance History Based Approach

Contributions of unattractive states for average delay time, average jitter, average packet loss, connection refusal, migration instances respectively for a specific UCN-AP with an id of \( x \), which is based on the measured SNR values. The total attractiveness \( A_{x/SNR} \) definition is straightforward and is given by:

\[
A_{x/SNR} = \delta_{bg} (1 - \pi_{cr}) + \\
\delta_{be} (1 - \pi_{pl}) (1 - \pi_{mi}) (1 - \pi_{cr}) + \\
\delta_{vo} (1 - \pi_{ad}) (1 - \pi_{mi}) (1 - \pi_{cr}) + \\
\delta_{vo} (1 - \pi_{ad}) (1 - \pi_{mi}) (1 - \pi_{aj}) (1 - \pi_{cr}) (1 - \pi_{pl})
\]  

Finally the UCN-AP selection decision in a set of UCN-APs \( X_{SNR} \) is based on a constructed probability distributions over the attractiveness of an AP under a specific SNR value. The AP selection is done in a probabilistic way and the probability of an AP to be selected is given by

\[
Pr_{x/SNR} = \frac{A_{x/SNR}}{\sum_{y \in X_{SNR}} A_{y/SNR}}
\]  

2.1.5 Experiments and Test Results

In this section we detail the methodology and the test scenarios for the evaluation of the proposed model. We implement a minimal Java based client network manager (CNM) software including the proposed framework as an additional component. Via testbed experiments for different network scenarios, the validity of our proposal is investigated, the technical details of which are presented in the proceeding subsections.

2.1.5.1 Brief Introduction to Testbed

In this study, an elementary wireless test-bed as a part of [49] is utilized, the architecture of which is briefly illustrated in Figure 2.9. Two APs are configured as UCN-AP running a load balancing algorithm, which is briefly introduced in the scenario description and the details of which can be found in [50]. Both APs run Voyage-Linux [51] as the AP operating system (OS). This Debian based minimal OS is tuned specifically for the wireless networking. We make use of PCEngine-Alix boards [52] and equip the APs with DNMA-92 Wi-Fi modules [53]. The central router provides access to the application servers and to the Internet for the UCN-clients, whereas the control&monitor unit is utilized for the configuration and experiment maintenance.
2.1.5.2 Scenario Description

The experiments are conducted based on the load balancing problem described in 
[50]. The main complexity is the potentiality for the formation of congestion areas 
among UCN sub-communities, which is due to the reasonings that i) the UCN-APs 
may have revenue (incentives gained from UCN-clients, e.g. virtual currency [11]) 
or load oriented strategies, ii) wireless capabilities, i.e. the provided QoS, differs 
enormously due to the diversity of participated user entities with UCN-AP modes. 
In our case, the UCN-APs run the software proposed in [50], where the load is 
balanced by means of AP collaborations. In case of mutual agreements, the APs 
may send a migration command to its resident clients in order to initiate a handover 
to another available AP.

Under these circumstances, our main motivation is to monitor effectiveness of 
the proposed model in characterizing the experienced UCN-APs by means of pre-
viously described performance indicators. We additionally track the distribution of 
network clients among UCN-APs via proposed AP selection approach for iterative 
coalescences.

2.1.5.2.1 Experiment 1 In this experiment, we degrade the provided QoS and 
congest a single UCN-AP unevenly from time to time using network testing tools 
D-ITG [54], IPERF [55] and TC [56]. We automize this process with an additional 
software, having a control over the utilization of these tools. We provide two different 
probability distribution functions (PDF) as inputs to this software. The monitored 
AP is congested from time to time based on these probability distribution functions. 
The AP is exposure to a high load for a random duration, the amount of which 
is drawn based on the first PDF, which is illustrated by Figure 2.10-b. Similar to 
the the previous cycle, the load on the AP is relaxed for another additional random
duration based on the PDF shown by Figure 2.10-a. As the expected value of the loaded duration is quite larger than the one for a relaxation duration, the AP is mostly loaded and provides critical network performances during the most of the operation time.

On the client device the measured SNR value alternates between 30 dB and 35 dB, where the RSSI fluctuates between -57 dBm / -62 dBm and the noise power is around -92 dBm. Although the average delay time, jitter and bit-rate are mostly critical during the live time, no packet loss is observed, which is potentially due to the fine SNR value.

It should be highlighted that we emulate very dynamically alternating AP operation states, which may be interpreted as an extreme situation in real network operations. With the states, we mean under-loaded and critically over-loaded operational conditions. As illustrated in Figure 2.10, the estimated PDFs are highly correlated with the real PDFs on state durations. Additionally, it is observed that the proposed model converges in a short number of active measurement sessions for the input PDFs (Figure 2.10-a and -b). Nevertheless the proposed solution suffers in negligible amount of modeling cycles due to the sampling rate, i.e. the network test requests rate, which has a great impact on catching the real system dynamics of the observed AP, and the extremely fluctuating AP load state. This system dynamic is not included in the previously described active measurement system objective. Nevertheless it motivates us for a potential extension for this study.

Finally, the attractiveness of the AP from different UCN-client’s perspectives are illustrated in Figure 2.11. With the perspective term, we mean the differing percentages in the utilization of various traffic categories (TC) depending on client network activities. As depicted in Figure 2.11, the AP is attractive for the UCN-clients who are mostly interested in applications based on background or best-effort categories. Nevertheless, the modeled AP is not an attractive AP for the wireless multimedia (WMM) activities.

2.1.5.2.2 Experiment 2 In this experiment, we investigate the facilitation of load balancing by UCN-clients via the proposed AP selection model. We include the second UCN-AP to the network scenario and run previously described congestion creating software, which is fed with the previously described input PDFs in an opposite way, i.e. the AP is congested based on the PDF illustrated in Figure 2.10-a instead of 2.10-b. In other words, the second UCN-AP is mainly operating under less loaded conditions.

We install our CNM software including proposed framework engines on to 4 PCs
Figure 2.10: Estimations on the pdf of Time Durations for Different Performance Indicators
2.1 AP Selection in UCN: A Network Performance History Based Approach

Figure 2.11: Steady State Probabilities of Critical States and AP Attractiveness based on different TC Percentages

and form various iterative coalescences. Our main objective is to observe the distribution among UCN-clients among two UCN-APs for future iterations. Initially, we assume a homogenous distribution among traffic categories among UCN-clients, i.e. the clients are assumed to generate 25% background, 25% best effort, 25% video and 25% voice traffic. In Figure 2.12, we illustrate the attractiveness of UCN-APs vs. system time among 4 UCN-clients. The red-line stands for the attractiveness of the UCN-AP operating under better load conditions, whereas the blue-line represents attractiveness for the critically loaded UCN-AP. We observe that the attractiveness definition of these APs among clients converge in a short time and is around 0.3, 0.2 respectively. In other words the APs are selected with the probabilities of 0.6 and 0.4 respectively, which has been observed throughout the experiment as well.

Finally, we change the percentages of the traffic categories for a specific UCN-client and set the voice traffic percentage to a higher level. We run the experiment once again by resetting the clients-beliefs. The respective attractiveness definition of
the UCN-APs for this specific client is drawn in Figure 2.13. The attractiveness of the UCN-APs are around 0.6, 0.3, which indicates that these APs are selected with the probabilities of 0.66 and 0.34 among UCN-clients preferring more voice traffic.

2.1.6 The Evaluation of the Proposed AP Selection Framework

In this section, we provide a learning framework for the AP selection problem in UCN, where the focus remains on using the past experiences in terms of network
2.1 AP Selection in UCN: A Network Performance History Based Approach

Figure 2.13: UCN-APs attractivenesses for a specific UCN-Client preferring voice traffic

performance indicators. It should be highlighted that one of the main motivations of this framework is to introduce the end-user facilitation for load balancing by means of past experiences-aware AP&client sub-coalescences. To some extent, the proposed solution has similar backbone of a reputation based incentive mechanism in UCN [11]. That is to say, the UCN members keep also how well a UCN subnetwork provides network quality, similar to their willingness to cooperate in a distributed reputation based incentive mechanism.

As the decision instance takes place on the client entities, one may claim that the experienced quality of services, i.e. the QoE, may be superior in comparison to the QoS based parameters. This, on the other hand, may bring about the problem of subjective critics on the performance of a UCN subnetwork, for which the coalesce is constructed and maintained by specific UCN-APs. Besides, a fair judgement in a distributed manner may increase the complexity in making decisions. Thus we focus on more common objective and transparent measurements in this study as detailed before. The main motivation in selecting these specific measurements is not only the load balancing but also helping for the latency in finding a better UCN-AP and client match, e.g., a higher expected rate of connection refusal indicates that the subnetwork constructed by the corresponding UCN-AP may not provide enough resources and the UCN-client may prevent latency for additional negotiation phases. Nevertheless, we believe that the proposed model may be enriched with various additional QoS or QoE metrics.

In our solution, we mainly focus on the UCN dynamics. We believe, however, that the proposed methodology may be extended for various IEEE802.11 networks. A good example for the application of this proposal can be the campus and company networks, where the iterative coalescences of specific clients and specific stationary APs are possible.

We believe that the proposed decision-making has a great impact on the user
satisfaction, which is one of the main factors for the maintenance of the UCN. In this work, our approach provides mainly the UCN-clients with the adaptation ability against fluctuations in the regular behavior of the UCN environment in terms of the aforementioned performance related measurements. The remaining AP selection strategies in UCN, e.g. the economical factors, such as total local credit, negotiated unit resource costs and etc., are not yet discussed in this thesis. A detailed work on the analysis of additional revenue oriented AP selection is left. Nevertheless, we believe that the these aspects and user preferences may also be worth considering as an improvement.

In this section, the AP selection is restricted to previously discussed network performance indicators. One another tackle in the UCN is the short term availability of UCN-APs in a UCN environment, which may stem from either the mobility behavior of the UCN-APs or the user preferences. In other words, a specific client and AP tuple may be relatively transient to each other due to i) mobile AP, ii) short term AP functionality of a UCN member due to owner preferences or finally iii) mobile client. The system complexity drastically increases in the UCN environment due to this reasoning. Hence such a setting lays out a wireless landscape where the clients and the providers may have short term interactions. This in turn brings about a great impact on AP selection decision as the connectivity of the client is at high risk under these circumstances. In the next section, we focus on the critical network performance related UCN dynamic due to UCN environmental settings, which we call relative transiency problem. With the relative transiency we mean the transient or short term AP functionalities in the UCN, which is critical in terms of connectivity.

2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

With the advent of revolutionary wireless technologies and equipments, one may expect a great user mobility. In a UCN environment, both the UCN-AP and the UCN-client can be mobile or stationary. In addition to the potential connection losses and thus worse UCN experience, the stochastic departure of providers from the UCN environment may cause fluctuations in the instant formation of the UCN. With the instant formation, we basically focus on the load distributions among the available APs. In other words, such an instance may risk the distributed load among the UCN and cause additional migration instances or connection refusals. Thus in the UCN, i.e. the envisioned immensely dynamic telecommunication landscape,
relying on the real-time measurements for decision-making may consequence in sub-optimal systems.

We believe that a learning mechanism for the level of relative transiencies at a specific location would i) provide the UCN-clients with the tendency of selecting relatively longer-term or static UCN-APs and secondly, ii) reduce the turbulences among UCN environment due to client migrations once shorter-term UCN-APs leave the UCN-environment. Additionally, a prediction on the next location of a UCN-client may improve the handovers and the resource negotiations in case there exists an expected location, where the UCN member is regularly stationary for a longer time. In this connection, we propose a learning mechanism for the UCN-clients based on past experiences with various WLAN APs. This learning mechanism defines the level of client transiency w.r.t APs on the regular mobility paths.

In this section, we study the mobility of users in user centric networking (UCN) environment. We propose the mobility prediction approach by WLAN group formation. We focus on mobility behavior of users and propose the WLAN group prediction approach. We base the proposed approach on the relatively simple parameters and evaluate the performance of our proposed concept by real implementation. Our approach is based on building the history of encountered WLAN groups, modeling an encounter instance of a new group as arrivals for the proposed model and finally adapting the prediction. In the granular level, the arrival rates are estimated using expectation maximization algorithm. We believe that the proposed approach is important for UCN like environments, where information on the hardware or software capabilities, available network resources, instant AP load and etc are not available for the end-user. We focus on the mobility behavior awareness of the UCN stakeholders based on history. The proposed solution can be generalized for the company or campus wireless networks.

The rest of this section is organized as follows: Section 2.2.1 provides a brief literature review on the mobility prediction in wireless networks. In Section 2.2.2, we evaluate the state of the art solutions from the UCN perspective. We state our approach in the proceeding section, namely the Section 2.2.3. The technical information on the proposed framework is given in Section 2.2.4. We discuss on the validation of the proposed model in section 2.2.5, where we present the experiment results. Next, we evaluate our model in Section 2.2.6.

2.2.1 State of the Art

Doss et al. [57], classifies the mobility prediction methodologies based on i) the physical patterns such as velocity and positional coordinates, ii) using Mobility His-
tory Bases (MHB) including previous movement patterns and finally iii) stochastic models. Senthilkumar et al. [58] and Meghanathan [59] propose similar location based mobility prediction for the MANET. The proposal covers least expiry time for each link between nodes. Pack et al. [60] propose a fast handoff mechanism using mobility prediction, where the mobility of clients are estimated based on handoff ratios and residence time of a mobile host in an AP. In [61] a hidden Markov model based mobility prediction methodology is proposed in order to develop a cluster based cooperative caching technique.

Based on the users’ non-random regular mobility behavior and using GPS location patterns, Su et al. [62], propose a mobility prediction algorithm in predicting future network topology of the MANET. The position information for each node is added to the data packets and the location of each node with respect to any other node is predicted accordingly. As a competitor to the GPS technology, [63] proposes a methodology using GSM data in detecting user mobility patterns. The mobility detection software collects signal strength values, cell IDs and channel numbers of nearby cells. In [64], an auto-regressive hello protocol is proposed for the MANET, where the next position of each node is calculated with the auto-regressive model and the GPS data. Similary, Hemmes et al. [65], proposes a mobility prediction technique using the physical movement and velocity parameters for vehicular networks. Wanalertlak et al. [66] propose a mobility prediction mechanism for a behavior-based handover using the location information in addition to take the client characteristics into account in terms of short-term and periodic behavior.

In a stochastic mobility modeling approach, Torkestani et al. [67] propose a mobility prediction scheme equipped with adaptive learning automata mechanism using Gauss-Markov model. The proposed algorithm takes the single input mobility history and prediction of mobility patterns such as speed, direction and randomness degree is the output of the proposed model.

2.2.2 Evaluation of the Literature from the UCN Perspective

Various studies rely either on the physical parameters based on geographical patterns acquired from GPS or using triangulation methods for GSM (or under the coverage of a specific WLAN). The variety in hardware or software capabilities for the UCN clients raises the difficulty for the utilization of GPS or GSM triangulation based techniques. This is due to the fact that these techniques are commonly restricted to mobile phones and are not applicable for various mobile computers, e.g. notebooks. The remaining Wi-Fi based triangulation method may not be sufficient for covering the mobility behavior of a UCN-client in a global manner as such an approach is
2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

applicable mainly for indoor applications, i.e. for short ranges and generally requires the complete information on the APs. Therefore, a smarter and feasible manner is required in order to handle these tackles in UCN.

It should be highlighted that aforementioned techniques provide real time information on the mobility behavior of the monitored client. As a second type mobility analysis approach, the next item prediction on the mobility path of a mobile entity is extensively studied in the Literature. These attempts have commonly the basic assumption of consistency in mobility behavior of clients. That is to say, that the mobility behavior is mainly analyzed in a single state. Nevertheless, the mobility behavior can change based on week days, holidays or can even be fully stochastic without showing any regular pattern. We believe that a solution with a deeper analysis capability is required catching these states of the UCN-clients, such as week days, regular holidays, e.g. weekend, or fully stochastic domains.

2.2.3 Our Approach

The UCN community is consisted of a wide variety of Wi-Fi enabled devices with AP capability, where most of these devices are not equipped with GPS modules. In other words the geographical location information and mobility parameters such as velocity, acceleration and position information are mostly not available. Additionally, as the exchange of personal interests of UCN community is performed on the wireless medium, we study the mobility parameters of a UCN member in terms of the encountered wireless APs.

When it comes to the another dimension in the UCN, i.e. the missing data in case of high instability in the RSSI levels of the APs in realistic environments or even disappearance of particular APs on the mobility path of a UCN-client due to AP-owner preferences: One needs to have a deeper look into the question how one handles missing data in an intelligent way? Alternatively, we investigate monitoring and tracking only the groups of encountered APs on the route of the mobile client rather than determining exact relative position. To this end, we perform an experiment with our Android application in order to examine groups of APs around a mobile client. In this experiment, the client is mobile for 3 hours and 47 minutes. The client changes his physical mobility patterns from time to time as a pedestrian, and a passenger of an overground or an underground vehicle. As illustrated in Figure 2.14, 754 APs are encountered during this time duration, the APs are observed in groups.

It is deduced from the experiment that except for autobahn and underground (e.g., from A to B) permanent existence of AP groups are encountered on the mobility
path. Certain groups are observed more frequently and longer once the client is pedestrian or roams around a specific area. Besides, certain set of APs are observed continuously when the client is stationary (e.g. at position F).

We define the mobility parameters of the UCN-clients in terms of the encountered AP groups rather than single APs. Due to the fluctuations in the RSSIs, selecting a specific AP among multiple APs requires additional decision making stages. Additionally, the unstable RSSIs caused by physical wireless medium parameters such as location of the AP, the antenna characteristics, signal interferences, and etc., raises the error rate causing instability on the mobility path AP chains. In other words, choosing different APs for the multiple iteration of presumed decision-making algorithm at the same physical position might result in difficulties for the upper level decision instances, e.g. mobility or relative transiency prediction. Additionally, in a realistic scenarios due to private AP owner preferences or for the UCN, due to mobile APs, the presence of an AP at a specific geographic position might not be guaranteed.

For a better understanding and interpretation of the proposed concept, an analogical illustration to mobility tracking experiment is provided in Figure 2.15.

Having defined the mobility model and corresponding patterns, we focus additionally on the analysis of the various states for the relative transiencies, hence the mobility, of the UCN-clients. To this end we define three states which handle regular paths, stochastic routines and the fully stochastic domains for the mobility and the transient, medium-term or long-term availability of UCN-AP and UCN-client interactions for the relative transiency. The technical details and the corresponding explanations on these states are provided in the next Section.
2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

2.2.4 Technical Details of the Proposed Model

In this section, we detail the learning mechanism which is stated in section 2.2. The proposed solution comprises of two sub-blocks, firstly the AP clustering and corresponding transiency of the encountered APs and secondly the mobility prediction. In the following sections, detailed information on these sub-blocks are provided.

2.2.4.1 AP Clustering and Relative Transiency

We use two criterions to form an AP cluster including a UCN-AP. We focus on i) the expected appearance frequency on the mobility path and ii) the expected appearance duration of an AP cluster (the relative transiency).

2.2.4.1.1 The expected appearance frequency  As illustrated in Figure 2.16, we analyze the appearance frequency of each AP cluster in three graded states, namely, fully stochastic \((\mu_R, \tau_R)\), stochastic routine\((\mu_N, \tau_N)\) and regular path \((\mu_F, \tau_F)\), where \(\tau\)’s represent the probability distributions and \(\mu\)’s stands for the expected appearance frequency at the corresponding state. A regular path state defines frequent appearance of the AP cluster. A stochastic routine state defines less frequently appearing AP cluster due to changing on longer time-periods (e.g., weekend) activities. Finally, the fully stochastic state stands for rarely appearing AP clusters on the mobility path of the client.

With these settings, we construct a clustering problem, where the number of cluster is restricted to three. We believe that the previously suggested EM approach
End User Cooperation for the Facilitation of Load Balancing in UCN

Figure 2.16: Three State Appearance Frequency with State Parameters

... can provide a better analysis for the constructed system settings. Nevertheless we relax the model this time and restrict the model based on only three states as the main motivation of the proposed framework is to find out the routines of the matter of subject. In other words, as the number of states for the model of subject is given beforehand, i.e., the number of mixture components are known, we make use of a Poisson Mixture with three components.

It should be highlighted that each state parameter analysis is performed specific to each AP group. On the regular mobility paths the probability of $k$ times appearance of an AP cluster in a pre-defined time period, given the set $\Theta = \{(\mu_R, \tau_R), (\mu_N, \tau_N), (\mu_F, \tau_F)\}$ and $S$ representing the states:

$$P\{k|\Theta\} = \sum_{S \in \{R,N,F\}} \tau_S \frac{e^{-\mu_S} (\mu_S)^k}{k!}$$  (2.36)

The environmental states for the hidden variables of EM model is represented with the set $Z = \{z_R, z_N, z_F\}$, where the probability distribution of hidden variables are $\tau_R, \tau_N, \tau_F$ respectively. Tracking the appearance counter vector $K = \langle k_1, k_2, \ldots, k_n \rangle$ and given $z_j \in Z$:

$$Pr\{K, z_j | \Theta\} = \prod_{i=1}^{n} [\tau_j (\frac{e^{-\mu_j} \mu_j^{n_i}}{n_i!})]$$  (2.37)

As commonly practiced in EM algorithm, taking the log of we get,

$$log\{Pr(K, z_j | \Theta)\} = \sum_{i=1}^{n} [\log\{\tau_j\} - \mu_j + n_i \log\{\mu_j\} - \log\{n_i!\}]$$  (2.38)

Now using Bayes theorem:

$$Pr\{z_j|k_i, \Theta\} = \frac{\tau_j e^{-\mu_j} \mu_j^{k_i}}{\tau_R e^{-\mu_R} \mu_R^{k_i} + \tau_N e^{-\mu_N} \mu_N^{k_i} + \tau_F e^{-\mu_F} \mu_F^{k_i}}$$  (2.39)
2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

referring to the results Eq. 2.36 - 2.39, the **E-Step** is:

\[ E_{Z|K;\Theta} = \sum_{i=1}^{n} \sum_{z_j \in Z} Pr\{z_j | k_i, \Theta\} log\{ Pr(K, z_j | \Theta) \} \] (2.40)

under the constraint \( \tau_R + \tau_N + \tau_F = 1 \) and utilizing \( n^{th} \) estimated set of \( \Theta_{\mu(n)} = (\mu_R^{(n)}, \mu_N^{(n)}, \mu_F^{(n)}) \), the \((n + 1)^{th}\) estimation of probability distribution over the environmental sets as the start point of **M-Step** is given:

\[ \tau_S^{(n+1)} = \arg \max_{(\tau_S)} E_{Z|K;\Theta_{\mu(n)}} (\tau_S) \] (2.41)

\[ \tau_S^{(n+1)} = \arg \max_{(\tau_S)} E_{Z|K;\Theta_{\mu(n)}} (\tau_S) \] (2.42)

\[ \tau_N^{(n+1)} = \arg \max_{(\tau_N)} E_{Z|K;\Theta_{\mu(n)}} (\tau_N) \] (2.43)

\[ \tau_F^{(n+1)} = \arg \max_{(\tau_F)} E_{Z|K;\Theta_{\mu(n)}} (\tau_F) \] (2.44)

Using Eq. 2.42, \((n + 1)^{th}\) arrival rates of corresponding environmental states is:

\[ \mu_n^{(n+1)} = \arg \max_{(\mu_S)} E_{Z|K;\Theta} (\mu_S) \] (2.45)

\( \forall S \in \{R, N, F\} \).

\[ \mu_n^{(n+1)} = \arg \max_{(\mu_S)} E_{Z|K;\Theta} (\mu_S) \] (2.46)

\[ \mu_n^{(n+1)} = \arg \max_{(\mu_N)} E_{Z|K;\Theta} (\mu_N) \] (2.47)

\[ \mu_n^{(n+1)} = \arg \max_{(\mu_F)} E_{Z|K;\Theta} (\mu_F) \] (2.48)

The AP clusters with higher probabilities for stochastic routine or regular path states are forwarded to an upper level mechanism, namely the mobility prediction and expected appearance duration estimation, which are detailed in following section.

**2.2.4.1.2 The expected appearance duration estimation:** Similar to the expected appearance frequency, the time duration, during the AP cluster is visible to the client, is analyzed in three states, namely, transient, medium-term and long-term. As illustrated in Figure 2.17, the state parameters (specific to each AP cluster) are
given as \((\upsilon_T, \rho_T), (\upsilon_M, \rho_M)\) and finally \((\upsilon_L, \rho_L)\), where \(\upsilon\)'s stand for the expected time duration and \(\rho\)'s stand for the state probabilities. The estimation of state parameters is performed with the similar EM construction process, which is detailed in the previous section. In the proceeding section, we comment on our mobility prediction method.

### 2.2.4.2 Mobility Prediction

In this part, the Markov Arrival Processes (MAP) are utilized in the proposed solution for the mobility prediction. Hence, we initially comment very briefly on the construction of Markov Arrival Processes in order to provide a better understanding on the proposed framework. The main motivation to use this theoretical model comes from the fact that it provides a suitable model for the formulation of the framework.

#### 2.2.4.2.1 Markov Arrival Processes

In continuous time Markov processes (CTMP), the state transitions of an irreducible continuous time Markov Chain (CTMC) are associated with special events having Poisson distribution with different parameters. In addition to the analysis of spontaneous transitions of states, MAPs define *arrivals* of special characters that are associated with the phases and result additional transitions from associated state to another. The *sojourn* time of the Markov Chain at a specific state depends on the spontaneous transitions of CTMC and the arrival instances. The versatility of modeling various stochastic systems and the maneuverability makes MAPs popular in different scientific branches \[68\]. In this part, we present a short introduction to Markovian Arrival Processes (MAPs). Detailed information on specific types of MAPs and mathematical formulations can be found in \[68, 69, 70, 71, 72\].

In continuous time Markov Chains (CTMC) model, the state transitions of an irreducible \(m\)-state continuous time Markov Chain (MC) are associated with special
2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

events having Poisson distribution with parameters \( q_{ij} \), where \( i \neq j \) and \( 1 \leq i \neq j \leq m \). The stochastic process \( \{I(t), t \geq 0\} \) (CTMC) represents the current state of the environment. Basically, the Markov Chain would stay at state \( i \) (i.e., \( I(t) = i \)) until the first event of other \( m - 1 \) Poisson processes occurs. The sojourn time of the Markov Chain at state \( i \) is also Poisson process with the parameter \( -q_{ii} = \sum_{1 \leq j \neq i \leq m} q_{ij} \).

The aforementioned stochastic process \( \{I(t), t \geq 0\} \), i.e., the \( m \)-state continuous time Markov Chain, has the infinitesimal generator matrix of \( mxm \) \( Q = (q_{ij}) \). The steady-state probability distribution vector \( \pi \) over \( m \) states satisfies:

\[
\pi Q = 0, \quad \pi e = 1, \quad e = (1)
\] (2.49)

In addition to the random events causing a state transition, Markovian Arrival Processes (MAPs) define arrivals of special characters that are associated with the phases and result additional transitions from associated state to another. Marked Markov Arrival Process (MMAP) is a generalization of previously proposed MAPs and counting processes like Poisson Processes, MMPP (Markov Modulated Poisson Processes) and BMAP (Batch Markov arrival Processes). MMAP models the stochastic arrivals of different types. In addition to the special event parameters \( \{d_{0,ij}, 0 \leq i \neq j \leq m\} \) causing transitions between states without arrivals, a finite set \( \Upsilon \) of different types of arrivals with parameters \( \{d_{v,ij}, 0 \leq i, j \leq m, \text{ and } v \in \Upsilon\} \) are defined. The stochastic process \( \{N_v(t), v \in \Upsilon, I(t), t \geq 0\} \) is a closed form of the MMAP model. \( N_v \) represents the counting process of the arrival \( v \in \Upsilon \). The stochastic process \( \{I(t), t \geq 0\} \) has the infinitesimal generator matrix \( D \) as given in Equation 2.50.

\[
D = D_0 + \sum_{x \in \Gamma} D_x
\] (2.50)

Next, we detail our prediction model based on MMAP construction.

2.2.4.2.2 Our Prediction Model Referring to the Figure 2.15, we focus on the likelihood of the next AP clusters on the route of the mobile device based on the current cluster, e.g. the likelihoods of route 1 to 2, 2 to 3 or 2 to 4. We model the appearance of each cluster as a unique world and form the corresponding CTMC, where AP clusters constitute the states for the chain. The mobilization of the client on different routes are modeled as the arrivals causing transitions in between states. The main purpose of the proposed framework is to determine the probability distribution of CTMC in the steady state, i.e., which AP cluster is most likely to be
encountered after expected ordinary appearance duration of the current cluster.

In this stage, we make use of the information provided by the previously discussed lower level EM based mechanism, which forwards the probability distributions over the frequency of appearance for a specific AP cluster. Thus the likelihoods of the next potential AP cluster in the mobility path are calculated accordingly. For instance if the current AP cluster belongs most likely to the regular path, the next \( N_{rp} \) regular path AP clusters construct the corresponding world states, where \( N_{rp} \) is the total number of AP clusters signed as regular path clusters. Similarly for the \( N_{fs} \) and \( N_{sr} \) represents the total number of AP clusters in fully stochastic and stochastic routine domains. For the sake of readability, we only refer to a specific set of AP clusters and provide our prediction method as follows.

Each AP cluster has an identity, however, for a better understanding of the proposed framework, we refer to the current cluster with the ID 1 and other cluster with numbers up to \( N+1 \) starting from 2 (in this case there exit \( N+1 \) number of clusters in the local repository of mobile device). The observation repository \( \Omega = \{o_{12}, o_{13}, \ldots, o_{1(N+1)}\} \) defines the movement of the client from current cluster to the any other cluster with the corresponding numbered ID. For the imaginary event, which defines the uncertainty level of the proposed model and causes transitions between each state, we define the \((N+1)x(N+1)\) matrix \( D_0 \) matrix as:

\[
D_0 = \begin{pmatrix}
-\lambda_1 & \lambda_{12} & \cdots & \lambda_{1(N+1)} \\
\lambda_{21} & -\lambda_2 & \cdots & \lambda_{2(N+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{(N+1)1} & \lambda_{(N+1)2} & \cdots & -\lambda_{(N+1)}
\end{pmatrix}
\] (2.51)

For the \( i \)’th arrival \( Ob_{i1} \), the \((N+1)x(N+1)\) matrix \( D_{Ob_{i1}} \) has a single non-zero term on the \( i \)th place of the first row as follows:

\[
D_{Ob_{i1}} = \begin{pmatrix}
0 & \cdots & \gamma_{Ob_{i1}} & \cdots & 0 \\
0 & 0 & \cdots & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & 0
\end{pmatrix}
\] (2.52)

where \( \gamma_{Ob_{i1}} \) defines expected rate of the \( i \)th arrival. The final generator matrix \( D \) is in the form:

\[
D = D_0 + \sum_{Ob_{i1} \in \Omega} D_{Ob_{i1}}
\] (2.53)
We model the uncertainty in the state transitions with homogenous distribution, i.e., \( \lambda_{1i} = \lambda \) for \( i \in 2, \ldots, (N + 1) \). Using this constraint and the Eq. 4.4, we derive the final minimal generator matrix as:

\[
D = \begin{pmatrix}
-N\lambda - \sum_{i} \gamma_{Ob_{1i}} & \lambda + \gamma_{Ob_{12}} & \cdots & \lambda + \gamma_{Ob_{1N}} \\
\lambda & -N\lambda & \cdots & \lambda \\
\vdots & \vdots & \ddots & \vdots \\
\lambda & \lambda & \cdots & -N\lambda 
\end{pmatrix}
\]

(2.54)

For the steady-state probability distributions \( \pi \), solving the equation \( \pi.D = 0 \) under the constraint \( \pi.(1 \ldots 1)^T = 1 \), we obtain:

\[
P\{\text{Current}\} = \frac{\lambda}{(N + 1)\lambda + \sum_{i=2}^{N} \gamma_{Ob_{1i}}} \]

(2.55)

\[
P\{2\} = \frac{(N + 1)\lambda + \gamma_{Ob_{12}} + \sum_{i=2}^{N} \gamma_{Ob_{1i}}}{(N + 1)\left\{(N + 1)\lambda + \sum_{i=2}^{N} \gamma_{Ob_{1i}}\right\}}
\]

(2.56)

\[
\vdots
\]

\[
P\{N\} = \frac{(N + 1)\lambda + \gamma_{Ob_{1N}} + \sum_{i=2}^{N} \gamma_{Ob_{1i}}}{(N + 1)\left\{(N + 1)\lambda + \sum_{i=2}^{N} \gamma_{Ob_{1i}}\right\}}
\]

(2.58)

In this mobility prediction model, the selection of \( \lambda \) variable is critical as it influences the dominance of arrival rates. A larger \( \lambda \) value makes the probabilities converge to \( 1/(N + 1) \) (a homogenous distribution among all clusters), which obviously maximizes the uncertainty level for the upper level decision making instances. Further discussion on the \( \lambda \) values is provided in Section 2.2.6. In the proceeding section, we comment on the experiments and test results for the evaluation of proposed model.

2.2.5 Experiments and Test Results

In this section we describe the evaluation methodology for the proposed solution. The proposed control framework is implemented for Android phones. Figure 2.18 illustrates the flow diagrams of the mobility behavior learning and prediction whereas the Figure 2.19 provides the flow-chart of AP group formation sub-block.

In order to evaluate the proposed model, we perform a mobility behavior modeling experiment for a client with an Android based mobile phone. This experiment is
conducted for the evaluation of mobility behavior sub-blocks in terms of i) AP group formation, ii) next AP group prediction and iii) relative AP group transiency prediction. The mobility behavior of the corresponding client is monitored for a week and during this time period, the prediction instances and the quasi static mobility durations are recorded. With the "quasi static" we mean either the mobile or static positioning of the client under the coverage of a specific AP cluster. The experiment results are illustrated by screenshots from the application in Figure 2.20 and 2.21 respectively, where the prediction instances are represented by red lines and the real client mobility is represented by green lines. The figure shows the ID of the encountered and predicted clusters on the y axis and the new iterations on the x axis, i.e., x axis provides the timing information.

Finally we provide example objects for cluster formation and the information model (EM related coefficients) for the appearance frequency of a cluster on the mobility path in Figure 2.22, which take 273 and 271 Bytes size on the disk respectively. A total number of 120 clusters together with different information models for the proposed solution takes 125 Kbytes, which is clearly negligible in terms of
2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

![Diagram of AP Group Formation](image1)

**Figure 2.19: AP Group Formation**

![Diagram of Next AP Group Prediction](image2)

**Figure 2.20: Next AP Group Prediction**

storage consumption. The evaluation of experiment results and the performance of the proposed solution is discussed in the next part.
2.2.6 Evaluation of the Proposed Framework

In this part, the experiments are conducted mostly in regions and buildings with a high number of different WLANs. Although proposed framework shows a remarkable success in predicting future AP groups on the regular AP group chains, it is observed that the framework suffers on the edge of groups due to fluctuations in the RSSIs of APs. An improvement in AP group formation by means of advanced clustering
2.2 Relative Transiency Problem and Mobility Behavior Modeling for UCN-clients

techniques can be considered as an extension for highly denser AP pools, though the solution provides yet a great inference on capturing the realistic behavior of a mobile client and the interactions between APs in terms of previously discussed transiency relation.

Secondly, the correlated mobility path chain formation of AP groups by means of direction detection may be considered as the vulnerable point of the proposed model for a state of art solution in mobility prediction. In other words, we believe that the decision of next AP group prediction should not only rely on the current AP group but also on a certain number of previously encountered AP groups, which would provide an improvement in the next AP group and the movement direction prediction. This problem is represented in Figure 2.23, where we depict the edge of routine chains and the positions, where the proposed model partly shows weakness.

![Figure 2.23: Regular Chain Formation and Intersection Points](image)

It is heuristically observed that clients represent a highly stochastic behavior in terms of the duration in their stationary position, i.e., where a client remains in the same AP group. Nevertheless the proposed model succeeds most of the time in detecting transient clusters with a tolerable error rate. As the transient AP groups on the mobility path of the client impacts immensely on the connectivity maintenance, which is related to AP selection problem in UCN, the proposed solution may help for eliminating those APs in a transient AP group.

In this part, the inference of the designer in the certainty level of proposed model is represented by \( \lambda \) parameters (ref. to Section 2.2.4.2). Depending on the frequency of coherence between framework estimations and the real mobility behavior, the dominance of arrivals and the uncertainty coefficient \( \lambda \) should be adjusted accordingly. In other words, the degree of certainty in the estimation of probability distributions requires a smaller \( \lambda \) values in case of more frequent routine paths, i.e. higher level
of regularity in the mobility behavior, whereas a contrary situation necessitates a choice of larger $\lambda$ values.

Last but not least, we believe that the proposed model helps for the additional dynamics of UCN such as i) the negotiation phase in terms of more realistic amount of resource utilization and crediting, and ii) a fair reputation of UCN members. Considering about one of the core mechanisms in a UCN, i.e., the negotiation phases, once a UCN-client is willing to have connectivity through UCN-AP, one would agree on the profitability of pre-knowledge on the duration of service availability or service request. The level of transiency in a specific location has an impact on the results of a successful negotiation as it might cause inessential credit losses and hence may result in an unfair reputation decision. This property of UCN dynamics has an impact on either the trust / reputation metrics of stakeholders or the resource negotiation phases.

The captured behavior model is used as an additional observation for the previously proposed AP selection framework. We describe in the next section how the mobility and relative transiency awareness is adapted to the proposed AP selection method as an additional realistic capture of the UCN stakeholders’ behavior.

### 2.3 Relative Transiency and Mobility Awareness for the AP Selection Problem in UCN

Referring to the Section 2.1.4.3, where the attractiveness of a UCN-AP is defined with the Eq. 2.35, we describe a new dimension in the decision making for AP selection. The expected appearance duration of an encountered UCN-AP with an ID of $x$ is:

$$ E_{x/\upsilon} = \upsilon_T \rho_T + \upsilon_M \rho_M + \upsilon_L \rho_L $$

(2.59)

, which has a great impact on the connectivity of a UCN-client. In other words, a relatively transient interaction between the UCN-client and the AP puts risk on the network activities of the client regardless of the traffic behavior. Hence this term appears in the attractiveness definition of a UCN-AP as:

$$ \tilde{A}_{x/SNR}(\Delta_{x/SNR}; E_{x/\upsilon}) = A_{x/SNR} E_{x/\upsilon} $$

(2.60)

It should be highlighted that for a mobile client in the coverage of a formed cluster $C_{id}$, where id is the ID of the cluster, the term $E_{x,\upsilon} \{i \in C_{id}\}$ has no impact on the decision as this term has been observed as similar for all APs in the AP cluster. In other words, the AP selection decision mainly relies on the elimination of mobile or
static UCN members with relatively shorter-term AP functionalities.

Additionally, pointing out the redundant credit losses due to the relative transiency problem, we believe that the prediction of next AP pools on the routine paths of a UCN-client provides useful inspirations from various perspectives. Depending on the end-user preferences, one may argue that an offline time-period for a client may be tolerated if there exists a UCN-AP on the mobility path of a client, which has a potential of relatively longer-term interaction with the UCN-client. With the offline, we mean that the UCN-client has no established connection to any UCN-AP. A realistic network scenario for a mobile client can be represented with the Figure 2.24. We model a new imaginary AP in addition to the UCN-AP pool based on the user preferences as a *toleration factor* for the proposed framework and expected encounter of longer term APs on the mobility path. The imaginary AP represents offline connectivity of the client and has its own attractiveness.

![Figure 2.24: Mobility Awareness in AP Selection Decision](image)

We model the attractiveness of the offline AP based on three parameters, namely, i) the tolerance factor as the user preference, ii) the number short term AP clusters until a possible long-term or medium term AP and finally iii) the likelihood of the corresponding routine chains. Considering that the current short-term AP cluster has \( N_{td}^{(j)} \) number of possible branches (next routine paths) and the connection to a long-term pool chain is constructed by \( N_j \) number of short-term clusters, the minimum expected time period for the first encounter of a long-term AP is given as:
For this model the searches for an expected paths are simply performed independently from the additional subbranches and we limit the number of short-term clusters on the search patterns in order to prevent the redundant computation power. Representing the tolerance factor with $E_{tolerated}^{(v)}$ as the user preference and an input to the decision-making, the algorithm is simply as shown in Figure 2.25.

![Flow Chart for the Mobility Aware AP Selection Improvement](image)

**Figure 2.25: Flow Chart for the Mobility Aware AP Selection Improvement**

We believe that the mobility awareness may deserve opening another broad dimension of the AP selection problem in UCN, where the main decisions take place in a user centric way including economical aspects. As the main focus of this thesis is the load balancing oriented AP selection problem, the discussion of the AP selection problem based on the mobility and the economical issues are modeled based on relatively simple parameters.
2.4 Conclusion

In this chapter, we discuss the facilitation of load balancing by the end users, i.e. the UCN-clients. Our main objective is to introduce intelligence in AP selection, the conceptualization of which is illustrated in Figure 2.26. This intelligence comprises of the characterization of the mobility patterns for a specific UCN-client and the expected network performances of UCN-subnetworks on the mobility path.

To the best of our knowledge, we introduced a novel approach for the utilization of EM algorithm, by introducing Fourier Series Expansions of PDFs. This approach provides us with the flexibility of EM based pdf constructions as the approach sidesteps the obligation for prior assumptions to a certain extent. Additionally, a "certainty" based control methodology in decision making is discussed. This control methodology in AP selection provides a negative feedback mechanism for the highly dynamic UCN-domains. By means of the corresponding feedback mechanism, the attractiveness of particular UCN-APs are suppressed to a certain degree once the behavior of UCN-subnetworks fluctuate extensively in terms of provided network performances. Thus the decision relies on a more homogenous likelihoods.

Having the user facilitation in load balancing introduced, we start the discussion of the load balancing attempts from the UCN-APs perspectives. In the next Chapter, we firstly discuss the collaborative load balancing among UCN-APs.
3 Cooperative Load Balancing in UCN among UCN-APs

The dynamic environment of UCN and the diversity of user entities (UE), acting either as micro-providers or as clients, are certain factors responsible for the increased complexity in designing enabler mechanisms for UCNs. The distribution of the QoS provided by different UEs with different hardware capabilities has a large variance, which in turn complicates the translation of consumed or provisioned resources into credits in a consistent and fair manner. It is crucial that the UCN ensures a certain level of QoS to the members of the UCN community, comparable to those expected from traditional network infrastructures. The QoS is one of the main factors for cooperation incentives in UCN. Despite adequate incentive mechanisms, deteriorating users’ perceived network performances would create barriers for clients against joining and cooperating in UCN communities.

Even in the existence of well-crafted cooperation incentives for micro providers, a UCN-AP may better contribute to the optimal system state by migrating some of its users to another AP in the vicinity. In this sense, a crediting mechanism based on the end-to-end QoS measurements of network clients can be a good motivation for the access points i) in cooperating with each other by sharing their resources and ii) balancing the load in a fair way. The fairness, in this thesis, is translated to the compensation between the total AP-load and the resource capacity of the AP. That is to say, we aim a balance in the load based on the AP network performances depending on the hardware or the software skills. As the UCN’s main dynamic is the incentivized bandwidth sharing, the fairness term is completed by the revenue gained from the served load. The motivation of the selfish nodes serving as a provider is the challenge for this objective, which is discussed later in this chapter.

In this chapter we comment on the cooperative load balancing among the UCN-APs. This chapter is comprised of three main sections. As it is extensively utilized in this chapter, we firstly provide background information on the decision making under uncertainty and discuss Partially Observable Markov Decision Processes (POMDP). Secondly we comment on a distributed mechanism as an incentivized load balancing activities among UCN-APs, where the UCN-APs are equipped with collaboration
units. Finally we briefly discuss on an exceptional case, i.e. the environmental conditions, where the collocutor APs are not capable of balancing the load in a collaborative way and the decision should be taken in a central manner.

3.1 Brief Overview on Decision Making under Uncertainty

In most of the real world applications, decision making engines (DME) may not necessarily have a complete perception of their environment [73]. Observation-action mapping may rarely be executed as expected due to the partial observability of the operated stochastic environment [3, 74]. Under these circumstances, introducing randomness to DMEs provides with taking different actions for the same environmental conditions, which may increase the success rate [74]. Taking a sequence of actions in a deterministic way may rarely executed as expected in a partially or completely unobservable stochastic domains [3]. Considering the most successful design ever, the human brain, which decides based on random decisions of different nerve cell groups [75], we believe that there is a satisfying motivation on introducing randomization into the decision-making, for the partially observable environments.

When it comes to shedding some light on the concepts of traditional decision making; traditionally a rational agent takes an action after the comparison of the inferred and aggregated data from the environment with the pre-defined goals [73]. The decision approaches are generalized as utility (the complex combination of costs and rewards) or probability based [76, 77, 78]. Though the term uncertainty necessitates the inclusion of probabilistic approaches in the decision making problem as the agents may not have the complete information on the matter of subject. The agents may track belief states of the object, on which the rational decision has an impact, although the core enabler of the proposed frameworks may rely on the utility oriented cornerstones. Hence a comprehensive decision-making approach would cover many key words from different disciplines and perspectives including belief states, utility functions, costs or rewards.

In this part, we mainly focus on the sequential decision making problems where the agent is required to take multiple actions in a sequential form. Generally, the main goal of a consecutive approach is to maximize the long-term rewards gained from the environment or minimize the cost in case of a contrary construction of the model where the decision instances take place. Although the reward or the cost definition may be attached to many phenomena depending on the description of the system, a fundamental approach is to relate them with the observations taken after the execution. Hence the optimal policy in such systems is generally analyzed.
3.1 Brief Overview on Decision Making under Uncertainty

by taking the trade off between intermediate rewards and the total reward for the long-term activities. It is worth mentioning that a definition of comparable utilities (rewards / costs) may not hold for all systems while selecting better strategies [79, 80], nevertheless in this part we mainly focus on the situations where the reward / cost definition of different strategies are in a common form and comparable.

Specifically, in the construction of various decision making models under uncertainty (either utility based or probabilistic) the system states are described with a form including actions as a function of states resulting in a transition or more generally a function of results [81]. A generic form of the proposed models can be represented as [82]:

\[ s_{k+1} = f(s_k, a_k, w_k), \quad s_k, s_{k+1} \in S \] (3.1)

where the \( S \) is the set of all possible states and \( k \) stands for the iteration time. The \( a_k \) is an element of the action set \( A \), the elements of which are the possible strategies with comparable utilities that the agent may take. Finally the \( w_k \) represents the random nature of the problem construction, which depends on the \( a_k \) and \( s_k \) and is an input to the design construction with the nature of a conditional probability distribution \( P\{w_k|a_k, s_k\} \). Clearly this formulation holds the Markovian Property as the next world state reformation is regardless of the historical behavior of the agent or the impact on the environment, nevertheless depends on the the current activities and the state[83]. We start with a very brief introduction to the Markov Decision Processes (MDP) for the optimal solution of problems designed with the Markovian Property. Before providing the details on the optimal policy solution for the sequential decision processes with the Markovian Property, we briefly discuss on the MDP and POMDP based approaches the Network literature.

3.1.1 MDP and POMDP in Network Literature

Altman E. [84], provides a broad survey on the applications of Markov Decision Processes in communication networks and classified the related work under i) information issues and action delays, ii) call admission control iii) buffer management and packet admission control iv) flow and congestion control v) routing vi) scheduling of service vii) polling viii) wireless and satellite communications. Among these variety of applications, Rathnasabapathy et al. [85], use POMDPs as a basic framework for multi-agent planning in the context of network routing. In this study, three different perspectives of agents with different MDP solution approaches are used. The main motivation of this study is to optimize routing by means of different routing
protocols at each time depending on the observations and system state and hence to decide on optimal information sharing in between agents. In this model, the omniscient agent models the state of the network as an MDP problem because it has exact information on the system state. This agent observes the state of whole network and dictates course of actions to all other nodes in the network. In this framework, the agents poll each other in order to have the current state information. The agent communication model is formulated with a POMDP model.

Panangadan et al. [86], propose a multi-agent system with autonomous abilities and are equipped with MDP optimization tools for the optimal utilization of energy as a primary concern in the sensor networks. The objective of the proposed system is to control adaptively the sensor sampling rates in a sensor network used for human health monitoring, while taking the criticality of the patient. It is claimed that the overall system lifetime is increased and a robust monitoring system is proposed guaranteeing optimal monitoring depending on patients. MDP controllers varied the frequency at which the data is collected in an optimal way. A very interesting point of this study is the methodology of reinforcement learning techniques that are used to decide on policies statically once the dynamic modeling of the proposed system is not available.

Haas et al. [87], propose MDP formulated methods for the optimization of admission-control policies in resource allocation optimization for the wireless networks. They provide an example of a user trying to make a call through a wireless communication channel for which the network as a whole system should decide whether to forward and admit the call or not, which may in future result in a drop of call for other users even if there exists enough resources. The motivation of formulating these processes with an MDP model is to construct and formulate admission-control policies in an optimal way that decides for each new call and handoff call, whether it should be admitted or rejected.

Iwanari et al. [88], believes in the popularity of distributed POMDP models for modeling multi-agent systems in uncertain domains and in this recent study, it is claimed that these models would be widely used in near future for intelligent decision makers of the network. They focus mainly on the communication problems, which may arise among agents in the network and propose an extended version of POMDP models called Network Distributed POMDPs. A communication model for agents working with POMDP models is suggested in order to synchronize agents with each other and update their belief states so that the size of local policies would not be growing exponentially and no more complex solutions would be needed.

Zhao et al. [89], focus on the decentralized cognitive MAC protocols that allow
3.1 Brief Overview on Decision Making under Uncertainty

Secondary users to independently search for the spectrum opportunities without a central coordinator or a dedicated communication channel. Due to the nature of wireless communication platforms, the secondary users may not be able to sense the full spectrum. This in turn results in an uncertain environment for secondary users, which may be well modeled with POMDP. In this study, an analytical framework for opportunistic spectrum access based on POMDP is developed. This approach is based on the motivation of an integration between the spectrum access protocol design, the physical layer dynamics and the traffic statistics determined by the application layer of the primary network.

The MDP and POMDP models are additionally utilized broadly in the security literature. For instance Darling et al. [90], model the network as a stochastic domain due to the possibly utilized IP spoofing techniques, where the source address may not be detected in an attack case. An MDP based solution is proposed in order to introduce the randomization to the optimal policy determination.

3.1.2 POMDP Model Basics

POMDP framework is a good fit to model partially observable real-world sequential decision processes, which introduces the randomization in decision making instances. In a POMDP model, the controlled world has pre-defined environmental states and actions of decision-makers might cause a transitions between world-states. The instant world states or the transitions between states are not completely vivid, but only partially observable, through imperfect observations. Therefore the DME takes periodically observations and keeps a Bayesian estimate of the likelihood of each state. The solution model includes i) transition probability distributions based on impacts of DME actions on the instant belief state ii) observation probability distribution function indicating how likely an observation might be taken for the state and a specific action, and finally, iii) reward or cost functions associated to each state, action couples. Consequently, the POMDP framework is a six tuple of states set, action set, state transition function, reward set, observation sets and observations probabilities. Although a very brief information on the model construction of the POMDP is provided in this thesis, the complete details of mathematical formulation can be found in [3, 91, 92, 93, 94, 95, 82].

Taking the system environmental states into the consideration and detecting uncovered system states would be a strong equipment in the adaptation of the network to the new working conditions. Therefore, we utilize a POMDP model in certain decision-making, which might be altered, enhanced and propagated as a new policy, to the decision making engines with different tasks in this Thesis.
3 Cooperative Load Balancing in UCN among UCN-APs

3.1.2.1 POMDP Construction

Markov decision processes (MDPs) are discussed broadly in the literature [96, 97, 98, 99]. The MDPs are used as basis for many sequential decision making processes. An MDP framework is represented with the form of $< S, A, T(s, a, s'), R(s, a) >$, where the $S$ represents the finite set of world states and is composed of elements $\{s_i\}$’s, which is the base environment model. The finite set $A$ is the set of actions $\{a_i\}$’s that an agent may take. The $T$ is the function known as state transition probability distributions and gives the likelihood of landing on the state $s'$, given that the current state is $s$ and the action $a$ is taken. Finally the $R$ is the reward for the action $a$ given that the state is $s$, which is the utility definition that the agent may except. It is worth mentioning that although subjective probabilities may be assigned in an intuitive way for the state transition probabilities $T$ in the construction of the system definition, the validation of the assigned probabilities is as discussion point as they may be descriptively inaccurate especially when the complexity of the problem is high [100].

It should be highlighted that the state of the agent world is known by the agent. In other words, the agent has complete information on the impact of the action and the transition of the state due to the action taken by the agent. Nevertheless there is still the uncertainty on the transition likelihood of the current state due to a planned action to be taken. This is where the MDP solutions are taken into account in order to reveal the optimal policy $\pi$ under the given circumstances, i.e. based on the model construction. The concept of the optimality necessitates the independency from the iteration stages $k$, where the main motivation of the agent is to maximize the long term rewards at each step. Thus the main objective of the MDP process is to find out an optimal policy beforehand instead of iteratively searching for the next steps at each action stage. An agent would gain in case of finite-horizon $K$ iterations:
3.1 Brief Overview on Decision Making under Uncertainty

\[ R_t = E\{\sum_{i=0}^{K-1} R(s_k, a_k)\} \] (3.2)

Clearly, if the size of finite-horizon iterations may not be known before hand and the agent is expected to run for a long-term, a more suitable infinite-horizon model is proposed with a discount factor \(0 < \gamma < 1\), where the expected reward by the agent is defined as:

\[ R_t = E\{\sum_{i=0}^{\infty} R(s_k, a_k)\gamma^k\} \] (3.3)

The two objectives of taking a discount-factor into account are i) to increase the dominance of the actions, in other words the rewards, of the agent for the earlier stages and ii) to ensure the convergence of the sum for the further optimizations.

When it comes to the discussion point on the optimality of the determined policy, it has been shown that an optimal policy is a unique solution for the MDPs\[101\]. With a classical Dynamic Programming solution method \[102\], the total expected rewards at state \(s\), for taking a set of iterative actions is defined as \(V_\pi = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s')V_\pi(s')\), which is called the value function. The term \(\pi\) represents a policy, which is not necessarily the optimal one. Associating a policy with a given value function as \(\pi_V\), the optimal behavior in taking the next action at state \(s\) is:

\[ \pi_V = \arg\max_a \{R(s, a) + \gamma \sum_{s' \in S} T(s, a, s')V(s')\} \] (3.4)

Considering on the optimal state-action mapping, clearly the next step policy components \(V_k(s)\) can be calculated with terms \(\pi_k\) and \(V_{k-1}\). Though the steady state definition nature puts a limit on the calculation of the next step value function, which forms the basic idea behind the optimality option. This is due to the fact that as \(\lim_{k \to \infty} V_k(s) = V(s)\) as the value function becomes independent of the iteration number. Hence for the faster convergence rates in finding the optimal condition for the policy definition, the Bellman error magnitude \[102\] \(\epsilon\) is defined over the iterations in order to guarantee faster termination of the iterations. That is to say, the iteration is terminated at the \(N^{th}\) iteration once \(\max_{s \in S}|V_N(s) - V_{N-1}(s)| \leq 2\epsilon\frac{1}{1-\gamma}\).

The very basic advantage of a system, for which the MDP model is used in determining the optimal policy, is the full observability on the instant state of the agent world. In partially observable environments however, the MDP may not provide the optimality condition. These circumstances necessitate the utilization of an enhanced model, namely, the POMDP, where the agent has a belief on the envi-
The POMDP model extends previously provided MDP framework tuple, \(< S, A, T(s, a, s'), R(s, a) >\), with two additional sets, namely, i) the finite set of observations \(\Omega\) and ii) the observation function \(O(s', a, o)\), which is the likelihood of making the observation \(o\), given that the next state is \(s'\) and the action \(a\) is taken. As illustrated in Figure 3.2, the POMDP model includes the state-estimator as an additional dynamic in order to provide a belief for the current state of the agent world. The next state estimation \(b_{k+1}\) is a function of the current belief state \(b_k\), the taken action \(a_k\) and the observation \(o_k\).

\[
b(s_{k+1}) = P\{s_{k+1}|a_k, b_k\} = O(s_{k+1}, a_k, o_k) \sum_{s \in S} T(s, a_k, s_{k+1}) b(s)
\]  

(3.5)

Now considering a \(t\)-step non stationary policy \(A\) and a \(k\)-step move of the agent, we are interested in calculating the expected total sum of the rewards gained as the value function. As the first reward is simple the reward associated to the known state:

\[
V_k(s) = R(s, A(s)) + \gamma \sum_{s' \in S} T(s, a_k, s') \sum_{o_k \in O} O(s', A(s'), o_k) V_{o_k}(s')
\]  

(3.6)

Remembering that the state information is not provided for the agent, the expected
optimal policy is calculated over the belief-states as the belief state value functions are given as $V_k(s) = \sum_{s \in S} b(s)V_k(s)$. Kaelbling et al. [3] proposes a more convenient method in representing the value functions, where the $\alpha_k = \langle V_k(s_1), ..., V_k(s_n) \rangle$ vector represents the value function distribution among possible states and $V_{k+1}(b) = \arg\max_{k \in P} b^T \alpha_k$.

Having introduced the basics about decision-making under uncertainty, we start with discussing the cooperative load balancing attempts of the UCN-APs in the following section.

### 3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

In this section, we discuss a utility-based decision making framework for the incentivized cooperation based load balancing mechanism in UCN environment. We model the problem of UCN load balancing using Partially-Observable Markov Decision Process (POMDP). Initially, we start with the case of pairwise load balancing attempt in the UCN, which comprises of a collaboration initiator, i.e. the requester, and the collocutor, i.e. the requestee. Thus we model the utilities of two parties (i.e., requester and requestee) involved in a collaboration for load balancing. The mathematical model that we present in this work forms the basis for a QoS-based crediting mechanism, providing incentives for APs to share the load i.e., the clients, in a fair way, thereby ensuring, to a certain extent, the homogenous level of QoS throughout the UCN community. The proposed models are presented here for UCN, however, we believe that they can be extended for inter-operator resource sharing scenarios.

The rest of this part is organized as follows: Section 3.2.1 provides the state-of-the-art solutions for the load balancing scenarios in wireless networks. In Section 3.2.2, we briefly comment on the related works from the UCN point of view, after which we state our approach in Section 3.2.3. Section 3.2.4 provides the incentivized cooperation based load balancing method; Section presents the evaluation method and the related real test-bed scenarios and results. Finally the section 3.2.6 is devoted for a discussion on the proposed model.

#### 3.2.1 Related Work

The load balancing problem is broadly discussed in the network literature as one of the core factor in the optimal resource allocation schemes. The term, a balanced load, bears various meanings depending on the implementation oriented disciplines.
Cooperative Load Balancing in UCN among UCN-APs

and on type of network resources where the load balancing activity takes place. Hence, the term attains different levels of attraction in different layers or mechanisms of the network as the activity may take place in the physical layer, such as balanced channel assignment or cell breathing techniques, in the admission control phase, even among access technologies, or based on the network entity where the load balancing decision takes place, such as AP selection techniques (the load balancing oriented AP selection proposals are discussed in Section 2.1.1 and hence will not be repeated in this part). Additionally, many protocol level activities have load oriented mechanisms such as TCP slow-start strategy. In this section we provide a brief overview on the load balancing activities in the literature.

In [103], Wenxiao discusses in the network architecture level for the load balancing among heterogeneous wireless networks. The central load balancing techniques are criticized due to the low reliability and less adaptability for the scalable systems, whereas the distributed load balancing techniques raise the traffic overhead due to additional signaling. This study proposes grid architecture for the heterogeneous networks and a semi-centralized control unit, which is responsible for load balancing among heterogeneous networks. The grid computing definition is provided in the paper as the service for sharing the computer power and storage in an organizational way. The heterogeneous networks are similar to the grid computing techniques.

Fischer et al. [104] propose a distributed load balancing algorithm by means of adaptive channel allocation strategies. They discuss simulated annealing, graph coloring, neural networks and simulated annealing as the techniques proposed for the channel allocation problem for cellular networks. Mishra, however, [105] discusses a client driven channel allocation method. The proposed channel assignment problem is based on the conflict set coloring spectrum. One of the traditional approaches in channel selection for the APs is carried out by monitoring the load on the assigned channel and dynamically changing the WLAN frequency band to the channel, which is less loaded. This is called as least loaded channel search (LCCS) approach. One another approach is to perform spectrum analysis and assign the channels at the boot up manually. The authors of [105] state the problem of interferences within the clients rather than the APs. As the clients are also in their coverage areas, which might be hidden for the APs, they may interfere themselves. The load balancing aspect in this proposal is related with the client-AP associations and refers to the [106] for the client-AP association model once the channel assignment is met. On the other hand, the [107] proposes the load distribution control by means of adjusting the transmission power of beacon frames, which is optimized further in a second level control mechanism assigning the channels to decrease interference.
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

The paper concentrates on load and interference definitions in terms of linear equations and proposed the distribution of channels among APs based on their mutual interferences.

In one of the cell-breathing based application, the [108] focuses on a technique, which differs from the ordinary coverage shaping in the sense that only the transmission power for the AP beacon messages are dynamically changed. To this end, the authors define the WLAN cell range is defined as a region where the corresponding AP beacon frame is transmitted with the strongest RSSI. It should be highlighted that the STAs check only the transmission power of the beacon frames for the WLANs especially when they are idle, which brings about the ability to fake the STA by means of receiver sensitivity threshold while keeping the throughput same. Nevertheless the solution suffers for the network conditions where the geographically AP adjacent clients consume the bandwidth highly in comparison to the farther clients. In [109], the proposed algorithm consists of two steps, firstly the most congested AP is detected and then after in the second phase the transmission power is decreased in discrete steps in order to find out the optimal users assignment. Once the AP transmission power received by a client drops below than the receiver sensitivity threshold -90dBm, the clients start scanning for the new APs for re-associations. In a similar approach [110], Wang et al. focus on the AP coverage shape. Low-cost four-sector semi-smart antenna array is utilized for the APs, where the collaborative agents running on the APs dynamically change the shape of APs coverage. The main motivation of using this type of antennas is to control the coverage patterns more effectively and efficiently, e.g. by expanding the coverage in the direction of overloaded areas in order to increase the throughput and decrease collision rate. The study [111] pointed out another load balancing approach using cell breathing techniques and focused on the traffic types in order to heal congestion prevention by means of QoS mechanism provided in IEEE802.11e standards. [112] discusses the signal strength effect for the QoS of the network and hence the a self-congesting network by means of retransmissions due to packet losses. The study criticizes the migration of client traffic load from one AP to another without the RSSI consideration of co-locutor APs.

Frame drop rate of real-time sessions in an access point’s transmission queues are proposed as the load measurements in [113]. Nevertheless, the backward compatibility of this method is criticized due to implementation complexity [114]. Similarly, delay time between scheduled and actual transmission time of periodic beacon frames can be a good measure for the load of an AP as proposed by [115]. Having a measure of the load on the AP, it is possible to balance the load among APs in the network
through both wireless station (WS)-based solutions or network-based solutions. AP selection for WS-based approaches can be realized in a static or dynamic fashion; however, letting stations to dynamically choose an AP can lead to unstable WS-AP associations. As a result, similar measurements among nearby wireless stations would create a collective handoff process, causing a so-called ping-pong effect [114]. A possible remedy to this problem is to assign random waiting times and number of measurement instances for each WS before executing the handover [116]. On the contrary, [117] proposes an AP-based load balancing system for which the overloaded APs reduce their transmission power of beacon signal so that it is less likely to be discovered by new stations. However, this approach may have an adverse effect on the quality of experience for already associated wireless stations. Moreover, this method may not guarantee load balancing among APs when each AP decrease their transmission power in a similar fashion without a coordination.

In an association control method based load balancing approaches, Bejerano et al. [106] propose a min-max load-balancing algorithm where the clients take measurements for the channel quality and provide the measurement parameters to a central controller deciding the client-AP associations. The load of an AP is defined as the effective bit-rate from the perspectives of each client. [118] focuses on the quantization of the relationship between the random channel access length, system channel utilization and random user blocking probability. The paper focuses on the analytical model and impact for the load balancing among baseline two-cell network with simplified physical layer models, realistic traffic scenarios and various network configurations. [119] points out that the IEEE802.11n is a promising ubiquitous networking technology, which suffers from the handover association latencies and the overloading problem as the association decision relies on the physical layer parameters rather than the QoS consideration. The paper focuses on the minimization of the handover delays in terms of scanning, detecting, re-association and authentication. Checking the bootstrap of the AP, the proposed solution initiates QoS estimation parameters, which is disseminated in the beacon frames. This value is stationary and each associated client performs Bayesian estimation with cumulative sum (CUSUM) by monitoring the AP packet loss rates and dynamically selects the network. In [120] two-phase control strategy is adopted for the load balancing schemes. In the first phase the statics on the QoS guarantee are provided to the stations, whereas in the second phase, dynamic vertical handoffs are performed among the stations to prevent fluctuations in terms of performance. Ning et al. [121] discusses an algorithm, which is claimed to balance the load based on the simulation results by means of instructions for the new client arrivals to hand off to the suit-
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

able under-loaded network. Finally, in another association control mechanism [122], a joint group call admission control (JGCAC) algorithm is proposed for heterogeneous networks. Based on a two-step procedure, the numbers of clients are detected for each network and then after the certain users are admitted to different networks. With an assumption of homogenous traffic load each network client, a central unit collects the bandwidth occupation per client and redistributes the clients among the heterogeneous networks.

3.2.2 Evaluation of the Literature from the UCN Perspective

The UCN is a very dynamic platform, where the main interests of platform stakeholders rely on the expansion of the coverage for individual Internet access. This, in turn, necessitates the acquisition of enough incentives by means of serving as the micro-provider from time to time with a revenue oriented manner. One another setting worth mentioning is the wide variety in the resource capacities provided by the UCN-APs.

These two dimensions are not yet, to our best of knowledge, broadly discussed in the literature as the UCN concept is a very fresh approach. Various studies focus on the load balancing attempts in the presence of identical APs in terms of resource capacities, e.g. hardware and software capabilities. Secondly, various network scenarios, discussed in the literature, omit the presence of selfish APs. Thus a feasible mechanism is required for the UCN addressing these settings.

3.2.3 Our Approach

We initially focus on how to stimulate the UCN-APs to cooperation for balancing the load. To this end we propose QoS based crediting model as an incentive mechanism for the UCN. We make use of the active measurement system, which is detailed in Section 2.1.4.1, as it provides a common understanding of the subnet-work performance both for the UCN-AP and the UCN-client. Based on the network performance indicators, we set a proportional unit resource cost (URC) and assume an agreement on the URC policy in UCN. In this context the resource is translated to the consumed bandwidth (BW). That is to say, a loaded or worse QoS providing UCN-AP earns less in comparison to an unloaded or better QoS providing UCN-AP. This, in turn, motivates the UCN-APs for sharing the load in a fair manner, which we define as revenue oriented strategy.

Considering about the participation of the resource limited devices into the UCN, we believe that there is enough motivation for limiting the BW utilization of UCN-
clients to a certain threshold as UCN-AP owners may demand simultaneously utilization of his/her entity for various network activities. Thus the UCN-AP owner may not be oblivious to extensive resource consumptions of third parties, which in our study is translated to the total amount of AP load. To guard the experience quality from the UCN-AP owner while serving as a micro-provider, we define a second strategy orientation, namely the load oriented, for load balancing attempts. Based on the discussed characteristics of UCN so far and the presence of incomplete information, we model the strategy selections of the UCN-APs (either requester or requestee) as a partially observable domain. We solve this problem using POMDP and verify the fair load balancing and strategy selections of the UCN-APs by means of real implementation, the details of which are provided in the next section.

### 3.2.4 Incentivized Cooperation based Load Balancing

This section focuses on elaborating the proposed incentive based inter-entities cooperation for load balancing. It should be noted that the incentives are translated on multiple dimensions in this work i.e., crediting and service quality. We also believe that for realizing the UCN vision, the existing bandwidth sharing, power and memory management, and enforcing the strategy proof approaches are provisioned. In this connection, we have contributed a combined virtual currency and reputation based approach in [11]. In this study, however, we focus on enforcement of entities cooperation by proposing a crediting mechanism, where the cost of resource depends on the service quality.

In the following, we elaborate on a few proposed functional concepts, which are of vital importance for formulating the load balancing problem.

#### 3.2.4.1 Proposed Functional Concepts

The contributed functional concepts are based on extensive research literature review, real system statistics, and envisioned behavior of the considered domain.

##### 3.2.4.1.1 QoS Based Crediting Model

To capture the realistic behavior of the system in the envisioned future UCN environment we propose QoS based crediting model. In order to explain the proposed approach, consider the system domain (topology) shown in Figure 3.3. As can be seen in the figure, that

The QoS metric values are captured randomly in the last hop, which is detailed in Section 2.1.4.1. Impressed by the most research work, we translate the QoS into average delay time, average jitter, average packet drop rate and average bit rate metric values. This further dictates that the decision of offloading / leaving the UCN
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

Due to its nature, sigmoidal functions provide graceful flexibility in multiple regions, suppression of the outcome with two pre-defined thresholds from top and down guaranteeing convergence and concave / convex regions with an adjustable state transition rate or position. Based on alike reasonings, many utility-based decision-making mechanisms for different domains, such as congestion avoidance, QoS / QoE based control frameworks, power-control schemes etc., broadly make use of sigmoid characteristics in order to represent utilities [123, 124, 125, 126, 127, 128]. In this section, we propose a utility-based load balancing model in UCN environment and we translate utility into revenue, i.e., the crediting with a sigmoidal curve like characteristic for each independent component.

We define the resource as the transmitted and received bytes by the clients. The cost function of unit resource is a 4-dimensional function of aforementioned measurements. The cost function components for each measurement is defined as given in Eq. 3.7.

\[
c_\omega(\omega) = K_\omega \left( 1 + \frac{1}{1 + e^{-k(A_\omega)(B_\omega - \omega)}} \right),
\]

where \( \omega \) represents the element from the finite set of \( \Omega = \{ \text{delay, packet loss rate, jitter, bitrate} \} \). \( K \) is a coefficient that assigns the offset and controls the increment.
or decrement rate of the function. Thus $K$ may be graded as the controlling lever for evaluating different involved components and their dominance. $A$ defines the rate of state change around the mean cost value. Intuitively, $k$ decides over the expectancy (e.g., the higher the better or the lower the better) of considered component. In Figure 3.4, for ready reference we present the curve for proposed cost function against QoS index (i.e., delay). It should be noted that cost curve is the consequence of experimentation, more on this experimentation setup is detailed later in this section.

Finally, we define the cost of unit resource as the sum of each component for each metric as:

$$C_x = \sum_{\omega \in \Omega} c_x(\omega) \quad (3.8)$$

### 3.2.4.1.2 Time Driven Client Cost Model ($\tilde{c}$)

The cost as the function of time is a public knowledge in the considered settings. For instance, the requestee GW has the belief on possible bandwidth utilization of the clients in the system, which in turn can be translated into resource valuation. Thus to capture the dynamics in the resource valuation as the function of time, we propose the function of client cost vs day time. To avoid complexity, we assume that resource cost increases linearly in its utilization i.e., the resource valuation is higher in the busy hours than that of less busy hours. Having known such resource valuation pattern, both stakeholders tie their strategies to the time of the day (or resource utilization pattern). The estimation of the total profit by associating a client for the life time of a call ($t_{\text{start}} - t_{\text{finish}}$) is given as:

$$\tilde{c} = \int_{t_{\text{start}}}^{t_{\text{finish}}} z(t)C_x(N_{\text{tot}})dt \quad (3.9)$$

where $z(t)$ reflects the expected resource utilization of a client vs. day time and the $C_x(N_{\text{tot}})$ represents the cost of unit resource as a function of $N_{\text{tot}}$ associated clients.
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

3.2.4.1.2.1 Future Estimation on Reputation

The incentive mechanism motivates the community members for a continuous and imminent cooperation. The community has a tendency of grading the reputation of the member(s) as non-cooperative as the time passes, due to the variations of trust metrics over time [11, 129]. Thus the future estimation on reputation has an impact on the decision instances for cooperation. We model the belief on reputation status in three states, namely, cooperative, normal and non-cooperative. Each cooperation decision of the AP in the cooperation history is modeled as an arrival resulting in a transition from worse reputed state to a better one. On the contrary, cooperation refusal results in a transition from better reputation state to a worse one as shown in Figure 3.5a. Finally the CTMC transits from a better reputed state to a worse one due to the variations of trust metrics over time.

Numbers of cooperation and refusal decisions (2 different types of arrivals) are counted at the time of a new cooperation request from another UCN-AP as shown in Figure 3.6. Counts of arrivals and window size form the parameters of arrival processes with Poisson distribution, i.e, the expected number of arrivals in unit time period.

We define a homogenous distribution on the CTMC states as given in Eq. 3.10 due to inadequate historical observations at the boot-up time of AP.

\[ \pi = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \]  

(3.10)

The \( D_0 \) Matrix, defining the parameters of imaginary event, which represents the variation of trust matrix, is given in Eq. 4.2:
Figure 3.6: Arrivals of Cooperation Request

\[
D_0 = \begin{pmatrix}
-\lambda_1^{(c)} & \lambda_{12}^{(c)} & \lambda_{13}^{(c)} \\
\lambda_{21}^{(c)} & -\lambda_2^{(c)} & \lambda_{23}^{(c)} \\
\lambda_{31}^{(c)} & \lambda_{32}^{(c)} & -\lambda_3^{(c)}
\end{pmatrix} \tag{3.11}
\]

there is no transition from less cooperative state to a more cooperative one without a cooperation arrival. Therefore;

\[
\lambda_{21}^{(c)} = \lambda_{32}^{(c)} = \lambda_{31}^{(c)} = 0 \tag{3.12}
\]

and we model the rate of a transitions from cooperative state to normal equal to the transition rate from states normal to non-cooperative. Hence,

\[
\lambda_{12}^{(c)} = \lambda_{23}^{(c)} = \lambda^{(c)} \tag{3.13}
\]

Finally we have a \( D_0 \) matrix as:

\[
D_0 = \begin{pmatrix}
-\lambda_1^{(c)} & \lambda^{(c)} & 0 \\
0 & -\lambda_2^{(c)} & \lambda^{(c)} \\
0 & 0 & -\lambda_3^{(c)}
\end{pmatrix} \tag{3.14}
\]

The cooperation-decision would cause a transition from lower cooperative states
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

to higher cooperative states, therefore the \( D_C \) matrix is in the form of:

\[
D_C = \begin{pmatrix}
0 & 0 & 0 \\
\gamma^{(c)} & 0 & 0 \\
0 & \gamma^{(c)} & 0
\end{pmatrix}
\] (3.15)

On the contrary, the refusal-decision would accelerate the transition rate from a higher cooperative state to the lower ones, hence \( D_{IC} \) has the form of:

\[
D_{IC} = \begin{pmatrix}
0 & \tau^{(c)} & 0 \\
0 & 0 & \tau^{(c)} \\
0 & 0 & 0
\end{pmatrix}
\] (3.16)

Finally the infinitesimal generator matrix for \( I(t) \), i.e, \( D \) matrix is given in Eq. 3.17.

\[
D = \begin{pmatrix}
-\lambda^{(c)}_1 & \lambda^{(c)} + \tau^{(c)} & 0 \\
\gamma^{(c)} & -\lambda^{(c)}_2 & \tau^{(c)} + \lambda^{(c)} \\
0 & \gamma^{(c)} & -\lambda^{(c)}_3
\end{pmatrix}
\] (3.17)

where;

\[
\lambda^{(c)}_1 = \lambda^{(c)} + \tau^{(c)}
\] (3.18)

\[
\lambda^{(c)}_2 = \lambda^{(c)} + \tau^{(c)} + \gamma^{(c)}
\] (3.19)

\[
\lambda^{(c)}_3 = \gamma^{(c)}
\] (3.20)

Note that \( D \) is an irreducible generator matrix, therefore the steady-state probability distribution of the reputation states \( \pi = \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \) vector satisfies the following equations:

\[
\pi \cdot \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T = 1
\] (3.21)

\[
\pi D = 0
\] (3.22)

solving the Eq. 3.22 based on Eq. 3.21, we define the steady-state probability distributions as:

\[
P\{Co\} = \frac{(\gamma^{(c)})^2}{\gamma^{(c)}(\lambda^{(c)} + \tau^{(c)}) + (\lambda^{(c)} + \tau^{(c)})^2 + (\gamma^{(c)})^2}
\] (3.23)

\[
P\{N\} = \frac{\gamma^{(c)}(\lambda^{(c)} + \tau^{(c)})}{\gamma^{(c)}(\lambda^{(c)} + \tau^{(c)}) + (\lambda^{(c)} + \tau^{(c)})^2 + (\gamma^{(c)})^2}
\] (3.24)
3 Cooperative Load Balancing in UCN among UCN-APs

\( P\{Nc\} = \frac{(\lambda(c) + \tau(c))^2}{\gamma(c)(\lambda(c) + \tau(c)) + (\lambda(c) + \tau(c))^2 + (\gamma(c))^2} \) (3.25)

A higher steady-state probability on a lower cooperative state would motivate the AP to cooperate, on the contrary a lower probability would provide more flexibility on the cooperation decision.

3.2.4.1.2.2 Future estimation on bandwidth utilization of the UCN community

History analysis for initiating a cooperation by the AP provides the AP with an observation on the characteristics of the community in bandwidth utilization. Based on the proposed MAP model, a frequent cooperation request initiated by the AP is interpreted as a higher probability of becoming congested most of the operation time. A seldom initiation, on the contrary, would provide the AP with the motivation of increasing its own revenue and the tendency of providing better QoS to its clients. We model the believes of our agent on the community’s bandwidth demand in 3 different states, namely bandwidth economic, normal and bandwidth hungry. The reduction in cooperation demand of the AP causes transitions to a community characteristic of less bandwidth consumption state. Initiating a cooperation request, however, is modeled as the single arrival of MAP model resulting in a transition from less bandwidth utilization state to a higher one, as illustrated in Figure 3.5b.

Following the similar procedure starting from Eq. 3.10 to Eq. 3.25 and starting with a homogenous distribution on the initial states, the steady-state probabilities for future estimation on community behavior are given in Eq. 3.26 to Eq. 3.28, where the \( \gamma^{(bw)} \) defines the rate of negotiation initiation.

\( P\{Ec\} = \frac{(\lambda^{(bw)})^2}{(\gamma^{(bw)})^2 + (\lambda^{(bw)})^2 + \lambda^{(bw)}\gamma^{(bw)}} \) (3.26)

\( P\{N\} = \frac{\gamma^{(bw)}\lambda^{(bw)}}{(\gamma^{(bw)})^2 + (\lambda^{(bw)})^2 + \lambda^{(bw)}\gamma^{(bw)}} \) (3.27)

\( P\{H\} = \frac{(\gamma^{(bw)})^2}{(\gamma^{(bw)})^2 + (\lambda^{(bw)})^2 + \lambda^{(bw)}\gamma^{(bw)}} \) (3.28)

3.2.4.1.2.3 Future estimation on the client population of the AP

Tracking the client arrival / departure rates, the AP constructs its own believes on the prospective arrival probabilities and thereby the load conditions. Depending on the arrival and departure rates, the AP continuously updates the MMAP model and calculates new steady-state probability distributions of the three state Markov Chain, given in Figure 3.5c.
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

The steady-state probabilities for the population of the community are given from Eq. 3.29 to Eq. 3.31, where $\tau^{(p)}$ defines the arrival rate of the clients and $\gamma^{(p)}$ defines the departure rate.

$$P\{D_s\} = \frac{(\gamma^{(p)})^2}{\gamma^{(p)}(\lambda^{(p)} + \tau^{(p)}) + (\lambda^{(p)} + \tau^{(p)})^2 + (\gamma^{(p)})^2}$$  \hspace{1cm} (3.29)

$$P\{N\} = \frac{\gamma^{(p)}(\lambda^{(p)} + \tau^{(p)})}{\gamma^{(p)}(\lambda^{(p)} + \tau^{(p)}) + (\lambda^{(p)} + \tau^{(p)})^2 + (\gamma^{(p)})^2}$$  \hspace{1cm} (3.30)

$$P\{C_r\} = \frac{(\lambda^{(p)} + \tau^{(p)})^2}{\gamma^{(p)}(\lambda^{(p)} + \tau^{(p)}) + (\lambda^{(p)} + \tau^{(p)})^2 + (\gamma^{(p)})^2}$$  \hspace{1cm} (3.31)

3.2.4.1.3 Arrival Rate Estimation  The arrival rates are estimated using Expectation-Maximization algorithm. In accordance with the aforementioned three scale belief on future estimations, and due to the reasoning that Poisson mixtures fit the observations in a much better way in case of multiple routines [130], the EM algorithm is applied on a mixture of three Poisson distributions. The main purpose of this lower-level mechanism is to provide the higher-level future estimation processes with the suitable arrival rate. We define the probability of counting $k$ number of the same observation in a fixed window, given the set $\Theta = (\lambda_{nC}, \tau_{nC}, \lambda_N, \tau_N, \lambda_C, \tau_C)$, where $\lambda$’s are defined as the arrival rates in non-critical, normal, critical environmental states, and $\tau$’s represent the instant probability distribution of states;

$$p(k|\Theta) = \tau_{nC} \frac{e^{-\lambda_{nC}} \lambda_{nC}^k}{k!} + \tau_N \frac{e^{-\lambda_N} \lambda_N^k}{k!} + \tau_C \frac{e^{-\lambda_C} \lambda_C^k}{k!}$$  \hspace{1cm} (3.32)

We follow the similar processes for the variable estimations as described before in Section 2.2.4.1.1. In this model, a case of an increase in the probability distribution of a specific environment state represents a transition to the corresponding state. Hence the corresponding arrival rate is forwarded to the upper level.

3.2.4.1.4 Load Parameters  : Assuming that the environmental impact on the QoE of each client is homogenous, we define the load parameters as a function of QoS measurements (the finite set of QoS measurements, i.e., the $\Omega$). Referring to the assumption that the clients would leave the UCN community after a threshold average delay time, the UCN-AP sets high load around the pre-defined thresholds, depending on its own strategy / policy. In order to provide a better understanding,
we provide an example Load definition based on the average delay time as:

\[
Load = \frac{\text{Load}_{\text{Avg}}}{\text{Delay}_{\text{th}} - \text{delay}}, 0 < \text{delay} < \text{Delay}_{\text{th}}
\]  

\[(3.33)\]

### 3.2.4.2 Load Balancing Model

The proposed load balancing approach is basically driven by two parameters namely, the number of clients those are to be offloaded to the requestee UCN-AP and the corresponding monetary cost (i.e., virtual credit, which is set by the requester). In order to model the load balancing in the considered settings, we first model the utility function of the involved stakeholders.

#### 3.2.4.2.1 Utility Function of Requestee (\(u_e\))

The proposed utility function is composed of reward and cost components. Given that we model the UCN environment in partially observable domain, we expect that the utility is greatly influenced by the POMPD model policy. Depending on the belief-state of the network and the probabilistic approaches on the future states, the decision making engines re-calculates the coefficients of a single issue. The utility function for the requestee is given in Equation 3.34:

\[
u_e(N, t, s, M) = N\hat{c}(t) + \hat{\pi}M - c(s)N
\]

\[(3.34)\]

where \(N\) represents the number of clients and \(M\) represents the monetary costs. The coefficient \(\hat{\pi}\) reflects the importance of the monetary service cost in the utility. The notation \(s\) in \(c(s)\) represents three different states namely i) load oriented, ii) reputation oriented, and iii) both load and reputation oriented. It should be noted that \(c(s)\) is state-dependent and is the function of different parameters for different states.

\(c(s)\) takes current load status, estimated load status, client cost with respect to time of the day, and client arrival pattern as parameters when it represents the loaded state. In the non-cooperative state, \(c(s)\) is the function of estimated reputation only. Finally in both loaded and non-cooperative state, \(c(s)\) is the function of all the parameters mentioned for loaded state plus the estimated reputation parameters. We define \(c(s)\) coefficient as:

- if \(s = \text{Load Oriented}:\)

\[
c(s) = \text{Load} \ast \text{Pr}\{\text{Cr}\} \ast \text{Pr}\{\text{H}\} \ast z(t)
\]

\[(3.35)\]
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

- if $s =$ Reputation oriented:

$$c(s) = A_{\text{compensation}} \times Pr\{Co\}$$  \hspace{1cm} (3.36)

- if $s =$ Both Load and Reputation oriented:

$$c(s) = \text{Load} \times Pr\{Cr\} \times Pr\{H\} \times z(t) \times Pr\{Co\}$$  \hspace{1cm} (3.37)

$\tilde{\pi}$ depends on the total credit of the UCN-AP. The monetary parameter loses its importance in the utility calculation in case of a high amount of total credit of the UCN-AP. This is due to the reasons that i) the incentive mechanism is based on the combination of reputation and monetary, i.e., the UCN-AP keeps a balance on its reputation with the monetary [11] and ii) the main goal of the UCN environment is to broaden connectivity by sharing own bandwidth and utilizing bandwidth of others rather than stocking virtual currency. We model this parameter a sigmoid-like curve as:

$$\tilde{\pi} = K_m \cdot \left( \frac{1}{1 + e^{-(A_m)(B_m - \text{Own Credit})}} \right)$$  \hspace{1cm} (3.38)

3.2.4.2.2 Utility Function of Requester ($u_r$) As opposed to the requestee’s utility function, the utility function of requester is straightforward. The requester is interested in the service quality, where his (requester’s) preferences over the service quality is strictly driven by the states. The utility function of requestor is thus given as:

$$u_r(s, \omega) = \begin{cases} 
1 & \text{(if } s \text{ is load oriented} \land \\ & \omega \geq \omega_{th}) \lor \\ & \text{(if } s \text{ is revenue oriented} \land \\ & \text{AP estimates earning more} \\ & \text{with better QoS and less clients)} \\
0 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (3.39)

Intuitively, the expectancy (e.g., the higher the better or the lower the better) decides the equality relationship between $\omega$ and $\omega_{th}$.

In a load oriented state, the UCN-AP decides on the $N_{neg}$ in order to prevent clients from leaving the UCN environment due to degraded QoS. Referring to the
3 Cooperative Load Balancing in UCN among UCN-APs

Eq. 3.33 and depending on a pre-assigned load threshold $\text{Load}_{th}$, the AP forces clients to churn out to the collocutor AP until the threshold measurement $\omega_{th}$ is reached as shown in Eq. 3.40. The setting of threshold values depends on the individual strategy of the UCN-AP.

$$w_{\text{delay},th} = \text{Delay}_{th} + \frac{A_{\text{Load}}}{\text{Load}_{th}} \quad (3.40)$$

In a revenue oriented state, the AP adopts the strategies aiming it increasing its own profit. The estimated profit of the UCN-AP from the $N_{tot}$ clients is given as:

$$\text{Profit}\{N_{tot}\} = N_{tot}\tilde{c} \quad (3.41)$$

$$N_{neg} = \arg\max_N \{\text{Profit}(N_{tot} - N) - \text{Profit}(N_{tot})\} \quad (3.42)$$

Now that we have presented the utility functions of both involved stakeholders, in the following section we discuss the POMDP policies for the APs.

3.2.4.2.3 POMDP Policy of the Requestee  We believe that requestee has three different approaches (which in terms of POMDP may be defined as world states). Three approaches play an important role in computing the requestee’s utility function. We term the mentioned states as i) Load-oriented, ii) Reputation-oriented and iii) Load & Reputation-oriented. The observations set includes the load parameters and requestee’s beliefs on future reputation. The action set basically aims at choosing the utility function i.e., based on the belief state the corresponding utility function is selected. We calculate the general POMDP policy of the requestee using the main states and transitions as is illustrated in Figure 3.7a. The model parameters, i.e., i) the state-transition probabilities, ii) the observation probability function, iii) the reward function is provided in Tables 3.1, 3.2 and 3.3 respectively.

3.2.4.2.4 POMDP Policy of the Requester  The requester has two world states namely load-oriented and revenue-oriented. The observations set includes only load parameters. Based on the belief state, the strategy of the requester changes in calculating the number of clients to be migrated. The general POMDP policy of the requester is illustrated in Figure 3.7b. The model parameters, i.e., i) the state-transition probabilities, ii) the observation probability function, iii) the reward function is provided in Tables 3.5, 3.6 and 3.7 respectively.
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

(a) Requestee POMDP Policy

(b) Requester POMDP Policy

Figure 3.7: POMDP policies of stakeholders

<table>
<thead>
<tr>
<th>Action</th>
<th>Start State</th>
<th>End State</th>
<th>Transition Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Loaded</td>
<td>Reputation Oriented</td>
<td>0.00</td>
</tr>
<tr>
<td>*</td>
<td>Loaded</td>
<td>Loaded</td>
<td>0.50</td>
</tr>
<tr>
<td>*</td>
<td>Loaded</td>
<td>Both Oriented</td>
<td>0.50</td>
</tr>
<tr>
<td>*</td>
<td>Reputation Oriented</td>
<td>Reputation Oriented</td>
<td>0.00</td>
</tr>
<tr>
<td>*</td>
<td>Reputation Oriented</td>
<td>Loaded</td>
<td>0.50</td>
</tr>
<tr>
<td>*</td>
<td>Reputation Oriented</td>
<td>Both Oriented</td>
<td>0.50</td>
</tr>
<tr>
<td>*</td>
<td>Both Oriented</td>
<td>Reputation Oriented</td>
<td>0.33</td>
</tr>
<tr>
<td>*</td>
<td>Both Oriented</td>
<td>Loaded</td>
<td>0.33</td>
</tr>
<tr>
<td>*</td>
<td>Both Oriented</td>
<td>Both Oriented</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Table 3.1: The state-transition probability function for requestee POMDP model
### Cooperative Load Balancing in UCN among UCN-APs

<table>
<thead>
<tr>
<th>Action</th>
<th>Start State</th>
<th>Observation</th>
<th>Observation Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>CR</td>
<td>0.75</td>
</tr>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>NL</td>
<td>0.05</td>
</tr>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>CR</td>
<td>0.15</td>
</tr>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>NR</td>
<td>0.05</td>
</tr>
<tr>
<td>SLO Reputation Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SLO Reputation Oriented</td>
<td>NL</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>SLO Reputation Oriented</td>
<td>CR</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>SLO Reputation Oriented</td>
<td>NR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SLO Both Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SLO Both Oriented</td>
<td>NL</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>SLO Both Oriented</td>
<td>CR</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>SLO Both Oriented</td>
<td>NR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>CR</td>
<td>0.45</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>NL</td>
<td>0.05</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>CR</td>
<td>0.05</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>NR</td>
<td>0.45</td>
</tr>
<tr>
<td>SRO Reputation Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SRO Reputation Oriented</td>
<td>NL</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>SRO Reputation Oriented</td>
<td>CR</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>SRO Reputation Oriented</td>
<td>NR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SRO Both Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SRO Both Oriented</td>
<td>NL</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>SRO Both Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SRO Both Oriented</td>
<td>NR</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>SBO</td>
<td>Loaded</td>
<td>CR</td>
<td>0.75</td>
</tr>
<tr>
<td>SBO</td>
<td>Loaded</td>
<td>NL</td>
<td>0.05</td>
</tr>
<tr>
<td>SBO</td>
<td>Loaded</td>
<td>CR</td>
<td>0.15</td>
</tr>
<tr>
<td>SBO</td>
<td>Loaded</td>
<td>NR</td>
<td>0.05</td>
</tr>
<tr>
<td>SBO Reputation Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SBO Reputation Oriented</td>
<td>NL</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>SBO Reputation Oriented</td>
<td>CR</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>SBO Reputation Oriented</td>
<td>NR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SBO Both Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SBO Both Oriented</td>
<td>NL</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>SBO Both Oriented</td>
<td>CR</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>SBO Both Oriented</td>
<td>NR</td>
<td>0.45</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: The observation probability function for requestee POMDP model
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

<table>
<thead>
<tr>
<th>Action</th>
<th>State</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>100</td>
</tr>
<tr>
<td>SLO</td>
<td>Reputation Oriented</td>
<td>0</td>
</tr>
<tr>
<td>SLO</td>
<td>Both Oriented</td>
<td>0</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>0</td>
</tr>
<tr>
<td>SRO</td>
<td>Reputation Oriented</td>
<td>100</td>
</tr>
<tr>
<td>SRO</td>
<td>Both Oriented</td>
<td>0</td>
</tr>
<tr>
<td>SBO</td>
<td>Loaded</td>
<td>50</td>
</tr>
<tr>
<td>SBO</td>
<td>Reputation Oriented</td>
<td>50</td>
</tr>
<tr>
<td>SBO</td>
<td>Both Oriented</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3.3: The reward function for requestee POMDP model

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Loaded</th>
<th>Not Loaded</th>
<th>HNcSSP</th>
<th>LNcSSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>8</td>
<td>7</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>9</td>
<td>8</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>11</td>
<td>8</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>12</td>
<td>8</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>13</td>
<td>8</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>13</td>
<td>8</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>14</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>17</td>
<td>8</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>16</td>
<td>8</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>17</td>
<td>8</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>17</td>
<td>8</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>17</td>
<td>8</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>17</td>
<td>8</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>17</td>
<td>8</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>17</td>
<td>8</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 3.4: The POMDP based policy for the requestee strategy

<table>
<thead>
<tr>
<th>Action</th>
<th>Start State</th>
<th>End State</th>
<th>Transition Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>Revenue Oriented</td>
<td>0.50</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>Load Oriented</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 3.5: The state-transition probability function for requester POMDP model
### 3 Cooperative Load Balancing in UCN among UCN-APs

<table>
<thead>
<tr>
<th>Action</th>
<th>Start State</th>
<th>Observation</th>
<th>Observation Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>CL</td>
<td>0.75</td>
</tr>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>NL</td>
<td>0.25</td>
</tr>
<tr>
<td>SLO</td>
<td>Revenue Oriented</td>
<td>CL</td>
<td>0.25</td>
</tr>
<tr>
<td>SLO</td>
<td>Revenue Oriented</td>
<td>NL</td>
<td>0.75</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>CL</td>
<td>0.75</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>NL</td>
<td>0.25</td>
</tr>
<tr>
<td>SRO</td>
<td>Revenue Oriented</td>
<td>CL</td>
<td>0.25</td>
</tr>
<tr>
<td>SRO</td>
<td>Revenue Oriented</td>
<td>NL</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 3.6: The observation probability function for the requester

<table>
<thead>
<tr>
<th>Action</th>
<th>State</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLO</td>
<td>Loaded</td>
<td>100</td>
</tr>
<tr>
<td>SLO</td>
<td>Reputation Oriented</td>
<td>0</td>
</tr>
<tr>
<td>SLO</td>
<td>Both Oriented</td>
<td>0</td>
</tr>
<tr>
<td>SRO</td>
<td>Loaded</td>
<td>0</td>
</tr>
<tr>
<td>SRO</td>
<td>Reputation Oriented</td>
<td>100</td>
</tr>
<tr>
<td>SRO</td>
<td>Both Oriented</td>
<td>0</td>
</tr>
<tr>
<td>SBO</td>
<td>Loaded</td>
<td>50</td>
</tr>
<tr>
<td>SBO</td>
<td>Reputation Oriented</td>
<td>50</td>
</tr>
<tr>
<td>SBO</td>
<td>Both Oriented</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3.7: The reward function for requester POMDP model

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Loaded</th>
<th>Not Loaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.8: The POMDP based policy for the requester strategy
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

3.2.5 Experiments and Test Results

In this section, we briefly comment on the experimental setup that we developed for evaluating the proposed approach, details of which is presented in Chapter 5. We implement the topology depicted in Figure 3.3 with limited scope (i.e., using WLAN APs). The java and C based software block diagram for the client and APs are given in Figure 3.8 and 3.9 respectively, where the APs are equipped with Debian derived Voyage Linux OS [51].

AP1 is configured as less congested (i.e., characterized by the higher credits, also evident from the Figure 3.10) and AP2 is more congested (thus providing deteriorated QoS indices values). Four WLAN clients get associated to the available APs. As can be observed from the figure, clients perform horizontal handovers when necessary in order to balance the load among the APs in a fair way, based on the proposed framework; thus validating our approach.
3.2.6 Evaluation of the Proposed Framework

In this section, we analyze the mathematical model for the incentivized cooperation based load balancing in a UCN environment. We propose a QoS based crediting mechanism motivating APs (together with concerns on their reputation) for sharing the load and clients in a fair manner depending on the hardware capabilities, in order to help for a homogenous QoS through the UCN community. The proposed framework is based on utilities of involved entities, we make use of probabilistic approaches for computing the utilities in efficient manner. In this work, the AP interactions were confined to paired wise and only single bid / single response protocol. Nevertheless, the proposed solution can still hold for the multiple interactions in a UCN-AP pool.

For the proposed incentivized load balancing framework, there exist two parameters as the matter of issue, namely the number of clients to be transferred and the amount of incentive (credit in our case), which should be provided by the Requester Access Point (AP) to the requestee AP. As the number of clients to be transferred might only be determined by the Requester AP, the single parameter to be mutually determined is the amount of incentives for the collaboration. In an AP pool scenario, one may argue for an offer of incentive parameters from the requestee APs. In other words, the requester AP can disseminate only the number of clients either in a single cast manner or by means of a broadcast. In reaction to the initiated collaboration request, the requestee APs can respond with the requested incentives once they are
3.2 Cooperation Incentives Based Load Balancing in UCN: A Probabilistic Approach

Figure 3.10: Evaluation outcome
Figure 3.11: A Load Balancing Scenario by means of broadcasted bid exchange

This approach is very convenient for the proposed model due to the fact that the agreement decision relies on relatively simple parameters. The requester AP may receive a positive response from the requestee APs either simultaneously or in a random queue manner depending on various factors such as current traffic load, hardware, software capabilities, the load on the broadcast medium, etc. Nevertheless, the decision on selecting a specific requestee for the collaboration is relatively simple as the very firstly received positive response is the initiator of the collaboration in the proposed solution. Though, one may argue that an iterative new bids on the incentive parameter is required and may be handled in second or third iterations, where the very first initial collaboration request may be broken by the requester AP for better agreement points. Nevertheless, this issue is out of the scope of this thesis and we believe that such an approach may increase the complexity and sacrifice the experienced network performance by the UCN-clients due to additional latency, which stems from the possibly increasing number of bid iterations.

Secondly, the requester AP may initiate collaborations by means of polling the nearby APs one by one in pairwise manner, which is mainly discussed in this section. For this scenario, the selection of AP sequence may rise question marks as the geographical stochastic distributions of the APs, the trustworthiness parameters of available APs, i.e. the pairwise historical experiences (a similar framework discussed in Chapter 2) and etc. may have a great impact on the decision. Nevertheless, we believe that the construction of collaboration request sequence should mainly be based on the geographical distribution of the APs, where the nearby collocutors can be determined by the RSSIs of the beacon frames. This is due to the fact that
3.3 Load Balancing in UCN in a Central Manner

In this section, we discuss a POMDP based load balancing strategy in UCN for the circumstances, where among two micro-providers only a single AP is capable of taking decisions for load balancing. This is due to the high variety of software capabilities among UCN-APs in the UCN (ref. to Section 1.1, where resource limited devices may participate with a provider role. In other words, the collocutor AP is non-collaborative, nevertheless is assumed to be manageable and support UCN communication framework, which is detailed in Section 1.1.

In the UCN environment the end user connectivity is provided by the short-range wireless access technology, i.e. the IEEE 802.11 based Wireless LAN (WLAN) standard as the de facto solution. The end-users may witness the rapid expansion of WLAN coverage by means of UCN coalescences at both residential and enterprise environments. This, in turn, enables the rapid expansion of WLAN coverage due to its ease of deployment, low cost, and ever-increasing data rates provided by the evolution in 802.11 family of protocols. On the other hand, the widespread

Figure 3.12: A Load Balancing Scenario by means of multiple interactions among APs
utilization of resource hungry applications over wireless networks, when combined with the high density of UCN sub-networks hurts the users’ experience, which is intensified by uncoordinated deployment and network operation. This bottleneck in the UCN environment is intensified not only when the proposed cooperative load balancing activities lacks but also when the resource limited UCN-APs are exposure to anomaly conditions. With the anomaly conditions, we refer to possible system failures, congestions or high bandwidth consumptions due to specific traffic activities of a certain number of clients in the UCN, which in turn bears resemblance of overload situations. Although the bandwidth utilization of this characteristics are investigated mainly in the network security literature under the denial of service (DoS) or specifically the reduction of quality (RoQ) type attacks \[131, 132, 133, 134, 135\], we refer these conditions simply as RoQ like anomalies as the DoS or RoQ type attacks are out of the thesis scope.

We introduce an integrated solution to congestion avoidance by means of a central load balancing activity and for a limited number of anomaly conditions. As illustrated in Figure 3.13, the load balancing capable UCN-AP monitors partially the traffic activities and system behavior of the collocutor AP in addition to itself. The proposed solution implements POMDP as the optimization tool for the strategy selection in the partially observable UCN domain. By successfully differentiating resource hampering anomalies from overload cases, the control system takes appropriate actions in order to prevent a disruption for the UCN network quality from the end users’ perspective. The proposed solution is fully implemented and experimented (ref. to Section 5, for which we present our observations and demonstrate the effectiveness of the system to detect and alleviate congestion situations.

For the proposed framework, the decision relies additionally on the trust level of the UCN stakeholders as will be detailed in the proceeding sections. The UCN-clients are key component of the resource allocation related decisions by means of facilitation through trusted feedbacks on the network performance. We propose increasing the impact of the end-users in addition to the state of art solutions for load balancing decisions. Observations and decisions for actions are performed remotely together with the help of network users. The proposed system is capable of sensing air traffic remotely and detecting all APs around the network, which gives the opportunity for researchers to add additional functionalities for the self-organization of UCN-APs in their coverages. This, additionally, prevents users from associating to misconfigured UCN-APs with the proposed client network manager program.
3.3 Load Balancing in UCN in a Central Manner

3.3.1 Solution Model

3.3.1.1 The POMDP Model

It should be underlined that the model we propose in the rest of this section is an outcome of our experimental studies and fine tuning of the involved parameters (states and transition probabilities, etc.) for intuitive behavior of the system. For clarity of presentation we directly provide the resulting state transition diagrams here without presenting the steps in the evolution of this model.

3.3.1.2 Observations

There are three observations. The Data traffic observation is done with patched airodump-ng tool [136] in order to sense high and low data traffic rates for the observed UCN-AP. To this end, the well equipped UCN-AP starts a virtual Wi-Fi interface in a monitoring mode and senses only the air traffic. Secondly, during SNMP check observation, the CPU load and MAC addresses together with the IP addresses of users attached to the observed AP is fetched in order to determine the softwares keeping CPU busy or anomaly resulting traffic activities. Final observation is the User network performance measurement, which is a core feature of the proposed system.
3 Cooperative Load Balancing in UCN among UCN-APs

3.3.1.3 States

In our proposed model, there are nine states, four of which are defined to be the main states; namely, OK, Congested, Anomaly and Critical System Failure. OK state represents the world state of a well performance working AP condition. In Congested state observed AP is congested giving rise to critical network experience for users. Anomaly state is another main world state representing the aforementioned RoQ like anomalies for the observed AP. Finally, the Critical System Failure state represents a failure case for the monitored UCN-AP or the failure of the UCN communication platform.

In addition to these four main states, five intermediate states are defined in order to provide our software with additional observation taking steps. The first intermediate state, namely the OK to Congestion ($O_C$), stands for the additional SNMP requests in addition to observed high data traffic and critical user network delay experience. Secondly, in Ok to Attack ($O_A$) state, data traffic check observation is done in addition to observed critical user network delay measurement and critical SNMP check observation. Ok to Critical System Failure ($O_S$) state is the third state for which SNMP related observations are performed in addition to observed non-critical data traffic and critical user network delay measurements. In Attack to Congestion ($A_C$) state, SNMP check observation is performed once again in addition to observed critical data traffic and user network delay experience. Finally, Attack to Critical System Failure ($A_S$) state is for data traffic check observation.

These additional states are introduced in order to differentiate the similarly characterized Congestion and Anomaly states, after the initial observation of an anomaly leading to those states.

3.3.1.4 Actions

No action is required for intermediate states during further analysis and OK state, which is also introduced as one of the four actions in our model. Secondly, Anomaly Response is defined to be command from controller AP on the monitored AP to drop packets once an untrusted client consumes the bandwidth highly and resulting critical network performances for the rest of trusted clients. Load Balancing is the third action where controller AP starts a network initiated handover process and commands on the user with worst network experience, i.e. the previously discussed performance indicators, to hand off another available AP. Finally, by Critical System Failure Report action, controller AP reports a critical system fail report to the owner.
3.3 Load Balancing in UCN in a Central Manner

3.3.1.5 Rewards

No action is rewarded highly for the OK state and intermediate states. On the contrary, it is a low rewarded action for anomaly, congested and critical system failure states. It is unnecessary to take an action for the desired world state and not feasible to take an action without increasing the belief of the states during intermediate states. Anomaly response is rewarded highly only for a possible anomaly state; however, it has a very low reward for a congested state because of the potential disturbance for the network clients, which gives rise to suffering users from the false alarms for anomaly precautions of the network. On the other hand, the load balancing is rewarded highly for congested states and very low for an anomaly case. This is because of the fact that an anomaly would spread easily among the UCN-APs in case of a false alarm for a congestion situation. With a similar manner, critical system failure report is rewarded highly for the corresponding states.

3.3.1.6 State transition functions

As observed from Figure 3.14, there is mainly no direct transition from OK state to any other state during analysis with the taken observations. No action maintains the belief state except for the intermediate states for which the uncertainty is eliminated by additional observations. Anomaly response results in a transition from $O_A$ and anomaly states to the OK state and it has almost no effect for the other states. Similarly load balancing results in a transition from intermediate $O_C$ state and congested state to OK state. This action is dangerous for the intermediate states between attacked and congested and results in a transition with a higher probability to the anomaly state. This is because of the characteristics of aforementioned anomalies, which throttle the network services rather than completely and resemble a congestion circumstance in the network. An attempt for the share of the load in an anomaly condition may spread the critical situation among UCN subnetworks. Finally, a critical system failure report results in a transition from critical system failure state to the OK state.

3.3.1.7 The POMDP based optimal Policy for the controller UCN-AP

We combine the aforementioned observation in order to simplify proposed POMDP model. In table 3.9, we provide the corresponding optimal policy for the proposed POMDP model.
Figure 3.14: State Transition Probabilities for No Action

Figure 3.15: State Transition Probabilities for Attack Response
3.3 Load Balancing in UCN in a Central Manner

Figure 3.16: State Transition Probabilities for Load Balancing

Figure 3.17: State Transition Probabilities for Critical System Failure Report
### Table 3.9: The POMDP based policy for controller UCN-AP

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>OK_ob</th>
<th>A_ob</th>
<th>C_ob</th>
<th>S_ob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>25</td>
<td>5</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>25</td>
<td>24</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>26</td>
<td>4</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>25</td>
<td>6</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>25</td>
<td>6</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>25</td>
<td>6</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>25</td>
<td>6</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>15</td>
<td>3</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>18</td>
<td>3</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>21</td>
<td>3</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>21</td>
<td>3</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>25</td>
<td>13</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>25</td>
<td>14</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>25</td>
<td>3</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>16</td>
<td>19</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>16</td>
<td>14</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>13</td>
<td>19</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>25</td>
<td>24</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>
3.3 Load Balancing in UCN in a Central Manner

3.3.2 Software Components

In our proposed solution, a network manager component is developed for the clients in order to corporate with the controller component on the access point. The system block diagram is given in Figure 3.18.

The central controller component is a multithread software that is responsible for performing observations, by opening a domain socket and communicating with the patched airodump-ng to count data packets through the observed AP with a specific BSSID and channel. Data count values are compared with a threshold periodically in order to set corresponding observation flags. The controller program also checks network performance related entries from the clients periodically and sets the critical observation flags for the POMDP engine when necessary. The CPU load is monitored for malicious programs and associated user’s IP and MAC addresses. Additionally, the traffic activities of the clients are monitored, which is then aggregated with the trust level of the known clients. This is due to previously mentioned RoQ similar anomaly cases. This program sets SNMP observation flags in case of: i) no response from observed AP, ii) critical CPU load, or iii) once the bandwidth utilization of the non-trusted clients are over a critical threshold, i.e., in case of continuous high bandwidth utilization, which disturbs network experience of trusted and collaborative UCN-clients.

The POMDP engine is the core thread of the controller component. This thread periodically collects the previously described observations from responsible threads, combines them for possible OK, anomaly, congestion or system failure observations, and takes an action accordingly.

3.3.3 Evaluation

We evaluate the performance of the proposed controller framework with three different network scenarios. We believe that the difficulties for conducting experiments with real wireless networks gives rise to the fact that majority of publications in this area are based on simulation results [138]. In order to evaluate our model in a physical system, we make partially use of the user centric wireless testbed (ref. to Chapter 5), for which we illustrate the corresponding components with the Figure 3.19.

In our test environment, we make use of two WLAN access points (AP). On AP1 we run the POMDP controller. The second AP is observed by the POMDP controller on AP1 and assumed to be resource limited. The POMDP controller is responsible for controlling both APs. We use six Linux-based notebooks running
Figure 3.18: System Block Diagram
multiple threads and emulating a large user population. The users are divided into traffic generating users, malicious users and trusted users. We use the distributed traffic generator D-ITG [54] to emulate additional users. D-ITG server resides on the application server and serves as a TCP traffic source. An independent D-ITG receiver runs for each user on the user laptops. The receivers request TCP packets that are exponentially distributed with a mean of 750 Bytes. The inter-arrival time of these packets are exponentially distributed with an average of 1 ms. After completion of packet download, D-ITG clients on the user side reports the experienced delay.

We observe that the reaction time of the system depends on:

- QoE of delay time observation period
- Data traffic observation period
- SNMP request observation period

It is possible to enhance the reaction time of the system by tuning the observation periods, however due to experimentation objectives and in order not to overload the network traffic with observations, we tuned the reaction time of the system to a slower rate.

### 3.3.3.1 Experiment Scenarios

We consider the following three scenarios to evaluate the performance of the control policy.
3 Cooperative Load Balancing in UCN among UCN-APs

3.3.3.1 Anomaly Case

We simulate an anomaly situation with the sudden appearance of 20 malicious flows hampering the bandwidth from one of the malicious clients. The motivation is to reduce quality of services provided by AP2, by means of numerous service requests. Our system is able to discern between this type of anomalies and a congestion case by the virtue of the comparing the performance related measurement entries and SNMP requests. We initiate two separate anomaly causing flows and observe the the delay experienced by a normal user in Figure 3.20.

The first anomaly causing flow is initiated at T2 instant and stopped at T3. At T3 we re-initiate the attack, and do not stop it for the rest of the experiment. The proposed framework is able to catch the anomaly situation at instant T4. The controller issues an anomaly response action at this instant, and commands the second AP to drop the non-trusted clients involved with the anomaly.

3.3.3.2 Congestion case

For emulating a congestion scenario, we associate five users to AP2 sequentially. In Figure 3.22 we plot the delay experienced by three users. Congestion starts after T4, when the fifth user enters the system. Immedi-
3.3 Load Balancing in UCN in a Central Manner

Figure 3.22: Congestion Scenario User Experiences w.r.t Time

Figure 3.23: Congestion Scenario Observed Data Rate w.r.t Time

ately at T5 this user is handed over to AP1, as a result of load balancing action. Nevertheless the AP1 is kept overloaded for a longer time, which leads the controller to force another client at T6 for the handover. We observe that the average delay measurements are kept under an acceptable threshold.

3.3.3.1.3 Critical System Fail case We emulate a critical system failure in the observed AP2, by initiating a simple Linux shell based fork bomb, that drives the CPU load of the AP up to 100%, therefore making it unresponsive. For anomaly and system failures, the POMDP controller gets a critical SNMP observation. Nevertheless in this case, the data traffic on the air interface is under an acceptable threshold although the AP1 does not respond to the client requests. At T1 a shorter duration system failure is started, which is stopped at T2 as depicted in Figure 3.25. At T4, we re-initiate the fork bomb and let it run until the end of the experiment. At T5, POMDP controller detects a system failure and reports it.
Figure 3.24: Critical System Failure Scenario User Experiences w.r.t Time

Figure 3.25: Critical System Failure Scenario Observed Data Rate w.r.t Time
3.4 Conclusion

In this Chapter, we discuss the cooperative load balancing in UCN among UCN-APs. The attention catching dimensions in this chapter are the questions "how one can motivate UCN-APs for sharing the load" and "how this load sharing can be performed in an intelligent and fair manner". To this end, we make use of POMDP based decision-making extensively. We additionally introduce a load balancing approach for a specific technical use case as an exceptional situation where certain UCN-APs are not equipped with collaboration units. We present the validation of the presented approaches by means of real implementations.

We envision the discussed collaborative load balancing approach as a potential strategy for the load balancing attempt. This strategy is easily extended to be utilized for admission control (AC) based load balancing approaches, the details of which are provided later in the next chapter. Having produced effective strategies in this chapter, next we discuss how the UCN-APs can be equipped with the autonomic ability for the load balancing. The next chapter provides a potential answer to this question, where we enrich the strategy repository of the UCN-APs with additional approaches in the literature and deal with the autonomous load balancing.
4 An Autonomous Framework for Load Balancing in UCN

A generic definition of congestion or an unbalanced load among the access points may not be convenient for the user centric networking. This sophistication stems mainly from the entity Wi-Fi performances and the personal preferences of the UCN stakeholders. Besides, the highly dynamic characteristics of the UCN environment raises the complexity for potential solutions to balance the load among access units. Certain determinants for this sophistication are i) stochastic coalescences of UCN communities in different day-time and the public areas, ii) different Wi-Fi capabilities and configuration support of the UCN-APs, iii) the mobility patterns of UCN stakeholders and iv) immensely alternating traffic behavior and application requirements of UCN clients. Hence there is enough motivation for the introduction of an autonomous system for the UCN-APs, where iterative strategy coalescences are formed i) targeting a balanced load and ii) depending on different environmental characteristics.

In this chapter, we detail the autonomous behavior for the previously discussed corner stones of load balancing in UCN. Our proposed framework is based on an autonomous resource allocation and strategy optimization (ARASO) model, which is studied under the light of a more comprehensive perspective for the wireless networks. This control framework aims at immensely dynamic, time varying, and less predictable network environments. The very first goal of this approach is to eliminate redundant resource utilization for specific network objectives while reducing the network response time. To this end, the decision-making engines of the control framework characterize the network environment in terms of the correlation with specific network objectives and heal the resource utilization accordingly. The second goal is to match network strategies with different environmental conditions in terms of impact rates on the environment. With the impact rate on the environment, we mean the success rates of the taken strategies in healing the corresponding critical network situations. We make use of dynamic POMDP (Partially Observable Markov Decision Processes) based control loop to improve the network strategies against alarm conditions. The control framework is implemented on IEEE802.11
4 An Autonomous Framework for Load Balancing in UCN

WAPs and is validated by means of real implementation and simulations of different network scenarios.

This Chapter is comprised of two main sections. In Section 4.1, we introduce our ARASO approach. This approach is applied for the special use case, i.e. the load balancing problem in UCN, which is detailed in Section 4.2. By the application of the framework, we mean the re-adjustment of the system parameters in order to address load balancing. In this chapter, we aggregate previously detailed load balancing strategies together with additional approaches in the literature and verify adaptation ability of the control framework to the changing environmental conditions in the highly stochastic and dynamic UCN domains.

4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

The comprehensive and large-scale organizational attempts on the conceptual and standardization level for the autonomic network architectures (ANA) have recently been placed in the limelight for the future Internet (FI) literature. Although clean slate or evolutionary approaches are the moot point between recent proposals, researchers agree on increasing the autonomic capability of the network with learning and the decision making engines (DMEs). To this end, different disciplines in the literature of artificial intelligence techniques are introduced in order to increase the intelligence level of the DMEs in the network.

IBM’s control loop [139] has been extensively studied in the FI literature as a technique for the autonomous behavior in terms of the analysis and aggregation of raw data and hence making efficient decisions. Researchers mainly focus on the periodic execution of control loops by means of gathering sensor data, the analysis and the aggregation of collected raw data, the reasoning and decision making, and finally the execution. However, for the realistic scenarios, this process may result in redundant data gathering and analysis under normal circumstances. In other words, an optimization mechanism for the activation instances or the duration of autonomous functionality may improve the resource utilization in the network. Additionally, conflict free architectures [140, 141] are studied in the literature for the harmonization of different solution approaches for similar complexities. This allows the network hold various strategies in an algorithm repository. We believe an evaluation on the impacts of the executed algorithms, based on different environmental conditions, may heal the network strategies.

Before we introduce our approach addressing the above-mentioned goals, we pro-
vide a literature review of autonomous network proposal in Section 4.1.1, where we mainly focus on large-scale FI proposals. We provide our approach together with the problem statement in Section 4.1.2. Later, we detail the technical aspects of the proposed approach in Section 4.1.3, whereas the implementation details are discussed in the proceeding section, namely Section 4.1.4. We comment on the experiments and test results in Section 4.1.5. Finally we evaluate our approach in Section 4.1.6.

### 4.1.1 Related Work

An ongoing research for the standardization of autonomic future Internet architecture is conducted by ETSI [140, 141]. The main motivation of the standardization effort is the harmonization of proposals by means of providing a roadmap for the corresponding studies and preventing conflicts among them. The GANA (Generic Autonomic Network Architecture) [142, 143] is proposed as a fundamental NA (Network Architecture), which mainly investigates the conflicts raised due to the different objective related decisions among the network entities. To this end, three different types of inter-relations are defined, namely, hierarchical, peering and sibling based on four hierarchical levels of abstractions. A similar generic NA approach, the Autonomic Network Architecture (ANA) [144], intends on the dynamic adaptation and self-organization depending on end user preferences.

In FOCALE [145, 146] (Foundation Observation Comparison Action Learn Reason) architecture, the network is composed of managed elements (ME) and autonomic management elements (AME), which are responsible for monitoring the current states of MEs, comparing with the desired states, and taking actions respectively. The AME units ensure the network policy by continuously executing FOCALE control loop based on IBM’s control loop [139] in order to capture and analyze the current status of the network. To this end, nested control loops are implemented throughout the network in a hierarchical way. The control loops are mainly classified as top (maintenance) loop for detecting anomalies and bottom (adjustment) loop for reconfiguration due to anomalies or business goals. Depending on the context, policies and semantics of the interpreted data, FOCALE architecture proposes changing functionalities of control loops.

In the Self-Net project (Self-Management of Cognitive Future InterNET Elements) [147, 148], distributed cognitive cycles are proposed for system and network management (DC-SNM) in a hierarchical manner. The aim at facilitating the promotion of distributed management. The management and control functions are distributed among the network elements in order to achieve faster decision and execution with continuous local optimization and knowledge building capabilities. Hints, recom-
recommendations or requests are disseminated among the hierarchical levels in order to indicate the change in the network administration decisions, i.e., policies. Two different types of agents are defined, namely, Network Element Cognitive Manager (NECM) as the local agents on the network entities and Network Domain Cognitive Manager (NDCM) as the domain agent orchestrating the local agents.

In SOCRATES (Self-optimization and self-organization in wireless networks) [149], a bottom to top vision is presented with the key objectives of operational expenditure reduction, enhancement of network coverage, resource utilization and conflict management due to resource allocation or service quality optimization. Conflict cases are simulated providing reasoning models for the control engines. Additionally, a subset of use cases are covered [150] and is implemented on the 3GPP E-UTRAN access technology. The main focus of the SOCRATES approach is to achieve O/CAPEX (operational and capital expenditures) for a SON (Self-organizing networks) application by reducing human intervention to the network. Detailed information on the chosen architecture types with respect to the use-cases can be found in [151, 152].

In addition to the organizational and large-scale projects, there exist many significant small-scale contributions to the autonomic network control and management. Costa et al. [153], focus on alarm correlation and root cause analysis, by means of a rule-management module, which is based on data mining for rule generation and reinforcement learning. Basically, historical alarm data is stored inside local databases and correlation rules are extracted periodically by alarm management system. Famaey et al.[154], propose a generic NA in which the distributed autonomic components are structured in a hierarchical manner in order to decrease the complexity of component interactions. The network overhead associated with management and control is reduced by grouping autonomic management components into a hierarchy. To this end, the dissemination of network context and policies are efficiently orchestrated. Lim et al. [155], suggests a decentralized self-organizing network management and control system based on concept of navigation patterns. Graph traversal algorithms are the core mechanism that control the execution of management operations. Moreover, graph traversal algorithms are used to control and coordinate the dissemination, analysis and aggregation of management information among the network.

A randomization-based control and management approach is studied in [156]. Brunner et al. studies probabilistic management paradigm in order to cope with the overhead of redundant information gathering and processing in dynamic and unpredictable environments. Basically, dynamic behavior inside the network results in a lack of real time view of system and prevents coordinative functionality especially
when the complexity of the network environment is high. In this study, it is claimed that the cooperation based decentralized control and management functionalities might fail in real dynamic environments while trying to assure the synchronization among the control entities. Due to these reasonings, a random activation of management and control functionalities is proposed.

The multi-agent systems (MAS) are widely issued in the autonomous network literature. In [157], Jones et al. suggest a MAS, which bears features of scalability, interoperability, survivability, concurrent diagnosis and anomaly detection capabilities focused on accurate alarm rate. The semi-autonomous agents have their own world model consisting of actions, history of past actions for further enhancement and thumbprints blocks in order to dynamically change world-detection model for adaptation. In [158], Timon et al. propose mobile agents for efficient distributed network control and management. The mobile agents, which carry management or control tasks, are delegated to remote hosts to execute the task and carry the measurements back to the sender hosts. A similar agent-delegation approach is studied in [159].

4.1.2 Our Approach

Before we state our approach, we provide brief information for the environment description and problem statement. After these motivation parts, we state our solution approach.

4.1.2.1 Environment Description

This work mainly aims at an immensely dynamic, time varying, and less predictable network environment. The motivation for this consideration comes from the fact that the proposed framework targets the wireless networks, which exhibit dynamics caused by the wireless medium, clients’ behavior, traffic demands, stakeholders’ preferences, and telecommunication infrastructure deployment, etc.

The behavior of focused network context is unsteady due to the time-varying interference degree causing different packet loss rates, shortcomings of network protocols in wireless medium, contingent hardware or software related failure of network elements or decision making entities, unpredictable traffic behavior of clients, alternating demand on different network objectives and etc. Additionally, in a realistic network scenario; i) the perceptual aliasing (same observations might show up in different states) ii) noisy/faulty sensors or iii) a combination of both situations result in partial and unreliable information for reasoning [91]. Finally, the dynamics of the
system may not be captured in real time owing to the problem of recency. Hence the reliability of information and thus the decisions may rely on partial information.

4.1.2.2 Motivation and Problem Statement

For each network objective, an extensive range of strategies are proposed in the literature based on a set of assumptions describing different network settings. An autonomous network approach requires an integration of these strategies supported by evidence and the execution in accordance with a match of strategy-observation pairs. In other words, an autonomous control system should heal the strategy by choosing algorithms, which have more impact on specific alarm conditions.

We believe that autonomous network agility in strategy optimization might come from tracking impact rate of executed algorithms based on triggering factors. We envision triggering factors as an indicator of instantaneous environmental conditions. Due to unreliable partial information, however, false triggers might not necessarily define an alarm condition in the network or indicate a less impact of an executed strategy.

Owing to the fact that decision instances are sensitive to execution time dictates that timely decision(s) should be made i.e., waiting long for all the decisions’ required inputs should be avoided and take the decision on partial/inferred observations. In other words, there may be decision instances, which if not taken quickly, may consequence in an unstable system. Thus one potential solution in this connection would be to avoid waiting long for all the decisions’ required inputs and take the decision on partial/inferred observations. Intuitively, such decisions may not be graded as optimal ones, however, literature encamp a number of well known approaches (e.g., partially observable MDP, Bayes’ model and or various learning approaches), which have been used extensively in the research literature in different settings. We envision the network objectives as characteristics of demand and supply. Hence, defining instantaneous environment characteristic in terms of demand for each specific network objective and dynamically prioritizing various objectives may help optimizing the network resource utilization and network response time.

4.1.2.3 Solution Statement and Our Approach

We propose a two-level learning mechanism targeting the aforementioned problems. As represented in Figure 4.1, the lower level mechanism characterizes the network environment in terms of demand and assigns relative priorities to each network objective. Hereafter, this process is referred as network objective prioritization.
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

The higher level mechanism; i) associate different algorithms specific to each objective, ii) helps in optimization of objective specific network strategy by tracking impact rate of each algorithm on a specific environmental condition and finally iii) handles false triggers. Hereafter, we refer this higher level learning mechanism as **strategy optimization**.

### 4.1.3 Technical Details of the Proposed Model

In this section the proposed solution is detailed. As a formulation of network policy, we suggest to use the ontology, as a knowledge platform for reasoning, relationships and restrictions [160, 161] in the network. Ontology is defined as the formalization of the agreement on shared knowledge or conceptualization in a domain in order to represent particular domain theories and provide an agreed ground of reasoning for
domain circumstances according to agent beliefs and actions [160]. The common knowledge platform for reasoning, relationships and restrictions might be utilized as control policies for decision-making engines in the network. Nejdl et al. [161], stated that the usage of ontology eases dealing with the complexity of many policy-management functionalities and the software where the intelligent agents might reuse and share the knowledge. The details are provided in the following sections.

4.1.3.1 Ontological Characterization of Network Policy

Ontology is defined as the formalization of the agreement on shared knowledge or conceptualization in a domain in order to represent particular domain theories and provide an agreed ground of reasoning for domain circumstances according to agent beliefs and actions [160]. The common knowledge platform for reasoning, relationships and restrictions might be utilized as control policies for decision-making engines in the network. Nejdl et al. [161], stated that the usage of ontology eases dealing with the complexity of many policy-management functionalities and the software where the intelligent agents might reuse and share the knowledge.

We model the ontology graph of the network policy in a gradual manner defining network objectives. The graduation is performed based on relations between the comprehensive network objectives, their subbranches, corresponding triggers (observations) and finally the relevant actions (proposed algorithms as solutions). Based on the entity type and its deployment in the network, the related assignments are restricted to certain branches, which is driven by the general network policy. In order to provide a better understanding of the proposed network control approach, and as a sample case, we focus on an access point (AP) of which responsibility branches are defined as illustrated in Figure 4.2. However, it should be noted that the proposed control framework is generalized in the proceeding sections.

4.1.3.2 Network Objective Prioritization

Marked Markov Arrival Processes are utilized in order to construct and detect state transitions of the CTMC for each objective and observation. The critical observations in specific repositories are mapped over special events of arrivals in the MMAP models. The arrival rates are estimated using expectation maximization (EM) algorithm and MMAP model is updated respectively. Prioritization is performed based on the steady-state probability distributions of environment states. The higher steady-state probability of critical environmental states for each objective implies a demand from the environment on the prioritization of the corresponding objective or the observation.
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

In the following section, we provide technical details and mathematically formulate the network objective prioritization.

4.1.3.2.1 MMAP Construction We introduce three system states, these states are characterized by system objective functions. On an abstract level, we assume that all the objective functions may be encamped in any of the proposed three states. The common three-state CTMC for level-2 objective and observations in corresponding level-3 observation repositories is depicted in Figure 4.3.

Each critical observation instance is a different type of arrival event and is analyzed independently in the observation repository \( \Omega_{o1,o2,o3} \). Here \( \Omega_{o1,o2,o3} \) is defined
Figure 4.3: State Transition Diagram of the Environment

as the observation repository for level-3 objective \( o_3 \), which belongs to level-2 objective \( o_2 \) and level-1 objective \( o_1 \). Each arrival results only in a transition from less critical state to a more critical one. Finally an imaginary event is defined causing transitions between any state. This is due to false trigger like problems and the possibility of inadequacy for the defined observations in covering all the network dynamics for the related objective.

We define a homogenous distribution (represented by \( \pi \)) on the CTMC states as given in Eq. 4.1 due to inadequate historical observations at the boot-up time.

\[
\pi = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix} \quad (4.1)
\]

The \( D_0 \) Matrix, in Eq. 4.2, defines the parameters for each state transitions without an arrival and rates of each transition from state \( i \) to \( j \) is defined as \( \lambda_{ij} \).

\[
D_0 = \begin{pmatrix} -\lambda_1 & \lambda_{12} & \lambda_{13} \\ \lambda_{21} & -\lambda_2 & \lambda_{23} \\ \lambda_{31} & \lambda_{32} & -\lambda_3 \end{pmatrix} \quad (4.2)
\]

Due to transition characteristic of imaginary event, the state transition rates are equal: \( \lambda_{12} = \lambda_{13} = \lambda_{21} = \lambda_{23} = \lambda_{31} = \lambda_{32} = \lambda \).

For a single arrival \( Ob_x \) (representing the \( x^{th} \) observation in \( \Omega_{o_1,o_2,o_3} \)), the \( D_{Ob_x} \) is defined in the form of:

\[
D_{Ob_x} = \begin{pmatrix} 0 & \gamma_{Ob_x} & 0 \\ 0 & 0 & \gamma_{Ob_x} \\ 0 & 0 & 0 \end{pmatrix} \quad (4.3)
\]

where \( \gamma_{Ob_x} \) defines rate of the arrival. As it is not possible for an arrival to cause a transition from lower critical state to a more one, corresponding parameters are set to 0.

The final generator matrix \( D \) is in the form:
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

\[ D = D_0 + \sum_{Ob_i \in \Omega_{o_1,o_2,o_3}} D_{Ob_i} \]  \hspace{1cm} (4.4)

Using Eq. 4.4 and for \( i \in \Omega_{o_1,o_2,o_3} \) the final minimal generator matrix is:

\[
D = \begin{pmatrix}
-2\lambda - \sum_i \gamma_{Ob_i} & \lambda + \sum_i \gamma_{Ob_i} & \lambda \\
\lambda & -2\lambda - \sum_i \gamma_{Ob_i} & \lambda + \sum_i \gamma_{Ob_i} \\
\lambda & \lambda & -2\lambda 
\end{pmatrix}
\hspace{1cm} (4.5)

Under the constraints for the steady-state probability distributions \( \pi.D = 0 \) and \( \pi.(1 \ 1 \ 1)^T = 1 \), the steady state probability distribution of CTMC are given as:

\[ P\{Non - Critical\} = \frac{\lambda}{3\lambda + \sum_{i \in 1} \gamma_{Ob_i}} \]  \hspace{1cm} (4.6)

\[ P\{Normal\} = \frac{3\lambda^2 + 2\lambda \sum_i \gamma_{Ob_i}}{(3\lambda + \sum_i \gamma_{Ob_i})^2} \]  \hspace{1cm} (4.7)

\[ P\{Critical\} = \frac{3\lambda^2 + 3\lambda \sum_i \gamma_{Ob_i} + (\sum_i \gamma_{Ob_i})^2}{(3\lambda + \sum_i \gamma_{Ob_i})^2} \]  \hspace{1cm} (4.8)

When it comes to the observation prioritization, we use similar methodology where a single arrival exists. The interpretation of these results are:

- Setting larger \( \lambda \) value would set a static priority for each level-2 objective or observations as the steady-state probabilities becomes homogenous.

- Smaller \( \lambda \) parameters would leave the prioritization of objectives or observations totally to the environmental behavior.

- Different \( \lambda \) values on different concepts results in a predetermined prioritization of specific concepts against others. In other words the network administration might guarantee specific services independent from the environment.

- The increase in the number of observations in a specific repository would provide the control framework with stronger prediction capability for the environment and network conditions.

4.1.3.2.2 Arrival Rate Estimation  \hspace{1cm} The arrival events are counted periodically as shown in Figure 4.4, where \( w(n) \) represents the \( n^{th} \) window.

Referring to the Figure 4.4, the control agent utilizes a fixed-size sliding window and counts the number of arrivals at each window in order to estimate rate of arrivals.
As this process is a basic counting process, we mainly base our assumptions about the arrival rate distribution on Poisson distribution. The arrival rate estimation is carried out using EM algorithm as described before in this thesis. In accordance with the three-state formulation of CTMC for each level-2 objective and corresponding observations, and due to the fact that Poisson mixtures fit the observations in a much better way in case of multiple routines [130], the EM algorithm is applied on a mixture of three Poisson distributions. The basic mixture form for the probability of counting \( k \) number of the same observation in a fixed window is: (given the set \( \Theta = (\gamma_{Ob_1}^{(nC)}, \gamma_{Ob_1}^{(nC)}, \gamma_{Ob_1}^{(N)}, \gamma_{Ob_1}^{(C)}, \gamma_{Ob_1}^{(C)}), \) where \( \gamma \)'s are defined as the arrival rates and \( \tau \)'s represent the instant probability distribution of states)

\[
p(k|\Theta) = \sum_{s \in \{nC, N, C\}} \tau^{(s)}_{Ob_1} e^{-\gamma_{Ob_1}^{(s)}} \frac{\gamma_{Ob_1}^{(s)}^k}{k!} \tag{4.9}
\]

In this model, a case of an increase in the probability distribution of a specific environment state, represents a transition to the corresponding state. Hence the corresponding arrival rate is forwarded to the upper level. The general algorithm in this phase is presented as.

### 4.1.3.3 Strategy Optimization

It is the higher level mechanism, in which the control loop for each objective are executed in case of a trigger from lower level objective prioritization process. The control loops are modified based on POMDP. With the modified control loop different strategies are associated with various environmental conditions by monitoring the success rates. In the proceeding sections, we briefly discuss the construction of proposed control loops.
Algorithm 1 Arrival Rate ($\lambda$) Estimation Algorithm

\begin{algorithmic}
\STATE $N = \text{PRE_DEFINED\_WINDOW\_SIZE}$;
\STATE $\lambda_1 = \text{CountArrival}[w(1)]$;
\STATE $\lambda_2 = \text{CountArrival}[w(2)]$;
\STATE $\lambda_3 = \text{CountArrival}[w(3)]$;
\STATE $\lambda_{nC}^{(1)} = \text{getMin}(\lambda_1, \lambda_2, \lambda_3)$;
\STATE $\lambda_C^{(1)} = \text{getMax}(\lambda_1, \lambda_2, \lambda_3)$;
\STATE $\lambda_N^{(1)} = \text{getMid}(\lambda_1, \lambda_2, \lambda_3)$;
\STATE $\tau_{nC} = 0.34$;
\STATE $\tau_N = 0.33$;
\STATE $\tau_C = 0.33$;
\STATE $K = \text{createNewArray}[N]$;
\STATE $\Theta = (\lambda_{nC}, \tau_{nC}, \lambda_N, \tau_N, \lambda_C, \tau_C)$;
\STATE $i = 0$;
\REPEAT
\STATE $i = i + 1$;
\STATE $\lambda = \text{CountArrival}[w(n)]$;
\STATE appendFromRight($K$, $\lambda$);
\STATE $\Theta_{\text{temp}} = \Theta$;
\STATE $\Theta = \text{expectationMaximization}(K, \Theta_{\text{temp}})$;
\STATE $\tau_1 = (\tau_{nC} - \tau_{\text{temp}})$;
\STATE $\tau_2 = (\tau_N - \tau_{\text{temp}})$;
\STATE $\tau_3 = (\tau_C - \tau_{\text{temp}})$;
\STATE $\tau_{\text{temp2}} = \text{getMax}(\tau_1, \tau_2, \tau_3)$;
\IF { $\tau_1 = \tau_{\text{temp2}}$ }
\STATE forwardToUpperLevel($\lambda_{nC}$);
\ELSIF { $\tau_2 = \tau_{\text{temp2}}$ }
\STATE forwardToUpperLevel($\lambda_N$);
\ELSE
\STATE forwardToUpperLevel($\lambda_C$);
\ENDIF
\UNTIL { $i = N - 1$ }
\WHILE { True }
\STATE $i = i + 1$;
\STATE $\lambda = \text{CountArrival}[w(n)]$;
\STATE shiftLeft($K$);
\STATE appendFromRight($K$, $\lambda$);
\STATE $\Theta_{\text{temp}} = \Theta$;
\STATE $\Theta = \text{expectationMaximization}(K, \Theta_{\text{temp}})$;
\STATE $\tau_1 = (\tau_{nC} - \tau_{\text{temp}})$;
\STATE $\tau_2 = (\tau_N - \tau_{\text{temp}})$;
\STATE $\tau_3 = (\tau_C - \tau_{\text{temp}})$;
\STATE $\tau_{\text{temp2}} = \text{getMax}(\tau_1, \tau_2, \tau_3)$;
\IF { $\tau_1 = \tau_{\text{temp2}}$ }
\STATE forwardToUpperLevel($\lambda_{nC}$);
\ELSIF { $\tau_2 = \tau_{\text{temp2}}$ }
\STATE forwardToUpperLevel($\lambda_N$);
\ELSE
\STATE forwardToUpperLevel($\lambda_C$);
\ENDIF
\ENDWHILE
\end{algorithmic}
4 An Autonomous Framework for Load Balancing in UCN

4.1.3.3.1 Control Loop  The construction and conceptual background of the IBM generic control loop is discussed in [162]. Based on this, we illustrate in Figure 4.5 illustrating the aggregation of the proposed mechanisms into the generic control loop. The first stage mechanism, i.e. the network objective prioritization, runs at the measurement collection and the analysis steps. Depending on the objective observation repository, the analysis is performed and in case of a critical instance, corresponding triggers are forwarded to the higher mechanism, i.e, the strategy optimization. The proposed strategy optimization mechanism has an impact on the decision making and the execution steps, by means of the selection of corresponding network actions. Moreover, we consider on potential supplementary cycles between decision-making, execution and the measurement taking for the further analysis on the executed actions.

Figure 4.5: Positions of the Proposed Mechanisms in the Generic Control Loop

We propose mainly an elimination mechanism on the network action repository by means of selecting the actions having more impact on the specific network objective tackle. To this end, we propose monitoring the success rates of the executed actions in order to determine potentially effective actions for an existing problem in the network. Having these objectives we illustrate the Figure 4.6 as the basic methodology loop for the analysis of the executed actions, which helps for the strategy optimization. The continuos lines stands for the straightforward transitions between iterative steps to be taken for the control loop. We comment on the dashed lines representing the additional transitions in proceeding parts of this section.

The description of the states represented in Figure 4.6 is as follows:
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

Figure 4.6: Basic Control Loop States

- **Wait for trigger state**: In case of a critical observation the lower level objective prioritization phase forwards an execution trigger together with the information model of level-2 objective and corresponding observation.

- **Choose action and take observation**: Based on the observation-algorithm match, a more successful algorithm is selected for execution. More details on the selection of algorithms is provided later in this section.

- **Execute state**: The agent executes chosen algorithm.

- **Take observation state**: After the execution state, a pre-defined time period is waited for the stabilization of network under the impact of executed algorithm and an additional observation is taken in order to observe judge on success rate of the executed algorithm.

- **Positive evaluation state**: The success rate of executed algorithm is increased.

- **Negative evaluation state**: The success rate of executed algorithm is decreased. It should be highlighted that this state is a normalization state for the elimination of algorithms at state $S_4$. In other words, the relative probability over the executed action for being selected at state $S_4$ is increased for remaining available actions.

- **Finish execution loop trigger state**: The control loop hands the token to the lower level objective prioritization mechanism.
4 An Autonomous Framework for Load Balancing in UCN

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A^{(ij)}$</td>
<td>Action set for third and corresponding second level objective $i, j$ respectively, with a size of $K$</td>
</tr>
<tr>
<td>$Ob^{(ij)}$</td>
<td>Observation set for third and corresponding second level objective $i, j$ respectively, with a size of $M$</td>
</tr>
<tr>
<td>$a_k^{(ij)}$</td>
<td>The $k^{th}$ action in $A^{(ij)}$ ($a_k^{(ij)} \in A^{(ij)}$)</td>
</tr>
<tr>
<td>$o_m^{(ij)}$</td>
<td>The $m^{th}$ observation in $Ob^{(ij)}$ ($o_m^{(ij)} \in Ob^{(ij)}$)</td>
</tr>
<tr>
<td>$\Theta^{(ij)}_m$</td>
<td>{($\gamma^{(nC)}$, $\tau^{Ob_m}$, $\gamma^{(N)}$, $\tau^{Ob_m}$, $\gamma^{(C)}$, $\tau^{Ob_m}$)} set for $o_m^{(ij)}$, determined by the network objective prioritization stage</td>
</tr>
<tr>
<td>$tr^{(ij)}_m$</td>
<td>The $m^{th}$ observation trustworthiness</td>
</tr>
<tr>
<td>$\rho^{(ij)}_k$</td>
<td>Success rate for the action $a_k^{(ij)}$</td>
</tr>
<tr>
<td>$\gamma^{(ij)}_m$</td>
<td>Expected rate parameter of observation $o_m^{(ij)}$</td>
</tr>
<tr>
<td>$t_{ij}^{(ij)}$</td>
<td>The state transition probability from $S_1^{(ij)}$ to $S_j^{(ij)}$, where $S_2^{(ij)}, S_3^{(ij)} \in S^{(ij)} = {S_1^{(ij)}, \ldots, S_7^{(ij)}}$</td>
</tr>
<tr>
<td>$W$</td>
<td>Window size for arrival observations</td>
</tr>
<tr>
<td>$o_{m,ij}$</td>
<td>Critical instance of $o_m^{(ij)}$ observation</td>
</tr>
<tr>
<td>!$o_{m,ij}$</td>
<td>non-Critical instance of $o_m^{(ij)}$ observation</td>
</tr>
<tr>
<td>$a_k^{(ij)}$</td>
<td>The success instance of $a_k^{(ij)}$</td>
</tr>
</tbody>
</table>

Table 4.1: Summary of Parameter Set for the Control Loop Dynamics

It should be underlined that this improved control loop represents a basic approach for the network environments, where the complete knowledge on the world states, i.e. the state indicators or the network observations, is available. Nevertheless, the proposed model attempts for very dynamic and stochastic environments, where only partial information on the corresponding network states and the transitions may be available. This is due to the previously mentioned reasonings such as false alarms, recency complications, unreliable or incomplete information, sensitivity for the execution times and network stabilization durations, etc. Thus further analysis is required for the proposed control loop by means of additional optimization attempts for an optimal policy decision under uncertainty.

A summary of the parameter set for the control loop dynamics is provided in Table 4.1, which have a great impact on the transitions between control loop states. Considering a specific loop for a specific objective (third and corresponding second level objective $i, j$), a critical instance for any observation ($o_m^{(ij)} \in Ob^{(ij)}$) generates a trigger for the control loop, which is represented as $o_{m,ij}$ in Figure 4.6. Thus state transition rate from the first state to the second one depends on the total critical instance rates for corresponding objective observations:
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

\[ t_{12}^{(ij)} = P\{Trig.\} = \sum_{(o_m^{(ij)} \in Ob^{(ij)})} P\{o_m^{(ij)} \text{ is selected, } a_m^{(ij)}\} \]

\[ = \sum_{(o_m^{(ij)} \in Ob^{(ij)})} P\{o_m^{(ij)}|a_m^{(ij)}\} P\{a_m^{(ij)}\} \]

\[ = \sum_{m=1}^{M} \left( \frac{\gamma_m^{(ij)}}{\sum_{m=1}^{M} \gamma_m^{(ij)}} \right) \gamma_m^{(ij)} W \]

where,

\[ \gamma_m^{(ij)} = \gamma_{Ob_m}^{(nC)} \tau_{Ob_m}^{(nC)} + \gamma_{Ob_m}^{(N)} \tau_{Ob_m}^{(N)} + \gamma_{Ob_m}^{(C)} \tau_{Ob_m}^{(C)} \]

As an additional observation is taken at \( S_2^{(ij)} \), the \( t_{12}^{(ij)} = t_{12}^{(ij)} \) and \( t_{27}^{(ij)} = 1 - t_{23}^{(ij)} \) as no other state transition is allowed. The \( t_{34}^{(ij)} = 1 \). By separating the \( S_3^{(ij)} \) and \( S_4^{(ij)} \) we represent the required time duration for the network stabilization depending on the sequential form of the selected action and observation tuples. One may expect a high dependency on the selected action at \( S_2^{(ij)} \), e.g \( a_k^{(ij)} \), and its success indicator, i.e the \( \rho_k^{(ij)} \), for the transition character from \( S_4^{(ij)} \) to either \( S_5^{(ij)} \) or \( S_6^{(ij)} \). In other words:

\[ t_{46}^{(ij)} = \sum_{(a_k^{(ij)} \in A^{(ij)})} P\{a_k^{(ij)} \text{ is selected, } a_k^{(ij)} \text{ is successful}\} \]

\[ = \sum_{(a_k^{(ij)} \in A^{(ij)})} P\{a_k^{(ij)}|a_k^{(ij)}\} P\{a_k^{(ij)}\} \]

\[ = \sum_{k=1}^{K} \left( \frac{\rho_k^{(ij)}}{\sum_{k=1}^{K} \rho_k^{(ij)}} \right) \rho_k^{(ij)} \]

\[ t_{45}^{(ij)} = 1 - t_{46}^{(ij)} \]

The positive / negative evaluation states are the most significant states for the proposed control loop, where two factors are of critical importance. These factors are i) the impact of the selected action and ii) the dynamic and stochastic behavior of the environment with an incomplete feedback. At the first glance, there is a great uncertainty on the additional observation at state \( S_4^{(ij)} \) due to the ambiguity for the trustiness of the observation, e.g. whether a critical observation instance stems from the less impact of the executed action or from the dynamics of the network environment. Nevertheless, this complexity is the focused item of the improved
control loop in this paper. For a better understanding of the remarked ambiguity, we focus on a network scenario related with resource allocation.

Considering a congestion case in the network due to the high client arrival rates, who have high bandwidth requirements; we assume that a load balancing activity is executed by the control loop on an access point of the network, which is based on forcing client migrations. Due to higher bandwidth utilizations and arrivals, the additional observation at $S^{(ij)}_4$ becomes critical despite the executed load balancing action. This situation resembles the complexity which is described with the aforementioned ambiguity. On the other hand, one may argue that the proposed algorithm is not effective for the predefined environment, which is a cut for the admission control based approaches rather than forcing client migrations. Thus the proposed control loop eliminates a migration based load balancing action not because of the ineffectiveness but because of the instant environment dynamics.

The very simple additional parameter, which is left as a design parameter, is the trustworthiness of the observations, which describe the transitions between states $S^{(ij)}_5$ to $S^{(ij)}_2$, $S^{(ij)}_5$ to $S^{(ij)}_7$, $S^{(ij)}_6$ to $S^{(ij)}_2$ and finally $S^{(ij)}_6$ to $S^{(ij)}_7$. With the trustworthiness, we define a subjective term for a specific observation class, which defines i) how this specific observation is capable of handling recency problems ii) how certainly can it be utilized as a reasoning in describing the network environment iii) how frequently the observation may create false triggers, etc. A useful discussion on trustworthiness of an observation can be found in [163], which is indirectly related with these items. At this point, It is worth mentioning that although subjective probabilities may be assigned in an intuitive way for the state transition probabilities in the construction of the system, the validation of the assigned probabilities is as discussion point as they may be descriptively inaccurate especially when the complexity of the problem is high [100]. We comment more on this parameter in Section 4.1.3.4.

\[
t^{(ij)}_{62} = t^{(ij)}_{52} = 1 - \frac{1}{M} \sum_{m=1}^{M} tr^{(ij)}_m
\]

\[
t^{(ij)}_{67} = t^{(ij)}_{57} = \frac{1}{M} \sum_{m=1}^{M} ts^{(ij)}_m
\]

Clearly, the transition parameters dynamically changes based on the environmental conditions. Additionally these parameters are to the objective and more concretely the corresponding action / observations. Nevertheless, considering on the instant shots of the proposed control loop, the policy state transitions hold memo-
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

ryless property as discussed before, i.e. the future state transitions depend only on
the current state of the loop. In other words the proposed policy loop forms a Markov
Chain with the specifications such as i) the world state set \( S^{(ij)} = \{ S^{(ij)}_1, \ldots, S^{(ij)}_7 \} \),
ii) the previously discussed instant state transition probabilities \( T^{(ij)} = \{ t^{(ij)}_{ij} | j \neq i \}, 1 \leq i, j \leq 7 \}, \) iii) the start state is \( S^{(ij)}_1 \).

4.1.3.3.2 POMDP Formulation

The main motivation behind using the POMDP
rather than Markov Decision Process (MDP) is the previously mentioned trustwor-
thiness of the observation set. An MDP framework is represented with the form
of \( < S^{(ij)}, A^{(ij)}, T^{(ij)}(s, a, s'), R^{(ij)}(s, a) > \), where in addition to the previously dis-
cussed parameters, the \( R^{(ij)} \) is defined as the reward for the action \( a_k^{(ij)} \) given that
the state is \( s \). The reward concept is the utility definition that the agent may except.
In an MDP formulation, although there is a great uncertainty for the transitions of
the states based on the executed action, the control agents are provided with a com-
plete information on the current world states. In this study, however, due to the
trustworthiness of the taken observations, additional complexity arises in determin-
ing the real world state. Rather than having a complete world state information,
there is the belief state concept, where the agent tracks a set of believes set \( Z \) by
means of the additionally taken observations. To this end, the observation function
\( O(s', a, o) \) is defined, which is the likelihood of making the observation \( o \), given that
the next state is \( s' \) and the action \( a \) is executed. In our model, the \( O(s^{(ij)}_l, a^{(ij)}_k, o^{(ij)}_m) \)
takes either \( \frac{1}{M} \sum_{m=1}^{M} tr^{(ij)}_m \) or \( 1 - \frac{1}{M} \sum_{m=1}^{M} tr^{(ij)}_m \) depending on the expectation of
either critical instance or a non-critical instance. The POMDP model includes the
state-estimator as an additional dynamic in order to provide a belief for the current
state of the agent world. For a specific objective, the next state estimation \( b_l^{(ij)} \) at
time step \( l \) is a function of the current belief state \( b_l^{(ij)} \), the taken action \( a_l^{(ij)} \) and
the observation \( o_l^{(ij)} \).

\[
\begin{align*}
b_{l+1}^{(ij)}(s_{l+1}^{(ij)}) &= P\{ s_{l+1}^{(ij)} | a_{l}^{(ij)}, a_{l}^{(ij)}, b_{l}^{(ij)}(s_{l}^{(ij)}) \} \\
&\sim O\{ s_{l+1}^{(ij)} | a_{l}^{(ij)}, o_{l}^{(ij)} \} \sum_{s_{l}^{(ij)} \in S^{(ij)}} T^{(ij)}(s_{l}^{(ij)}, a_{l}^{(ij)}, s_{l+1}^{(ij)}) b_{l}^{(ij)}(s_{l}^{(ij)})
\end{align*}
\]

Now considering an l-step non stationary policy \( A \) and a \( k \)-step move of the agent,
we are interested in calculating the expected total sum of the rewards gained as the
value function. As the first reward is simple the reward associated to the known state:
\[ V_k(s) = R(s, A(s)) + \gamma \sum_{s' \in S} T(s, a_k, s') \sum_{o_k \in O} O(s', A(s'), o_k) V_{o_k}(s') \]  \hspace{1cm} (4.16)

Remembering that the state information is not provided for the agent, the expected optimal policy is calculated over the belief-states as the belief state value functions are given as \( V_k(s) = \sum_{s \in S} b(s)V_k(s) \). Kaelbling et. al [3] proposes a more convenient method in representing the value functions, where the \( \alpha_k = < V_k(s_1), ..., V_k(s_n) > \) vector represents the value function distribution among possible states and \( V_{k+1}(b) = \arg\max_{k \in P} b, \alpha_k \).

The iterative calculation of the complex models may require a long computation time and hence computation power. In this study, we re-estimate the POMDP based optimal policy for the previously discussed control loop in case certain model parameters, i.e. the state transition parameters, differs over a previously determined threshold \( \epsilon_t \).

### 4.1.3.4 Framework Restrictions and Requirements

A certain amount of framework parameters are required for the control operation. These parameters are:

- \( \lambda \), which is discussed in Section 4.1.3.2.1
- a network ontology model for the graded network objectives
- information models for the action and observations including definitions for corresponding network objectives
- the \( \epsilon_t \) as a threshold for the sensibility of framework to the changing control loop parameters to construct new POMDP based control policies
- \( tr_{ij}^{(ij)} \), the trustworthiness for each observation, discussed in Section 4.1.3.3.1

### 4.1.4 Software Architecture

This section is devoted to the details of the software implementation. The proposed control framework is implemented using Java programming language, and we make widely use of dynamic class loading in Java programming [164].

138
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

4.1.4.1 Block Diagram

The block diagram of a cognitive network entity is provided in Figure 4.7. As we selected an access point (AP) in our technical network scenarios, additional AP related blocks are added into the illustration.

As can be inferred from Figure 4.7, the proposed control framework is implemented as the network controller unit having a set of observations, actions and the proposed network objective prioritization unit as the decision making engine (DME).

Additionally, we implemented a dynamic action / observation loader mechanism from a repository in order to provide the cognitive entities with the ability of gaining more efficient strategies for each objective defined in the network ontology. The network administration can dynamically change, update or upgrade observation / action sets for the corresponding cognitive entities. The similar mechanism is implemented also for the POMDP based control loop policies. The cognitive entity polls periodically the related repositories for the updates on the network policy, the POMDP modified control loop and the action/observation repositories. This is to provide the researchers using proposed autonomous control framework with the ability of dynamically changing the framework related parameters, which are described in Section 4.1.3.4.
The observation sets are in conjunction with the measurement / reasoning plane in order to ease the implementation. Additionally various observations can be added to the system in order to populate various observations at different points in the network, e.g. from the backbone network or the last hop. Finally, an additional client collaboration unit is added for the technical scenarios, which are explained later in Section 4.1.5.

4.1.4.2 Software Flow Charts

The main DME software consists of two main blocks. The first block is responsible for the organizational issues and the second block is responsible for dynamic resource allocations. We illustrate the flow chart of first block in Figure 4.8.

The initialization process starts with loading and analyzing the network ontology based on the network entity type. Hierarchical objective arrays are generated during analysis phase and corresponding administration based object-priorities are assigned. A similar action is done for the information model of POMDP based control loop policy. The initialization process ends up after creating DME threads, namely, dynamic action loader, dynamic observation loader and the main resource allocation / control loop executor thread. The second organizational thread type, namely, the dynamic action or observation loader thread, is responsible for updating hierarchical objective arrays by attaching new observations or action objects in case of an update.
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

In this model, we make widely use of information models (IM) for the boot up processes and also dynamically handling newly added action or observation sets or changes in the framework parameters. We illustrate in Figure 4.9 examples for the information model of a specific observation and action.

Figure 4.9: Example Information Models (IM) for an Observation and Action

The second type of threads is assigned as the main DME thread and is responsible for prioritizing the network objectives and executing the corresponding POMDP based control loops. The flow chart of this thread is given in Figure 4.10. A probability distribution function (pdf) is created based on the priorities of second level objectives and a random second level objective is chosen according to the created pdf. A similar randomization approach is utilized for the observation set of the corresponding second level object and the POMDP based control loop is executed in case the chosen observation is critical. Finally depending on the taken observation,
the priorities for corresponding second level objective and the observation is updated using the proposed network objective prioritization algorithm.

4.1.5 Experiments and Test Results

In this section we describe the evaluation methodology for the proposed control framework. The behavior and decisions of the control framework is tracked through simulation and real experimentations. Following subsections give the details of experiment environments, test scenarios and the corresponding results.

4.1.5.1 Simulation Results

4.1.5.1.1 Scenario Description  To illustrate and evaluate the performance of the proposed control framework, this section presents a sample simulation test scenario. A cognitive network entity has the sample ontology as the network policy, which is represented in Figure 4.2. Referring to this Figure, the action and the observation sets for both the Congestion Avoidance and Reduction of Quality (RoQ) mitigation objectives consist of three functionalities. We prepare a Java based simulation software in order to realize this network scenario. This software basically resembles the network environment, where critical instances are generated based on the provided probability distribution models. The generated critical conditions may be handled if an additional probability distribution model is provided to the software. These conditions are maintained if no input is provided. Such situations are assumed to
be handled once the generated random numbers, i.e one as the critical case and one as a corresponding solution, are compatible with each other. Our main objective is to make use of this property in testing i) the prioritization of network objectives, which are resembled by the probability distributions with a higher expected occurrence rate and ii) the elimination of the corresponding actions, which are resembled with the probability distributions generating compatible random numbers, i.e as impact.

We name the actions in the action set of Congestion Avoidance as Action 1, Action 2 and Action 3. Similarly the observations for the corresponding objective are named as Event 1, Event 2 and Event 3. We model the occurrence of events and successful impacts of the actions with Poisson distribution. In the beginning of the simulation the occurrence rates of the events have the following relation:

\[ \nu_{\text{event}3} > \nu_{\text{event}2} > \nu_{\text{event}1} > \nu_{\text{event}5} > \nu_{\text{event}6} \]  \hspace{1cm} (4.17)

In other words, we perform the simulation in a mostly congested network environment. We change the rates during the simulation from time to time. The relation between successful impact rates of the corresponding actions are:

\[ \varphi_{\text{Action}3} > \varphi_{\text{Action}2} > \varphi_{\text{Action}1} \]  \hspace{1cm} (4.18)

and

\[ \varphi_{\text{Action}6} > \varphi_{\text{Action}5} > \varphi_{\text{Action}4} \]  \hspace{1cm} (4.19)

We run a three hours simulation and observe the behavior and the decisions of the control framework under these conditions.

4.1.5.1.2 Results and Evaluation  
As can be inferred from Figure 4.11 the instant probability distributions of the belief states for specific objectives are dependent on the observation rates of special events in the corresponding observation set. Hence, the priorities of network objectives are changed during the simulation and the network resources are allocated accordingly and the network response time is decreased for the prioritized objectives. Moreover, the observation of events with higher rates of occurrences are prioritized under the same objective observation set. The proposed control framework takes 946 times a measurement in order to track Event 3, 324 times for Event 2 and 275 times for Event 1. Additionally, the network adapts to the changing conditions in terms of resource allocation. It is observed from Figure 4.11 that the priority of Security objective is increased once the rates of occurrences for security related events are increased.
In case of an increase in the success rate of an action, the proposed control framework has a tendency for performing the same action for the critical situations, which are defined under same objective observation set. As illustrated in Figure 4.12, Action 3 has the most effective impact on the environment in case of a congestion and is taken 99 times, while the opponent algorithms Action 2 is taken 50 times and Action 1 is taken only 19 times.
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

4.1.5.2 Test Bed Experiments

4.1.5.2.1 Scenario Description Two wireless APs have the aforementioned sample ontology, which is represented in Figure 4.2. Based on different combinations of observation sets and the different environmental conditions, we define two test scenarios. For the first scenario, the action set for the Congestion Avoidance objective consists of three functionalities, namely, increasing the transmission power, decreasing the transmission power and finally migrating the client with worst performance feedback to the second AP. For this last action, we make use of a similar methodology with the one in Section 3.3 in order to off-load a specific client to another available AP. We define the instantaneous bandwidth utilization of AP as the single observation for this objective. Decreasing the transmission power is the single action for the RoQ mitigation objective. Similarly the single observation of this objective is threshold delay time for the network performance observation from each client perspective. With this configuration we perform the experiment where the network environment is highly congested from time to time.

For the second scenario, we keep the action and observation sets for the Congestion Avoidance same as it is in the first scenario. In the beginning of experiment, no functionality exists in the action repository for the RoQ mitigation objective. We add the instantaneous bandwidth utilization of AP as the second observation for this objective. In other words, both objectives have the common observation set. The network environment includes malicious clients and is rarely congested during the experiment. We add later an attack mitigation action, which is a similar methodology proposed in Section 3.3, to the repository during the experiment.

4.1.5.2.2 Results and Evaluation For the scenario one, we illustrate the behavior of control framework on one of the APs and the network performance tests of three clients in terms of average delay time in Figure 4.14 and Figure 4.15. Based on the observation arrivals, the AP successfully organized objective priorities and the network response time in critical situations is improved. With an additional mechanism, the AP detects the clients with higher bandwidth utilization and balances the load in a fair way. Moreover, it is clearly observed that the AP is capable of choosing better actions against congestion.

The Figures 4.16 and 4.17 illustrate the behavior of the AP in the aforementioned scenario two. The control framework successfully detects the new attack mitigation algorithm in the attack/observation repository and takes the action against RoQ attacks. We observe that the performance tests of the clients are improved even in intensive attack instances.
4 An Autonomous Framework for Load Balancing in UCN

4.1.6 Evaluation of the Proposed Approach

In this part, the action and observation repositories are limited to the executables, which are restricted to a finite run-time period and are composed of a sequential order of function calls. In other words, the repository elements that are dynamically loaded and executed, are guaranteed to be terminated in a short time period. This aspect may be strengthened by splitting algorithms of loop characteristics into various function sets, which may in turn be compiled as separate executables and added to the corresponding repositories. Nevertheless, we define this limitation as vulnerable sector, which we will address in the next section of this chapter.

Another attention catching dimension in this study is the natural question, “how do we define the false triggers?” In this work, we limited the definition of false triggers to a single perspective, i.e., a case of critical observation might not define an alarm condition in the network or indicate a less successful impact on the network environment for an executed action. This factor is represented with the design parameter \( tr \), i.e. a grade representing the trustworthiness of the observation. However, an additional perspective on the false triggers, i.e., the missing data, where no trigger interrupt the control framework despite of an anomaly situation in the network is left as a future work for this study.

When it comes to shedding the light on the designer inference factor, i.e., the
4.1 Autonomous Resource Allocation and Strategy Optimization in Wireless Networks

Figure 4.14: Object Prioritization based on Event Arrivals and Client Experience

Figure 4.15: Success Rates and Instances of QoS Objective Actions vs. System Time

\(\lambda\) parameters in \(D_0\) Matrix (ref. to Section 4.1.3), which are defined as constants and are shown to be the normalizing factors in the control framework’s believes and effecting the priorities of network objectives. \(\lambda\) values are set by the network admin-
An Autonomous Framework for Load Balancing in UCN

Figure 4.16: Object Prioritization based on Event Arrivals and Client Experience

Figure 4.17: Success Rates and Instances of Network Actions vs. System Time

istration at the boot up time. With this single tuning parameter, the impact of the network administration can be adjusted in the decision instances of the proposed control framework. This parameter provides flexibility for the designers to interfere
on the operation of the network as a human to machine mediator. On the other hand, for very dynamic environments boot up assignments of these parameters may not be appropriate and sufficient as presumed by the designer. We believe that the optimization on the control of $\lambda$ parameters may potentially be helpful for the stability and reliability of the proposed control framework. A similar discussion is done on the inclusion of certainty to the decision-making in Chapter 2, where the certainty level of the environment behavior, i.e. the measure for the environment routines, has a great impact in the decision. Though one may argue that the design complexity would increase depending on the administration preferences as the proposed framework targets a more comprehensive perspective and excluding the administration preferences may not be suitable for various network scenarios and architectures. Hence we keep the mediator parameter in the proposed framework without including further machine interference on the administration preferences.

In the next Section, we apply this framework for the load balancing problem in UCN.

4.2 Application of ARASO Framework for the Load Balancing Problem in UCN

The highly dynamic characteristics of the UCN environment raise the complexity in the decision of solution approaches for the load balancing. The very brief summary for the determinants, which characterize the UCN system settings in terms of AP load can be listed as:

- Stochastic coalescences of UCN communities in different day-time and the public area, e.g. the load in an Airport may arise from excessive number of client arrivals and departures, whereas in a public social areas the clients may comparably prefer longer session time.

- Different Wi-Fi configuration capabilities among the UCN-APs, e.g., different wireless modules may or may not support various Wi-Fi related configurations, which limits action capabilities for load balancing instances.

- The mobility behavior of UCN stakeholders, i.e., a stakeholder may join to the UCN community with the same Wi-Fi entity as an AP in different UCN environments.

- Different traffic types depending on the immensely alternating client / community interests or the physical wireless medium characteristics, e.g., the UCN-
AP may be operating under heavy interference conditions.

- The fully stochastic deployment/positioning of the UCN-APs in terms of geographical distribution and the number, i.e. a certain number of critically important factors for the signal interference and the load distribution is not preplanned and hence may not be forecasted.

Considering the variety of the environmental conditions, which are listed above, the impacts of different load balancing oriented actions or algorithms may vary depending on the characteristics of operation environment. For instance, adjusting transmission power, a very broadly investigated method and also known as cell breathing technique for the operator networks, may be effective for the environments, where the network is exposure to higher client arrivals and departures. Nevertheless this superiority may hold only over the network environments, where the network clients are quasi static, i.e. they are either static or mobile in the coverage area of a collaborative AP pool. In case of quasi-static clients’ presence, balancing the network load by means of client traffic migrations among UCN-APs is clearly more convenient in comparison to cell breathing technique. On the other hand, the contrary situation, i.e. higher client arrivals or departures, necessitates the utilization of cell breathing techniques or admission control (AC) based approaches. Thus the load balancing strategies of a UCN-AP should be adapted to the changing environmental conditions in an autonomous way.

In this part, we detail our autonomous framework for the load balancing problem in UCN. Our method is based on the characterization of the environment and generating corresponding probability distributions over the belief-states of the operation environment, which constitutes the first stage of the proposed framework. The second stage is to monitor the impact, i.e. the success ratio of the load balancing oriented actions, algorithms. Besides we discuss an improvement point to the previously discussed ARASO framework, i.e. how we introduce long run algorithms to the autonomous framework in addition to the actions with sequential functional characteristics.

It should be underlined that in this study, we focus on longer and static AP functionalities of the UCN members with a provider motivation. With the longer and static, we mean that the AP functionality is turned on when the UCN member is not mobile and the AP operation is long enough for the stabilization of the proposed solution. Nevertheless the position of the UCN stakeholder with a provider role may be varying once the AP functionality is not activated.
4.2 Application of ARASO Framework for the Load Balancing Problem in UCN

4.2.1 The Load balancing Policy

In this part, we realign the previously discussed autonomous framework dynamics focusing on a single problem, i.e. the load balancing problem, in a more comprehensive and detailed manner. The very generic functionalities and components of the proposed autonomous framework are replaced with the corresponding load balancing problem conjugates. We describe the load balancing policy of the UCN-AP in a graphical and ontological form. We replace the objectives of the network with the environmental setups. We model these different environmental states, namely, high interference, higher client arrival & departure rates and homogenous bandwidth utilization among clients. With the homogenous term, we mean the similarities among network clients in terms of the amount of uploaded and downloaded bytes per time without going into details, e.g. the network activities or the traffic categories. It should be highlighted that these environmental states are not necessarily complementary or independent. In other words, the combinations of these environmental conditions may arise.

As illustrated in Figure 4.18, the "Load Balancing" concept is analyzed in three different environmental states. In this study, we focus specifically on these three different states, nevertheless the description of environment may be diversified with additional observation and action sets. The observation sets provide information on how likely the environment may be described, which forms the aforementioned first stage. Finally, the action & algorithm sets form the action pool as the steps which are likely to be taken against unbalanced load, which are categorized in the second stage based on the action / algorithm cost and success rate. More details on the observation and action & algorithm sets are given in the proceeding sections.

4.2.2 The Observation Set

In this study, the observation sets are utilized for the classification of environmental characteristics, based on which the AP strategies are adapted. The observation parameters are kept relatively simple, which may be replaced with better ones based on extensive researches, as the contribution of this section lies mainly on a skeleton with generic framework orientation. When it comes to shedding some light on the critical parameters as triggers for load balancing; each algorithm & action includes also their specific critical parameter descriptions. In other words, each execution of a specific algorithm or an action for the load balancing requires a test on the specific critical parameters, which are individual triggers specific to the executed algorithm or the action. The main objective of this study is to provide the UCN-AP with an
autonomous framework mainly for the strategy optimization and resource allocations depending on the environmental conditions. Thus the observation sets included in the load balancing policy forms the critical parameters that are utilized in describing the environmental characteristics. In this study, i) the associations & de-association rates of the clients are used in describing higher client arrival & departure rates, ii) the experienced signal-to-interference ratio (SIR), collected from the network clients are used in describing the high interference and finally the BW utilization of UCN-clients are monitored for the description of homogenous bandwidth utilization among clients.

4.2.3 The Action & Algorithm Set

One of the very first main motivations of the UCN µ-Providers is to guarantee an acceptable threshold of the QoS for the UCN maintenance. Although the load balancing philosophy in UCN necessitates a fair resource sharing among UCN-clients, this may not be applicable as the distribution and the amount of UCN resources are fully stochastic due to the stakeholders’ preferences. In cases of insufficient resources, the system is relaxed with the assumption of potential client off-loads to different telecommunication landscapes or access technologies as previously dis-
4.2 Application of ARASO Framework for the Load Balancing Problem in UCN

discussed in Chapter 3. Hence the proposed solutions for different dynamics of load balancing problem in UCN focus on an acceptable QoS degree for the UCN-clients throughout this thesis.

In this framework, we make use of three different approaches as action & algorithm against load balancing problem in UCN. The first approach is a cell-breathing like transmission power control technique, where the transmission power of the AP is decreased for a predefined amount of time and set back to its initial value. One may argue for a more enhanced transmission power control approach by means of the introduction of two additional parameters to the control algorithm, namely, i) the degree to what extend the transmission power is lowered, and ii) the duration of network operation with a specific transmission power value of the AP. Nevertheless we keep these parameters relatively simple and static as the main focus of this chapter lies on the proposed autonomous framework for load balancing.

The second approach is an admission control technique based on the UCN dynamics, which are discussed in Chapter 3. The load is balanced in a collaborative way among UCN-APs by means of the admission control for the newly arriving clients instead of dynamic migrations of already associated clients. The proposed algorithm is distributed and hence scalable. We keep the same POMDP controllers for requester and the requestee, which determines the utility components of the UCN-APs. Nevertheless, we slightly modify the collaboration request parameters and replace the number of migrated clients and incentive with a single admission control collaboration boolean (ref. to Section 3.2.4). Finally, the third approach is the cooperation incentive based load balancing algorithm, which is detailed in Chapter 3.

4.2.4 The Action & Algorithm Execution Control

In this model the algorithms & actions are triggered in case of a critical bandwidth utilization observation and the POMDP modified control loop is executed accordingly (ref. to Section 4.1.3.3.1). As an improvement for the previously detailed ARASO model, we include the long-term algorithms into network action repository by introducing a new parameter for the repetitive activities (loops) of the algorithms. The algorithms are slept for a dynamically alternating duration, which is determined by the positive and the negative evaluation steps of the proposed POMDP based control loop. In this wise, the algorithms & action coalescences are formed with various dominance levels and percentages based on the changing environmental conditions.
4 An Autonomous Framework for Load Balancing in UCN

4.2.5 The Algorithm Evaluation

It should be highlighted that the algorithms are evaluated based on a load balancing perspective and the efficiency in terms of the algorithm impact on keeping the network performance related measurements over an acceptable threshold. One may argue that the UCN dynamics, such as reputation / trust parameters, monetization or the virtual currency are the effective parameters in describing an optimality condition for each network control related activity. Nevertheless, a generic formulation for the real time calculation of the utilities may not be applicable for the UCN due to potential conflicts with the proposed algorithms. Moreover, the targeted stabilization time period may not be easily synchronized with a potential generic formulation. Hence we keep the the impact of UCN dynamics related parameters in the proposed network activities against load balancing problem.

4.2.6 Evaluation of the Framework

In order to evaluate the proposed framework for the load balancing problem in UCN, we make use of the setup detailed in Section 4.1.5.2. For the ARASO software, we provide the corresponding new policy ontology, the observation sets, the action sets, and finally the aforementioned slightly modified POMDP based control loop. With the new settings, we emulate different environmental conditions and monitor the beliefs, the changing strategies and the decision instances of the ARASO software.

We patch the Hosatpd [165] in compliance with the admission control based load balancing algorithm and also with the client departures / arrivals observations. For the sake of generating different environmental conditions, we emulate from time to time the client arrivals and departures on different clients by means of additional software, which simple enables and disables the wireless connectivity from time to time. We also connect long-term bandwidth hungry clients in order to emulate the heterogeneity as we believe that the homogenous bandwidth utilization among clients may not hold in a realistic network scenario. Finally we generated dummy and heavy traffic on the APs from time to time with the additional network traffic generators in order to create interference on the clients. With this motivation we also make use of traffic shapers on the APs by means of dynamically changing the available bandwidth from time to time. As the impact of each action item depends on the environmental conditions and hence the rate for which the network stabilizes itself, we kept the emulated environmental conditions are for longer time periods in order to achieve reliable outcomes.

As illustrated in Figure 4.19, the beliefs on, and hence the priorities of, the envi-
4.2 Application of ARASO Framework for the Load Balancing Problem in UCN

Figure 4.19: Experiment Results: Impact Rates of Various Algorithms in a very dynamic UCN Environment

Environmental conditions are changed from time to time. We start with an environment, where the bandwidth utilization among UCN-clients are heterogenous and continue with this setting till the end of the experiment. For the start-up settings, the arrival and departure rates of the UCN-clients is not critical and we emulate low interference among the UCN environment, where the clients are not far away from the APs and hence the RSSI values are well enough. We start congesting the one of the APs with a long-term UCN-client and observe the network actions accordingly. As can be inferred from the Figure 4.19, the proposed cooperation incentives based load
balancing algorithm dominates, receives a high success rate and hence is selected frequently as the strategy by the ARASO software. The POMDP loop period of this algorithm is decreased accordingly. We conclude that the given environmental conditions is cut for the cooperation incentive based load balancing algorithm. For the proceeding phases of the experiment, the environmental conditions are changed stepwise and the arrival / departure rates of the UCN-clients are increased over a critical threshold. The ARASO framework catches the changing environmental conditions and increases the priority of the corresponding environment label as shown with the blue lines. Clearly, the strategies of the UCN-AP are adapted accordingly; the admission control and transmission power control based algorithms are taken more in comparison to the previous phase, while the success rate of the cooperation incentives based load balancing algorithm has less impact rate. This is due to the fact that the second UCN-AP is congested from time to time and becomes non collaborative.

Keeping the second phase environmental conditions for a longer term, we observe that the impact rates of the proposed algorithms get less. Nevertheless, this stems from the characteristics of the environment due to high arrival and departure rates, as the system is congested from time to time and in a very dynamic way, which was due to lack of enough resources, i.e., available UCN-APs. Although it is observed that the proposed framework reacts rapidly to the changing conditions and forces an adaptation to the environment, the availability of resources among UCN environment becomes significantly important. At the final stage of the experiment, where the interference is added to the present settings, this fact plays a great role as various strategies become less effective in extreme environmental conditions. For the higher interference environments, due to additional packet losses and continuous retransmissions, the bandwidth utilization of the AP increased although the throughout and the QoS measurements get worse. It should be highlighted that this increase in the amount of additional bandwidth is slightly perceivable. Nevertheless the system is maintained in a way that the environmental settings include high interferences as we decrease the threshold values. Under these circumstances, the transmission power control based load balancing approach is appraised for low throughout the experiment and clearly results in a degradation of the QoS measurements. We believe that this property is very dependent on the geographical distributions of the clients as well.

Finally, we conclude the following take home notes for the load balancing problem in UCN:

- A cooperation incentives based approach, which is discussed in details in Chap-
4.3 Conclusion

In this chapter, we introduce an autonomous framework for the load balancing problem in UCN. The chapter starts with a more comprehensive perspective, however, is confined to a single problem, which is the focus of this thesis. In the next chapter, we comment on the implementation aspects and introduce a user centric wireless testbed, which is extensively utilized in validating proposed approaches in this thesis.
5 Test Environment and Implementation Aspects

This thesis focuses on an autonomous framework for the load balancing problem in UCN, where problem is handled with the extensive end user collaborations. To this end, the collaboration among various UCN entities takes great role for the proposed solutions. Additionally the thresholds of provided network performances and hence the resource capacity of a UCN-AP is taken into account for certain number of decisions instances. These settings can be generalized for most of the UCN dynamics, which are discussed in Section 1.1. In this chapter, we present the test environment, which is utilized throughout the thesis in validating proposed approaches. We, additionally, comment on the implementation aspects for the load balancing addressing above-mentioned settings, i.e. the natural questions for what are the impacts of i) deploying a collaboration unit for various UCN related activities on the end-user device, and ii) extensive resource utilization by others from the end-user perspective.

The chapter consists of two main sections. In the first section (Section 5.1), we introduce the user centric wireless test-bed, which interprets the user as a key component of the network control and operation. The proposed testbed offers programmable entities in both core and access network edges, enabling us to implement the previously discussed models. We aim at providing a flexible validation environment for UCN oriented researches with our testbed. In the second section (Section 5.2), we put emphasis on the feasibility and share our concerns on the realistic implementation for the end-user collaborations and resource allocations. Having this motivation, we propose a novel approach for the compatibility of APs to the UCN environment and give details for how we equip UCN-APs in the user centric wireless testbed.

5.1 User Centric Wireless Testbed

Consumers in today’s telecommunication networks are faced with an end-to-end value proposal, where the network path traverses multiple organizational and tech-
5 Test Environment and Implementation Aspects

Wireless access technologies in many different forms started to take an increasing and critical share in this end-to-end path. Therefore it is essential to study the effects of wireless technologies, in conjunction with core networks, from an end-to-end service quality perspective.

Simulation studies have strong dominance as the means for network protocol analysis, particularly in wireless network research. Unfortunately, simulation tools and models largely depend on simplifying assumptions that significantly limit the accuracy of such studies in real-life scenarios, intensified by the physical-layer aspects of wireless communications [166]. Therefore there is an increasing need in the research community for performing controlled real world experiments on dynamically programmable testbeds [138].

For the next generation networks, new networking approaches are suggested where the network is defined as a collection of resources that need to be allocated consciously, for which the intelligence level of network should be increased. This intelligence can be realized by decision entities embodied in the network fabric that collect data from the network and store in semantic repositories [167]. Our aim is to realize an experimental research environment for developing and testing such semantic and programmable network approaches that covers both wired and wireless domains for the UCN oriented research.

Our testbed aims to offer a set of powerful tools for researchers working in UCN domain: (i) programmable entities in both core and access network edges that enable the implementation of cooperative decision mechanisms for enhancing the UCN experience, (ii) resource configuration and traffic generation tools for easy creation of realistic test scenarios.

In this section, we initially provide an overview of the wireless testbeds literature in Section 5.1.1. We evaluate the related work and state our approach in Section 5.1.2. Then after, we introduce the testbed architecture in Section 5.1.3, whereas technical details of the testbed components is presented in Section 5.1.4. Finally we give the details of a sample experiment in Section 5.1.5.

5.1.1 Literature Review on Wireless Testbeds

There exist several substantive definitions of testbed in the literature. Erek et al. define testbed as the “perfectly normal instance of the system that is under study in a particular experiment, which is used for meeting various experimental objectives such as collecting data to be interpreted for obtaining indicative results for the system under test (SUT)” [168]. Considering wireless technologies and protocols as the SUT, one encounters numerous studies in recent years.
The Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) [138] is an indoor radio grid emulator over 400 nodes designed for controlled experimentation, which also gives the opportunity for the researcher to receive feedback from end-user evaluations. This massive indoor testbed is a scalable system in terms of the total number of wireless nodes. The testbed is open-access and flexible in terms of high-level control given to the experimenter and supports reproducible experiments. The ORBIT control framework additionally supports remote access, where extensive measurements are allowed to be done. ORBIT provides the researchers with the ability to perform tests on the Medium Access Control (MAC) layer experiments by giving them full node access.

The EMULAB [169] is designed for the emulation of not only arbitrary wired network topologies but also wireless sensor networks. This testbed provides a real mobile wireless sensor testbed by which users can remotely control the robots carrying sensor motes. The main purpose of this testbed is to provide researchers with the ability to evaluate Wireless Sensor Networks (WSN) applications under mobility with real wireless local area network (WLAN).

Many aspects of centralized WLAN systems are claimed to be poorly understood in the sense of wired delay and jitter properties, therefore Ahmed et al. describe a large-scale WLAN testbed for centralized control, in order to issue centralized control algorithms [170]. This testbed is mainly intended for the researchers who are interested in experimenting centralized control for traffic scheduling and data rate adaptation. Doing experiments on the proposed testbed, Nabeel et al. try to confirm the requirements for a centralized control based on the assertion that central control is necessary to support network optimizations such as centralized packet scheduling [171].

WART (University of Colorado Wide-Area Radio Testbed) is a well known example of outdoor WLAN testbeds, which is designed as a facility for studying smart antenna [172]. The testbed consists of eight phased array antenna nodes that are mounted to the rooftops of the university and is dedicated for studying the impact of omni-directionality, directionality, null-steering and beam-forming throughout the network stack. In Comparison to WART, Roofnet [173] is a primitive example deployed on Cambridge, which also provides Internet access as a multi-hop mesh network. Roofnet is not a dedicated testbed, which, in turn, limits the ability of researchers working on this testbed. In addition to these two famous testbeds, RuralNet [174] can also be given as another example of outdoor wireless testbed, which is designed for experimenting on very long range point to point communication.

Some of the WLAN testbeds are offered for special purposes. Caltech multi-vehicle
wireless testbed [175] is a good example as a platform for testing decentralized control methodologies for multiple vehicle coordination and formation stabilizations. Moreover, some testbeds proposed in the literature aim at receiving direct feedback from actual users. For example, Gonguet et al. suggest an IMS (IP Multimedia Subsystem) experimentation testbed experimenting innovative services that will be designed and developed by a composition mechanism with actual users [176]. In another study, Reality Mining, a project at the Massachusetts Institute of Technology (MIT) Media lab, researchers collected data from 100 Nokia Symbian series mobile phones over a period of 9 months in order to understand the social networks (i.e. the human social behavior). It can be inferred from these studies that the new trend in network research is to work with end users and improve mainly Quality of Service (QoS) of the network in cooperation with network users.

5.1.2 Evaluation of the Literature from the UCN Perspective and Our Approach

Several wireless network testbeds provide frameworks for experimenting with specific network technologies and network entities with hardware/software limitations. However, we believe that the end user should cooperate with the core decision elements inside the network for the realization of reliable end-to-end service quality. It should be underlined that for the UCN, this claim becomes stronger as the UCN is an autonomic formation of the network by the end users. Considering the UCN dynamics, one may not easily adapt the experimentation phases for various UCN oriented proposals to the existing solutions. This is due to the fact that the user preferences play a great role. Thus there is a great need for the construction of a valid user centric testbed, where the experiences of UCN stakeholders based on their preferences can be extensively investigated.

In this study, we focus on a wireless testbed, where the wifi enabled entities play either the role of the access point or the consumer. We propose a joint wired and wireless research testbed where end users are the core element of the network, and the network entities provide interfaces for programming and storage capabilities. Our intention is to create an initial driving force for a validated UCN testbed having above-mentioned objectives. We additionally emphasize on an environment platform for utilizing distributed artificial intelligence (DAI) techniques in the Future Internet research, particularly with a focus on user centricity.
5.1 User Centric Wireless Testbed

5.1.3 Testbed Architecture

For the proposed testbed roadmap, we have the main motivation of allowing the researchers to test innovations on wired and wireless nodes jointly over a variety of realistic topologies. Our guiding assumption is that the end-to-end principle will be gradually replaced by more intelligent nodes on the service delivery path. Considering the UCN, where the participation of various stakeholders is realized by means of smart phones, tablet computers, PCs etc., we believe that there is enough motivation for replacing various network elements with high capacity programmable entities. This is reflected in our choice of using configurable nodes on each level of the network hierarchy. The testbed architecture is illustrated in Figure 5.1.

On the starting level, we have various mobile end user devices, e.g. Google Android, on which we run the client network manager software, which we developed for collecting various network performance related measurements and reporting them to the centralized repository. On the wireless access level, we currently employ 802.11 access points that are based on Voyage Linux [51] and OpenWrt [177] operating systems. The Linux-based operating system (OS) allows researchers to deploy innovative control algorithms on the access point. Secondly, we use OpenFlow [178] firmware to allow researchers to test innovations on Layer 2. Different WLAN access points are connected to a set of Linux based routers running OpenFlow controllers. The connections between the wireless section and the routers are made via a reconfigurable switches providing the capability of dynamically changing the network topology.

As shown in Figure 5.1, researchers can program network entities in both core and access network edges that enable the implementation of cooperative decision mechanisms for enhancing the end-to-end service experience. One additional opportunity of this testbed is to provide researchers with a basic implementation of the knowledge plane as a distributed peer-to-peer repository on individual network elements. Researchers can monitor distributed knowledge base and fetch semantic data, depicting the network performance related measurements. Similarly, distributed intelligent agents running on different network entities may revise and improve network service quality based on the feedbacks provided by the end users. The proposed testbed provides a flexible and easily configurable hardware platform together with configuration tools. In the next section, we give the technical details on the testbed components.
5 Test Environment and Implementation Aspects

5.1.4 Testbed Components

5.1.4.1 Hardware Components

5.1.4.1.1 Wireless Access Points  Alix boards from PC Engines [52] are configured as the wireless access points for the proposed testbed. This board, the hardware of which is illustrated in Figure 5.2, is equipped with 500 MHz AMD Geode LX800 CPU, 256 MB SDRAM and 1 CompactFlash(TM)-Slot for the operating system.
5.1 User Centric Wireless Testbed

installation. The board has 2 Fast Ethernet slots for the backbone connection and 2 mini PCI slots for wireless module expansions.

Figure 5.2: Alix Board

5.1.4.1.2 Wireless Module Compex (miniPCI) wireless modules are used for the wireless extension to the Alix boards. These modules support IEEE 802.11 a, b, g mod operations with 108 Mbps maximum transfer rate. The modules are configurable in 2.4 GHz - 5GHz band and are designed with Atheros chipsets.

5.1.4.1.3 Clients and Traffic Generators PCs are used as well-aimed or malicious clients and traffic generators in the proposed architecture. Traffic generators have 4 wireless LAN interfaces in order to throttle bandwidth when needed.

5.1.4.1.4 Control and Configuration Machine This component is also a Linux based PC connected to the backbone and also used as a traffic generator inside the system when needed.

5.1.4.1.5 Miscellaneous Tools NEO-Industrial PC IPC Embedded computer is used as openflow routers, user experience database and application servers. This computers consist of 1.6 GHz Intel Atom Processors and 2.5” HDD with 1 PCI Card for the operating system and additional software installation. Moreover, there exist 4 ethernet port and 2 PCMCIA sockets for networking purposes. Figure 5.3 shows the top view of this embedded computer.
5 Test Environment and Implementation Aspects

5.1.4.2 Software Components

5.1.4.2.1 AP Operating Systems  The operating system running on the wireless access points is Voyage linux [51], which is a Debian derived distribution and suitable for running full-feature firewall, wireless access points, Asterisk/VoIP gateway, music player or network storage devices. Although this distribution is a stripped-down version of Debian, it is possible to customize it and expand the capabilities with Debian packets.

5.1.4.2.2 Routing Table Configuration Program  This program is written with Qt cross platform [179] and gives the researcher the ability to select network elements from a given library and connect them through network interfaces. The output of this software is an XML file defining routing tables for each component that will be used during experiment. The control and configuration machine configures routing tables for each selected component accordingly.

5.1.4.2.3 Traffic Generators  Distributed internet traffic generator (DITG) [54] is extensively used as traffic generation tool. This software may generate network traffic at packet level accurately replicating appropriate stochastic processes for inter departure time and packet size. It is possible to define probability distribution functions for both of the given random variables as exponential, uniform, cauchy, normal and pareto. Packet randomization capability of DITG gives the opportunity for researchers to generate more realistic traffic load over APs during the experiment.

5.1.4.2.4 Protocol Experimenting  Although the protocol based approaches are out of interest in this thesis, we provide resources for lower layers related experiments for future purposes. Openflow [178] is installed on the routers in this testbed. This is to encourage new ideas proposed for UCN on protocol level. This software is basically an Ethernet switch of which flow-tables can be manipulated dynamically by adding or removing flow entries. A controller program (in our case NOX
5.1 User Centric Wireless Testbed

5.1.4.2.5 Configuration and Control Program  This software runs on the control and configuration machine and is the first interface on the network for the researchers. An experiment XML script defining various test related parameters is provided by the researcher. These parameters include mainly the timing related configurations, applications running on specific nodes, outcome monitoring software provided by the researcher and the corresponding network entity, NOX controller applications for protocol experimentations if applicable and etc... Similarly routing table xml should be created using routing table configuration program by the researcher and these two xml files together with corresponding experiment software is forwarded to this main control and configuration tool. This program interprets both xml files, configures all components inside the network and loads all softwares provided by the researcher to the corresponding entities. After finishing experiment, this programs collects experimental results defined on the experiment xml script and deletes all software and reboots any node inside the network for a new experiment. This program is in construction and will be expanded in functionality if needed.

5.1.5 Sample Experiment and Results

The testbed validation of the proposed approaches throughout this thesis are performed on user centric wireless testbed for which we provide experiment results in the preceding sections. An additional sample experiment is conducted in order to explore the capabilities of the proposed testbed. In this sample experiment, the researchers focus on a solution for the real-time inter-operator load balancing problem. In [181], Toker et al. propose a control algorithm for real-time inter-operator load balancing. This control algorithm runs on the access points belonging to different operators. The operators are in an agreement to carry each other’s traffic in the case one of them is congested and the other is under-loaded. The main obstacle in front of such a real-time sharing is the fact that operators are not willing to share their operating information such as number of users connected to the AP. A congested operator should make sure that the other operator which would receive additional traffic is under-loaded. Similarly, an under-loaded operator would not help another under-loaded operator. Both operators use the user QoE database to gauge the congestion status of their peer operators. Based on their observations, they take decisions to share or to stop sharing. To this end, we vary the traffic load on two access points by using a realistic stochastic traffic model, and measure jointly the
overall throughput and user perceived QoE. We quantify the QoE in terms of the probability that the end to end delay exceed a given threshold.

In Figure 5.4 we plot the ratio of sessions that have a delay larger than one second in the congested access point. It can be seen that the algorithm is able to reduce the ratio from 40% to 12%. Similarly the access point that accepts additional traffic from the congested access point is able to increase its average throughput from 6.6 Gbps to 7.4 Gbps over a period of an hour, as depicted in Figure 5.5.

Up to this point, we described basics and the skeleton of the proposed initial model for a potential and validated user centric wireless testbed. We improve our approach addressing aforementioned feasibility concerns, which pertain to realistic implementation aspects for end-user collaborations and resource allocations in UCN. In the next Section, we detail our approach.

### 5.2 Microkernel Based OS Virtualization as A Trusted Cooperation Platform for UCN-APs

The collaboration of end users on various parameters is not a cornerstone only specific for the UNC community. With a broader perspective, i.e. for the future
cognitive systems, the end user collaboration have been placed in the limelight of the large-scale research and development activities. In a simplest set-up scenario for the UCN coalescences, the end-user collaborations over network can be defined with the following key items:

- End-users can use resources of others entities for computation power
- An end-user can allow others to use his/her network resources for Internet access
- End-users can exchange critical information through network, leaving open ports / addresses to the others.

This set-up gains attention for not only UCN like unique telecommunications landscapes but also for cloud computing, connected technologies and also networked vehicles. We pictorially represent this concept in Figure 5.6, where the resources allocated for the entity owner and the resources allocated for the collaborations are resembled with two blocks on each entity.

The afore-mentioned scenarios may take place in many correlated or uncorrelated network architectures, where the end-user entities are equipped with additional softwares enabling the collaboration. The collaboration instances may, however, put a
risk on the experiences of the end users during the collaboration activity. This is particularly for the dynamic network formation by means of stochastic coalescences, where each stakeholder has the motivation of providing resources for others from time to time. On the other hand, the discontent, which arises due to frequent collaborations and hence excessive resource consumption, may raise concerns about UCN permanency. Last but not least; through the Internet, i.e. an information pool with particular routes for each resource, any end-user collaboration may have a privacy-related impact on others.

In order to cope with this tackle, a very straightforward approach may be to limit the resource utilization for the UCN environment in an efficient and sufficient way. This is to prevent i) an excessive resource utilization by collaboration units in order not to disturb the device owners tasks, ii) malicious operations from risking the end user privacy, i.e. replacing a privacy airbag between personal and collaboration related tasks. Nevertheless, considering the feasibility issues, one needs to handle a great complexity due to the implementation and realistic applications, which may address all these items. We believe that this complexity may be solved if the UCN and personalized activities of the stakeholders’ entity are separated by means of OS virtualization techniques. This is to prevent additional complexity due to feasibility related issues, instead of dealing with the evolutionary approaches on the de-facto telecommunication solutions. Having these motivations, we propose a resource allocation framework for collaborations and privacy oriented OS virtualization for the UCN-nodes. More in details, we propose an efficient OS virtualization, which guarantees related restrictions on the access of public interactions to the private personalized area.
5.2 Microkernel Based OS Virtualization as A Trusted Cooperation Platform for UCN-APs

In the following section, we provide the technical details of our approach.

5.2.1 Technical Details

The cores of the proposed model are i) u-Kernel (L4 [182]) based efficient OS virtualization, ii) the separation of the public and private resources of end user devices in a strict way and finally iii) the operation maintenance in case of critical OS collusions.

![Figure 5.7: A UCN Scenario](image)

As illustrated in Figure 5.7, the proposed OS model is equipped with private OS instance, which is based on either Android or OpenWrt, and the OpenWrt based UCN collaboration OS instance. We configure the L4 para-virtualized Linux kernels of the both OSs in a restrictive manner, where the private OS instance has the control over the entire hardware, whereas the collaboration OS unit has only network access through the open access unit. Additionally, we equip the platform with the specific services and drivers in order to give opportunity for further development and future works. We install our proposed autonomous load balancing solution on the UCN collaboration unit, with which the stakeholder entity participates to the UCN community. In this solution we allocate limited resources for the UCN collaboration OS instance in order not to disturb the personalized activities.

The proposed solution is implemented on the specialized UCN-APs in the user centric wireless testbed. These entities are equipped with Intel Atom processors.
5 Test Environment and Implementation Aspects

(D510) based PC motherboards [183], which is illustrated in Figure 5.8.

Figure 5.8: Access Point Entity with Virtualization Solution

In the next section, we comment on the network performance related experiment in order to explore the efficiency of our approach.

5.2.2 Exploring Network Performance of the Proposed Solution

In this part, we briefly give details on the evaluation methodology for the network performances of the proposed solution. We install same applications on a monolithic kernel based UCN-AP together with our solution and connect a certain amount of network clients to the APs. The client devices run the client network manager, details of which are provided in Section 3.2.5. With an additional Java based software we load the allocated CPU up to 95 % for the UCN collaboration instances from time to time. This is to test the main feature of micro-kernels, i.e. the device drivers are maintained based on the same privileges and the priorities with the user space applications [184]. Our main motivation is to test the effect of additional resource consumption by means of CPU hampering potential UCN solutions on the network performance. We collect the network performance related measurements from two specific clients connected to different APs.

As illustrated in Figure 5.9, we compare the outcome for both APs. We draw the average delay time, average jitter and average bit-rates with the blue line for the ordinary monolithic kernel based UCN-AP, whereas the red line is used for the micro-kernel based UCN-AP. We do not draw the average packet loss for both UCN-APs as there exists no packet loss during the experiment. In other words no critical observation is taken, which is related with the connectivity threatening performance degradation. Critical peaks are observed for the average delay time and the average jitter once the CPU is loaded up to 90% by means of aforementioned additional experiment software. This is an expected result due to the previously mentioned reasoning, i.e., as the corresponding application is run with the same privilege with the network device drivers. Additionally, It is observed that the average bit-rate is
highly variant for the micro-kernel based solution due to the aforementioned reason-ings. Nevertheless, the network performance is almost identical for both APs under ordinary conditions.

![Network Performance Comparison](image)

Figure 5.9: Network Performance Comparison of Trusted Cooperation Platform with a Full-featured AP

We believe that this property brings about a certain amount of advantageous in comparison to the utilities related with the resource allocation and privacy related issues. For instance, the reaction time of the network control or management related algorithms gets better as they also are maintained on the same level with the drivers. Moreover, we believe that researchers may also evaluate their approaches in
5 Test Environment and Implementation Aspects

terms of a certain amount of specific parameters, which may indicate performance characteristics, such as reaction time, speed and feasibility.

5.3 Conclusion

In this chapter, we provided details on the test environment and implementation aspects for the validation of previously discussed approaches. We present our user centric wireless testbed, which we believe have a potential to create a driving force for a valid UCN dynamics oriented testbed. We focus more specifically on the access points of the our testbed addressing a certain number of research allocation related aspects and propose our $\mu$-Kernel based virtualization approach. In the next chapter, we provide a brief summary, a general evaluation of the whole thesis, and finally state potential future works.
6 Conclusions

In this thesis, we focus on the load balancing problem in the very dynamic and stochastic UCN environment, where different approaches are studied addressing different load balancing aspects. The main objective of the proposed approaches are to introduce the autonomic behavior for load balancing in UCN. This is to keep the QoS indicators over acceptable thresholds among UCN-subnetworks. The solutions and approaches are relaxed mainly to two classes of decision-making sides, namely, the UCN-client and UCN-AP side. The UCN-client side decision is restricted to the access point selection stage then after which the UCN-client is loyal to the decisions driven by the UCN-AP. This stage is mainly proposed for the facilitation of the load balancing attempts of UCN-APs by means of end user collaborations. To this end, we come up with the answer for the question, how we can equip the UCN-clients with the ability of characterizing UCN subnetworks in terms of network performances and selecting more suitable ones. Additionally we extend our answer with the functionality of taking the short-term UCN-AP functionalities into account while selecting a specific subnetwork. For these abilities we also operate the UCN-clients with an active role by means of performing network performance related measurements and even forwarding them to the UCN-APs. The forwarded measurements are then utilized by the UCN-APs for the UCN-AP side decision-making. Finally on the UCN-client side, we discuss the mobility awareness while selecting the UCN-APs, where we define an imaginary and permanently available UCN-AP representing the user preferences in risking connectivity for the regions where the end user is mobile and transient.

On the UCN-AP side, we initially introduce our approach on how the load balancing can be performed among UCN-APs under the presence of selfish nodes operating with the $\mu$-Provider role. Our solution covers the distributiveness, fairness and selfishness of the stakeholders. To this end, we introduce our QoS-based crediting method as an incentive mechanism, which creates driving forces for the load balancing attempts. That is to say, the selfish APs are motivated for sharing their load with available APs based on the introduced principle that an AP may gain more credits while serving in an underloaded state, i.e. with better QoS. These de-
6 Conclusions

cisions are taken in a distributed manner by means of collaborations. The fairness is realized by the natural dynamic of the proposed QoS based crediting mechanism. This principle addresses the diversity of providers in terms of their hardware or software capabilities. In other words the load is balanced in a manner that the better equipped UCN-APs serve more clients than the ones serving with worse QoS. We additionally took the UCN-AP owners’ preferences into account while preventing excessive resource consumption from the associated UCN-clients and hence risking the owners’ network experience. Finally, we discuss an exceptional case, i.e. the environmental conditions, where the colocutor APs are not capable of balancing the load in a collaborative way and the decision should be taken in a central manner.

A higher level mechanism on the UCN-AP side is introduced for the autonomic behavior in load balancing. To this end, we initially focus on a more comprehensive model for the autonomous behavior in resource allocation and strategy healing in wireless networks. The first motivation of the proposed model is the elimination of redundant resource utilization for specific network objectives while reducing the network response time. The second motivation is to introduce the ability to the network for matching network strategies with different environmental conditions in terms of impact rates on the environment, i.e. the success rates of the taken strategies. Having validated this framework, we adjust the model parameters in order to focus on a single network objective, i.e. the load balancing. We define three different characteristics of environments, namely, homogenous BW utilization among UCN-clients, high arrival & departure rates and high interference. We additionally equip the UCN-APs with three different strategies for the load balancing. These strategies are the above-mentioned cooperative load balancing method, an admission control based method and finally cell-breathing like transmission control. At this stage, we envision the load balancing activities in terms of two iterative steps, i.e. the driving force for the decision making by means of a suitable strategy and the action to be taken. Hence we translate above-mentioned decision-making strategy (the one for cooperative load balancing) to admission control and pack it as another solution for the load balancing with a different action (admission control or refusal) into the algorithm repository of the UCN-AP. We validate our approach via real implementation and observe how the strategies of the UCN-AP change with the changing environmental conditions. The experiment results are interesting as they draw our attention on the significance of the network adaptation capability.

Finally we comment on our test environment and discuss the implementation aspects. We introduce the user centric wireless test-bed, which interprets the user as a key component of the network control and operation. It becomes more significant for
the UCN environment that the end user cooperates with the core decision elements inside the network for the realization of reliable end-to-end service quality. This is due to the fact that the UCN is a composition of end users having the provider and consumer roles, which is realized in an autonomic manner. Thus in our testbed we focus more on the end user availability as the access unit. This thesis studies different approaches for the load balancing problem based on frequent collaborations among stakeholders in the UCN. Thus we additionally state the risk for the UCN permanency due to the excessive resource consumptions by UCN-clients, which may arise due to frequent collaborations. In order to handle this problem, we introduce our methodology for limiting the resource utilization to a certain extent for the UCN environment in an efficient and sufficient way. We present our μ-Kernel based OS virtualization technique for the UCN-APs, which limits excessive resource utilization by others while decreasing the complexities in decision-making.

In this thesis, different methodologies are studied in order to realize above-mentioned dynamics in UCN. Learning and decision making under uncertainty are the basics of the proposed frameworks. One of the main methodologies, which is followed to form a common probability based decision-making, is the characterization of various phenomenon in the network. With the term characterization, we mean a learning approach based on combinations of certain artificial intelligence techniques for the partially observable domains with a motivation of constituting a mixture of various probability distributions. To this end we concentrate on the utilization of routine approaches proposed in the literature for the expectation maximization (EM) algorithm. We take the advantage of common EM utilization in defining different aspects throughout the thesis, where we assume a prior information on the probability distribution functions. The EM based characterization is forwarded as an additional information / belief to the decision-making stages. As a theoretical improvement, we focus on how one may neatly sidestep the obligatory pre-knowledge on the centered phenomena, i.e. defining a prior probability distribution model. The validation of the proposed approach encourages us for the future utilization of the proposed method as a facilitating technique for the solution of various tackles. Although we additionally implemented the engine with a Java based library, as a future work, we left the formal definition and the extension of our method in order to come up with an open source library.

With a similar approach, i.e. focusing on reusable methods, we define an autonomous framework for the resource allocation and strategy optimization for wireless networks. This mechanism is fully implemented using Java. We leave motivation of the centered framework to more comprehensive way, which can additionally be
6 Conclusions

put on the limelight for the single objectives. One another future work, that draws our attention, is to release the framework as an open source service application.

As the UCN is cut for a definition of partially observable domain, we focus on the utilization of partially observable Markov decision processes (POMDP) in decision-making. Different examples of modeling with POMDP are investigated as the solution of optimal policies in various components of the network. We slightly touch on how one may define dynamic POMDP solution by means of alternating state transition probabilities. Additionally, in the decision making phase we introduce the impact of the precision, i.e. the certainty level of the believes on the behavior of the monitored phenomena. When it comes to shedding some light on the drawbacks of the decision-making methodologies; the proposed techniques imitate control systems with negative feedbacks. Our main motivation is to combine different phenomena, which have great impact on the load balancing activities of the network, nevertheless may not be represented based on a common utility definition. Especially for the UCN, where various factors such as user preferences, monetization issues, crediting aspects, trust management etc are effective on the decision making, this tackle is highly tangible. An additional complexity causing factor is the reality that the previously ordered terms are generally based on subjective perspectives. In other words, a common understanding of various terms may not be applicable for the UCN environment, where the decisions are taken in a user centric way. This is the main ground for the proposed methodologies, where various subjective terms are kept over an acceptable threshold rather than struggling on a common utility definition.

Last but not the least, we believe that the proposed trusted OS virtualization technique may bring a generic-like solution for not only the UCN environment but also for various connected technology scenarios. Clearly, the immediate future is a candidate for the development history of the connected technologies such as networked cars, connected living scenarios, cloud computing or various network based social community applications. A more connected life is the target of the connected technologies, nevertheless, various security and legislation issues or the excessive consumption of resources by strangers are the certain open issues. Besides the privacy concern is one the main tackles for these technologies in case of malicious activities in the community. Although the proposed solution necessitates several difficulties and implementation complexity, when considering on feasibility, we believe that this method may help for various problems in connected technology scenarios. Our focus for the future contributions will additionally include the healing of the potential drawbacks of this system. The main roadmap includes coming up with an end-
user oriented entity definition for the future connected technologies based on our approach.

Each approach is verified by means of full implementation and long-run experiments with various network settings and scenarios. Additionally the proposed solutions are implemented with various software technologies in order to support different platforms. In this study, we mainly focus on a network platform, where the end-users and the entities at the access collaborate intensively for the network optimization in terms of the concentrated resource allocation aspect, i.e., the load balancing. We believe, however, that the proposed frameworks may easily be used for various network scenarios addressing different objective by slightly manipulating few model parameters.
Bibliography


Bibliography


Bibliography


Bibliography


184


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


