Universidade de Aveiro Departamento de
Eletrónica, Telecomunicações e Informática
2013

**Ricardo Jorge
Magalhães de Matos**

Redes em Malha sem Fios baseadas em Contexto

Context-Based Wireless Mesh Networks

Universidade de Aveiro Departamento de
Eletrónica, Telecomunicações e Informática
2013

Ricardo Jorge
Magalhães de Matos

Redes em Malha sem Fios baseadas em Contexto

Context-Based Wireless Mesh Networks

Tese apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Doutor em Engenharia Eletrotécnica, realizada sob a orientação científica da Professora Doutora Susana Isabel Barreto de Miranda Sargento, Professora Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

O trabalho desenvolvido no decorrer desta tese foi suportado pela Rede de Excelência EURO-NF (European Network of the Future), financiada pela Comunidade Europeia, e pela Bolsa de Doutoramento SFRH/BD/45508/2008, financiada pela Fundação para a Ciência e Tecnologia (FCT).

o júri / the jury

presidente / president

Professor Doutor Artur Manuel Soares da Silva
Professor Catedrático da Universidade de Aveiro

Professor Doutor Paulo da Costa Luís da Fonseca Pinto
Professor Catedrático da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Professora Doutora Marília Pascoal Curado
Professora Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Professora Doutora Solange Rito Lima
Professora Auxiliar da Escola de Engenharia da Universidade do Minho

Professora Doutora Susana Isabel Barreto de Miranda Sargento
Professora Auxiliar da Universidade de Aveiro (Orientadora)

Professor Doutor Paulo Jorge Salvador Serra Ferreira
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Este trabalho assinala o fim de mais uma etapa da minha vida académica, mas não o teria conseguido concluir sem a contribuição de algumas pessoas. Por isso, não queria deixar de expressar aqui os meus mais sinceros agradecimentos a todos os que, direta ou indiretamente, colaboraram no desenvolvimento desta Tese.

À minha orientadora Prof. Dra. Susana Sargento pela disponibilidade para esclarecer as dúvidas existentes e partilhar os seus conhecimentos, e pela criatividade e inspiração que sempre me transmitiu ao longo do período de desenvolvimento da Tese. Revelou-se importante o seu espírito crítico para a correta abordagem dos desafios encontrados ao longo desta Tese.

Ao Instituto de Telecomunicações de Aveiro por me ter oferecido todo o apoio e condições necessárias para o correto desenvolvimento do meu trabalho. Destaco os colegas do grupo Network Architectures and Protocols (NAP), principalmente o Nuno Coutinho, André Cardote, Tiago Condeixa, André Reis, João Soares e Lucas Guardalben pelas discussões científicas, e os alunos de mestrado Carlos Marques e Hélder Bastos pelos resultados que me ajudaram a obter.

Ao Prof. Dr. Kurt Tutschku pela partilha de conhecimentos e experiência.

À Fundação para a Ciência e Tecnologia (FCT) pela bolsa de doutoramento que me foi atribuída, e à Rede de Excelência EURO-NF (European Network of the Future) pelo apoio finaceiro prestado.

Aos meus Pais, António e Maria Estela, e Irmão, Tiago, pelo incansável apoio, paciência e motivação que sempre me deram durante o desenvolvimento desta Tese, assim como a todos os meus Amigos e restante Família.

Palavras-chave

Contexto; Rede em Malha sem Fios; Virtualização de Rede; Controlo de Rede baseado em Contexto; Controlo e Gestão de Rede Distribuído;

Resumo

Na sociedade actual, novos dispositivos, aplicações e tecnologias, com capacidades sofisticadas, estão a convergir na mesma infra-estrutura de rede. Os utilizadores são também cada vez mais exigentes nas suas preferências e expectativas pessoais, desejando conetividade à Internet em qualquer hora e lugar. Estes aspectos têm desencadeado muitos esforços de investigação, dado que a Internet atual está a atingir um ponto de rutura ao tentar promover flexibilidade para os utilizadores e lucros para os operadores, enquanto lida com as exigências complexas associadas à recente evolução.

Em sintonia com a linha de investigação para a Internet do futuro, muitas soluções têm sido propostas para melhorar as arquiteturas e protocolos da Internet atual, de forma a torná-los sensíveis ao contexto, isto é, adaptá-los dinamicamente à alteração da informação que caracteriza qualquer entidade de rede. Neste sentido, a presente Tese propõe uma nova arquitetura que permite criar várias redes com diferentes características de acordo com o contexto das mesmas, sobre uma única rede em malha sem fios (WMN), cuja infra-estructura e protocolos são muito flexíveis e auto-adaptáveis.

Mais especificamente, esta Tese modela o contexto dos utilizadores, que pode abranger as suas preferências de segurança, custo e mobilidade, capacidades dos seus dispositivos ou requisitos de qualidade dos seus serviços, de forma a transformar uma WMN num conjunto de redes lógicas. Cada rede lógica é configurada para satisfazer um conjunto de necessidades de contexto do utilizador (como exemplo, suporte de mobilidade elevada e de baixa segurança). Para implementar esta arquitetura centrada no utilizador, esta Tese utiliza a virtualização de redes, que tem muitas vezes sido defendida como um meio para implementar arquiteturas e serviços de rede de uma forma independente, enquanto permite uma gestão dinâmica dos recursos. Desta forma, a virtualização de redes pode permitir uma configuração flexível e programável de uma WMN, a fim de ser partilhada por várias redes lógicas (ou redes virtuais - VNs). Além disso, o grau de isolamento introduzido pela virtualização de redes pode ser utilizado para diferenciar os protocolos e mecanismos de cada VN baseada em contexto.

Esta arquitetura levanta vários desafios para controlar e gerir as VNs em tempo real, e em resposta à dinâmica dos utilizadores e da WMN. Neste contexto, abordamos os mecanismos para: *(i)* descobrir e selecionar a VN a atribuir a um utilizador; *(ii)* criar, adaptar e remover as topologias e rotas das VNs. Também exploramos a possibilidade de considerar a taxa de variação dos requisitos de contexto dos utilizadores de forma a melhorar o desempenho e reduzir a complexidade do controlo e gestão das VNs. Finalmente, devido ás limitações de escalabilidade das soluções de controlo centralizadas, propomos um mecanismo para distribuir as funcionalidades de controlo ao longo das entidades da arquitectura, que podem cooperar para controlar e gerir as VNs de uma forma distribuída.

Keywords

Context; Wireless Mesh Network; Network Virtualization; Context-Aware Network Control; Distributed Network Control and Management;

Abstract

In the modern society, new devices, applications and technologies, with sophisticated capabilities, are converging in the same network infrastructure. Users are also increasingly demanding in personal preferences and expectations, desiring Internet connectivity anytime and everywhere. These aspects have triggered many research efforts, since the current Internet is reaching a breaking point trying to provide enough flexibility for users and profits for operators, while dealing with the complex requirements raised by the recent evolution.

Fully aligned with the future Internet research, many solutions have been proposed to enhance the current Internet-based architectures and protocols, in order to become context-aware, that is, to be dynamically adapted to the change of the information characterizing any network entity. In this sense, the presented Thesis proposes a new architecture that allows to create several networks with different characteristics according to their context, on the top of a single Wireless Mesh Network (WMN), which infrastructure and protocols are very flexible and self-adaptable.

More specifically, this Thesis models the context of users, which can span from their security, cost and mobility preferences, devices' capabilities or services' quality requirements, in order to turn a WMN into a set of logical networks. Each logical network is configured to meet a set of user context needs (for instance, support of high mobility and low security). To implement this user-centric architecture, this Thesis uses the network virtualization, which has often been advocated as a mean to deploy independent network architectures and services towards the future Internet, while allowing a dynamic resource management. This way, network virtualization can allow a flexible and programmable configuration of a WMN, in order to be shared by multiple logical networks (or virtual networks - VNs). Moreover, the high level of isolation introduced by network virtualization can be used to differentiate the protocols and mechanisms of each context-aware VN.

This architecture raises several challenges to control and manage the VNs on-demand, in response to user and WMN dynamics. In this context, we target the mechanisms to: *(i)* discover and select the VN to assign to an user; *(ii)* create, adapt and remove the VN topologies and routes. We also explore how the rate of variation of the user context requirements can be considered to improve the performance and reduce the complexity of the VN control and management. Finally, due to the scalability limitations of centralized control solutions, we propose a mechanism to distribute the control functionalities along the architectural entities, which can cooperate to control and manage the VNs in a distributed way.

Contents

List of Figures

List of Tables

Acronyms

AAA Authentication, Authorization and Accounting

ABC Always Best Connected

ACK Acknowledge

AODV Ad hoc On-Demand Distance Vector

APs Access Points

ARP Address Resolution Protocol

BL Backward Learning

BLI Backward Learning Information

CAPEX CAPital EXpenditures

CBR Constant Bit Rate

CoA Care-of-Address

CPU Central Processing Unit

CXTP Context Transfer Protocol

DCF Distributed Coordination Function

DES Data Encryption Standard

DHTs Distributed Hash Tables

DSDV Destination-Sequenced Distance Vector

DSR Dynamic Source Routing

E2E End-to-End

ED Extra E2E Delay

EIDs End-point-IDentifiers

EH Extra security Header

ETT Expected Transmission Time

ETX Expected Transmission Count

FA Foreign Agent

FCFS First-Come First-Serve

FCT Fundação para a Ciência e Tecnologia

FDMA Frequency Division Multiple Access

FEC Forward Error Correction

FL Forward Learning

FLI Forward Learning Information

FTP File Transfer Protocol

gARP gratuitous ARP

GIST General Internet Signaling Transport

GoP Group of Picture

GPS Global Positioning System

GTK Group Transient Key

HA Home Agent

HO Handoff

I/O Input/Output

iAWARE Interference-Aware Routing Metric

ID IDentifier

IEEE Institute of Electrical and Electronics Engineers

ILA Interference-Load Aware Routing Metric

INM In-Network Management

IP Internet Protocol

IPTV IP TeleVision

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

ISPs Internet Service Providers

L2 Layer-2

L3 Layer-3

LANs Local Area Networks

LTE Long Term Evolution

MAC Medium Access Control

MANETs Mobile Ad Hoc Networks

MIC Metric of Interference and Channel Switching

MIP Mobile IP

MIT Massachusetts Institute of Technology

MMAC Multi-channel MAC

MOS Mean Opinion Score

NAT Network Address Translation

NSIS Next Step in Signaling

NSLP NSIS Signaling Layer Protocol

NTLP NSIS Transport Layer Protocol

OLSR Optimized Link State Routing

OPEX OPerational EXpenditures

OS Operating System

OTcl Object-oriented TcL

P2P Peer-to-Peer

PC Personal Computer

PDA Personal Digital Assistant

PHY Physical

PSNR Peak Signal-to-Noise-Ratio

PTK Pairwise Transient Key

QoE Quality-of-Experience

QoS Quality-of-Service

QSPEC QoS Specification

RED Random Early Detection

SC Semantic Cluster

SCV Squared Coefficient of Variance

SHA Secure Hash Algorithm

SINR Signal-to-Interference-and-Noise-Ratio

SIP Session Initiation Protocol

SNR Signal-to-Noise-Ratio

SSCH Slotted Seeded Channel Hopping

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

VLAN Virtual Local Area Network

VMM Virtual Machine Monitor

VNs Virtual Networks

VoD Video-on-Demand

VoIP Voice-over-IP

VPNs Virtual Private Networks

WCETT Weighted Cumulative Expected Transmission Time

WiMAX Worldwide Interoperability for Microwave Access

WLANs Wireless Local Area Networks

WMNs Wireless Mesh Networks

WPANs Wireless Personal Area Networks

Chapter 1

Introduction

In this introductory chapter, we define the research context that has driven the Thesis, along with the motivation to develop a novel user-centric context-aware architectural paradigm to be applied to wireless mesh networks through network virtualization. This chapter describes the approach and goals that guided the work performed under the Thesis scope. Finally, we present the contributions that resulted from the concepts explored during the Thesis period, followed by the overview of the Thesis structure.

1.1 Research Scope

The daily routine of humans is driven by a set of different factors that allow them to successfully enjoy their life by interacting and interfering with the surrounding society. These factors can span from: *(i)* the same base language that humans speak; *(ii)* their ordinary vision of how the world works; *(iii)* the common understanding of their rights and obligations; *(iv)* the clear notion of what is (or can be) good or wrong; etc. All of these factors, or *context*, allow humans to easily convey and share ideas with each other, or to properly and dynamically adapt their normal ways of living. Beyond the daily life of humans, there are multiple fields that take advantage of implicit *context* information, available in the environments upon which they act, in order to set up or adapt the strategies to achieve their aims.

In the computing and networking fields, which are the background of the presented Thesis, the description and synthesization of the word *context* is a very challenging task, and it has already led to a multitude of definitions. Although the first definition was provided by [1], the most widely agreed one was stated by [2]: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between an user and an application, including the user and applications themselves*". According to the above definition, the *context* of an user that accesses to the Internet needs to be defined in a multi-dimensional way, since it encompasses several parameters, such as: *(i)* the user location and his/her preferences in terms of, e.g., mobility, price or security; *(ii)* the required Internet service or application, which can have strict quality requirements; *(iii)* the user device and its connectivity and battery capabilities; *(iv)* the status of the wireless/wired technologies and networks that provide Internet access.

The recent explosion of user-friendly portable devices, accompanied by the proliferation of high quality multimedia services and the evolution of wireless/wired technologies able to provide ubiquitous Internet access, are triggering many research efforts to replace or enhance the current network architectures and protocols, in order to become *context-aware*. But what does *context-aware* mean? Tightly coupled with the *context* definition, the term *context-aware* was also defined by [2]: "*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the users task*". The materialization of this definition in networking environments is not a straight-forward and completed task; the digital, service and technological innovation is appearing to network designers as a set of challenging *context* dimensions, each one having conflicting or opposite targets, which hampers the development of an overall *context-aware* solution to dynamically configure or adapt the networks to any type of *context* change.

The original Internet was designed over 40 years ago to provide host-to-host communication services. However, it is now facing unprecedent limitations due to the need to enable the convergence of the emergent devices, services and technologies in an all-IP integrated network. Therefore, the current research towards the future Internet [3] has to be seen as an opportunity to consider the term *context-aware* as a undeniable design goal of future Internet paradigms, which can stimulate the *context-aware* networking research. In this whole process, we cannot neglect to mention that the programmability and flexibility enabled by network virtualization [4] can play an important role in the development and materialization of new architectures and protocols for the future Internet. Further, network virtualization will certainly impose more degrees of freedom in potential *context-aware* network solutions.

The multi-availability of *context* parameters that are slowly and steadily flooding the already existing networking environments triggers several questions. How can the networks become *context-aware* in order to personalize and enrich the services of single or group of users, while preserving a high level of network quality? How should the network control plane be designed in order to quickly adapt the networks to any type of *context* change? These are the two main questions that will guide the work performed in this Thesis.

The presented Thesis aims to define a novel *context-aware* paradigm to be applied to the wireless network infrastructure, an undoubtable integrant part of the future Internet, in which we can found a myriad of user *context* sources. Briefly, this paradigm is focused on the quantification of measurable user *context* sources available in wireless environments. This quantified information is subsequently used to dynamically set up several adaptable logical networks, each one specialized to meet a specific set of user *context* features. In such approach, network virtualization will enable a programable and flexible sharing of the wireless network infrastructure by these logical (or virtual) networks, and will ease the configuration of each logical network according to its context purpose. Further, the user *context* regularities and relationships will be understood and embedded in the definition of an autonomous *context-aware* mechanism to discover, select, control and manage virtual networks on-demand.

The wireless connectivity has now reached a state where it is considered to be an indispensable service as electricity or water supply [5]. Therefore, beyond the technical aspects of this Thesis, we aim to enable the context-aware configuration of the wireless networks up to a high degree, enhancing the overall experience perceived by mobile end-users, while still guaranteing an acceptable network quality. This frames the basis of the presented Thesis, in which we will make use of analytical models and simulation tools to assess the potential of this novel *context-aware* paradigm for wireless networks.

1.2 Motivation

Context, as it was defined by [2], is an integral denominator of any network entity, encompassing both explicit factors and more subjective ones. Although context information has always been present in all sorts of networks since the Internet genesis, it is only recently that networking society has devoted special attention to context-aware networking research.

Over the past few years, we have witnessed the appearance of a fast-growing number of: *(i)* mobile devices with interesting capabilities; *(ii)* services and applications with diverse quality characteristics; *(iii)* end-users with a variety of price preferences and special traffic or mobility patterns; *(iv)* high throughput wireless/wired technologies; etc. Due to this whole body of evolution, current Internet-based network architectures are reaching a breaking point trying to reconcile flexibility and productivity for end-users with the growing list of strict requirements of devices, services and other network elements. This complex trade-off is driving a huge amount of context-aware networking research. By tracking, sensing or monitoring context information available in networking environments, new architectures, protocols and mechanisms are arising. These sorts of innovation aim to automatically configure and adapt network connections, routes, paths and control operations to any type of context change without explicit user intervention. Following this way, the research community has a twofold objective: *(i)* increase the usability and effectiveness of the current Internet to deal with the digital, multimedia and technological revolution; *(ii)* start thinking in concrete and appellative solutions to provide the groundwork for the future Internet.

Concerning the new digital era, a variety of user-friendly, cheap and portable devices (spanning from laptops, smart phones, tablet PCs, PDAs, IP cameras, netbooks or sensors) are nowadays penetrating in the daily life of users, offering them distinct services and Internet connectivity in an ubiquitous fashion. It is then important to enhance current network solutions to be aware of the device context. For instance, the knowledge of the processing and buffering capabilities of devices can be useful to adapt the download transmission rate on-demand or improve battery life of devices. Further, the detection of a GPS device system can trigger optimized management functions driven by the accuracy of the user location.

Accompanying this digital evolution, new multimedia applications and services (such as VoIP, IPTV, interactive gaming or e-commerce) are emerging and changing the life experience of users, which can now access, consume and produce audio and video contents in a cost-effective way. From the point of view of the service or content providers, they can try to explore new pathways to increase revenues by programming their applications to meet the audio and visual expectations of end-users. The configuration and adaptation of network transport paths can also be done in a more flexible way by taking into account the specific quality requirements of the user services, which can even improve the network resource usage.

With regard to user preferences in terms of, e.g., mobility, security or price, network providers can start thinking to use them to differentiate the network protocols. Examples span from the proper configuration of different security protocols or policies, which can optimize the control delay and overhead due to security updates. The correct prediction of the user movement can also enable the automatic configuration of the user sessions according to his/her location (e.g, home or work).

The recent plethora of new wireless (UMTS, WLAN, WiMax, Bluetooth or LTE) and wired (gigabit ethernet or optical fiber) technologies, each one providing specific bandwidth, coverage and reliability characteristics and mobility support, are also playing an important role towards the future Internet. This widespread technological evolution, especially at or near the edge of the networks where more users are located, can favor the context-aware networking innovation. For instance, user devices can be equipped with several wireless interfaces and technologies. This can encourage the development of intelligent schemes to select the interface and technology on-demand to be adapted to the current user context, following the ”always best connected” (ABC) paradigm [6].

The low up-front cost and easy deployment of networks at the edge has also to be seen as an opportunity to networking innovation. Due to these appealing economic conditions at the edge, new self-configured and self-organized wireless networks, the Wireless Mesh Networks (WMNs) [7], are gaining a huge attention by the academia, industry and standard organizations. However, WMNs certainly impose more context dimensions that need to be deeply understood to be embedded in an overall context-aware solution for future network architectures.

1.2.1 Wireless Mesh Networks

As a key part of the wireless segment of a network infrastructure, the WMN, in which this Thesis has a strong focus, is composed by a set of mesh routers that automatically establish and interconnect static ad hoc networks, providing not only wireless access for end-users but also wireless infrastructure up to the wired domains. Mesh routers are usually equipped with multiple wireless interfaces, built on either the same or different access technologies, and allowing multi-hop transmissions with less power over short distances.

Three types of WMNs can be identified: infrastructure, client or hybrid [7]. In infrastructure WMNs, heterogeneous mesh routers form a wireless broadband backbone for mesh

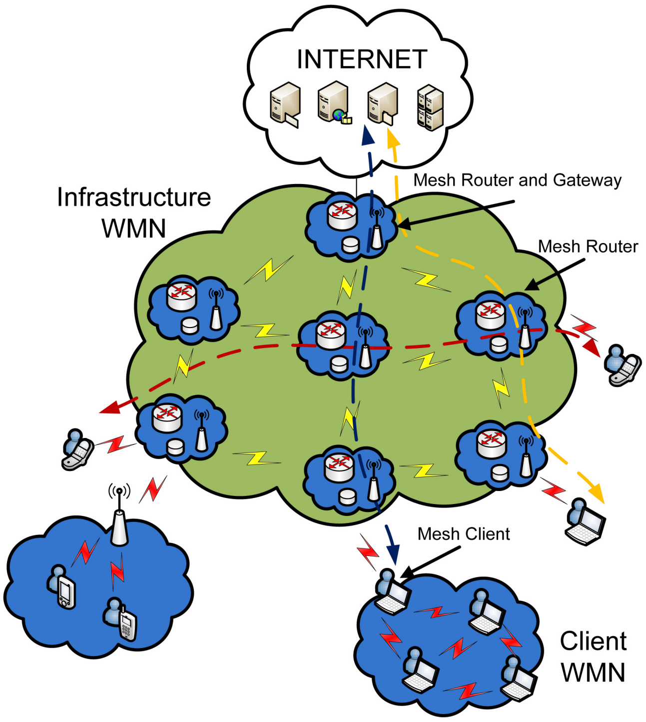


Figure 1.1: Architecture of a Hybrid Wireless Mesh Network.

clients. An infrastructure WMN is meant to be self-configuring, self-forming and self-healing, offering gateway functionalities for connections to other networks (both wired and wireless). Client WMNs are ad hoc networks formed by clients among themselves. In a client WMN, no dedicated routers or infrastructure exists, and thus, the clients have to be self-configuring and act as routers. Hybrid WMNs combine the advantages of the previous ones. In a hybrid WMN (an example is shown in Fig. 1.1), both mesh clients and routers participate in the traffic routing, which increases the connectivity coverage.

Recently, WMNs are attracting significant research efforts. They have many favorable characteristics, such as self-organization, easy maintenance, high coverage, limited mobility and low energy consumption of mesh routers. Therefore, WMNs have been advocated as a cost-effective approach to support high speed last mile connectivity and ubiquitous broadband access in the context of home networking, community networking or building automation. Further, the self-* and flexible properties of the WMN infrastructure and protocols require novel control and management network solutions that leave behind the limitations of the ones for single-hop and ad hoc networks. Due to these special characteristics, WMNs have been viewed as an important component of the future Internet, since they can provide support for new communication and architectural paradigms; and, within the scope of this Thesis, they can be envisaged for new context-aware network paradigms.

1.2.2 Context-Aware Wireless Mesh Networking

WMNs can be accessed by a multi-variety of mesh clients with particular connectivity, traffic and mobility patterns, devices with demanding technological and battery requirements, and services with challenging Quality-of-Service (QoS) needs.

In the last years, several solutions have been proposed to ease the WMN control and management by monitoring context information, and then, configuring and adapting WMN nodes or protocols to react upon context change. These context-aware solutions can span from: *(i)* incorporation of QoS metrics in the design of WMN routing protocols; *(ii)* dynamic adaptation of the power consumption of WMN nodes to meet specific energy-efficiency purposes; *(iii)* decrease of the data generation rate to handle the congestion at a specific WMN link; etc. Despite the potential of these context-aware solutions to smooth the WMN control and management, they are focused on single (or a very small set of) context parameters, without providing an integrated vision of the impact of context availability on WMNs.

Thus, we aim to combine a more broad set of context dimensions in the design of novel context-aware paradigms to be applied to WMNs. One of these paradigms can be the dynamic splitting of a physical WMN infrastructure into a set of adaptable logical networks, each one connecting the nodes in which users with similar context needs are attached to (see Fig. 1.2). Each logical network can be configured in terms of, e.g., protocols, resources and wireless channels, to properly meet the context requirements of its associated users.

Beyond the personalization and enrichment of the user connectivity according to the user context needs, this paradigm can bring other appealing advantages in the scope of the WMN control and management. For instance, new users arriving or moving inside the WMN can be connected to logical networks already available in their attached nodes that meet their context requirements, which can decrease the control delay and overhead involved in the creation of new network connections and paths. Further, by forwarding data flows with the same context requirements through the same routes, transport connections or channels (which can be dynamically changed in WMN environments), the level of interference among communications of different types can be decreased. Finally, the control processes of each logical network can be differentiated according to its context purpose.

The design and implementation of the aforementioned context-aware WMN paradigm will certainly make use of the highly attractive properties of network virtualization [4]. Network virtualization has often been advocated as a mean to deploy independent network architectures and services towards the future Internet, while allowing a flexible and dynamic resource management. Despite the benefits that can arise from the use of network virtualization to materialize this WMN paradigm, it introduces several challenges that need to be understood and addressed.

1.2.3 Network Virtualization

Network virtualization has been identified as a key technology to overcome the widely diagnosed Internet ossification, and so, as a clear path to enable the migration from the current to a future Internet. Several characteristics of network virtualization have witnessed the previous sentence: *(i)* it allows the parallel use or multiplexing of network resources, leveraging both resource isolation or aggregation in an easy programmable manner; *(ii)* it enables the building of different virtual networks on top of the same physical substrate, possible connecting networks provided by different infrastructure providers (which can increase the reliability of communications and reduce the deployment costs for network operators); *(iii)* it enables the simultaneous support of emerging and innovative paradigms and architectures, providing enough isolation to separate experimental from real traffic; *(iv)* it introduces a high level of flexibility to move or exchange virtual resources and flows at multiple levels.

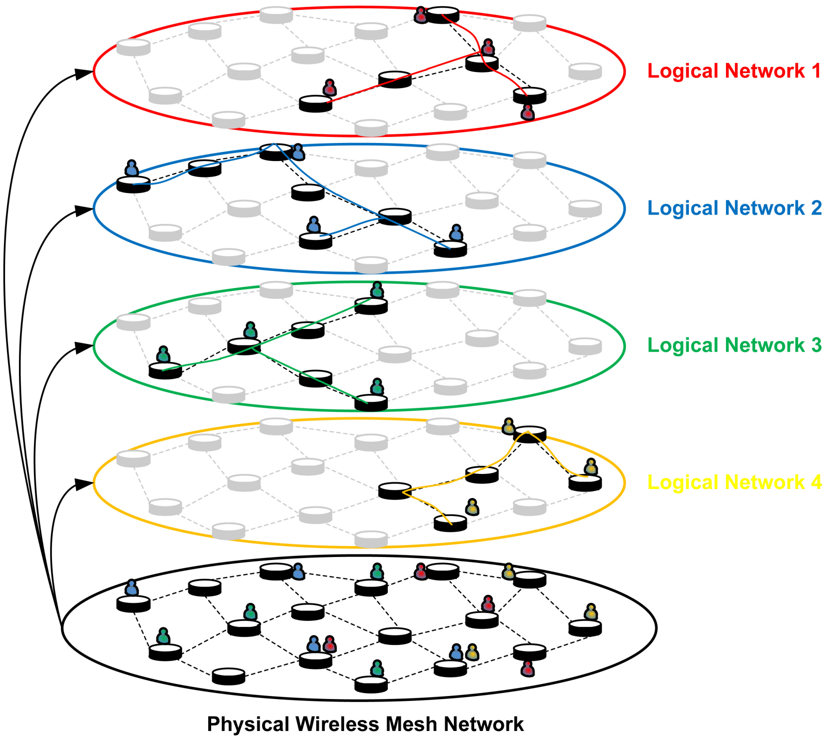


Figure 1.2: Novel Context-Aware WMN Paradigm.

The features of network virtualization make it an appealing technology to enable the materialization of the context-aware WMN paradigm depicted above. In the one hand, network virtualization can allow a flexible and programmable configuration of the WMN infrastructure in order to be shared by multiple context-aware logical networks (or virtual networks - VNs). With network virtualization, each physical WMN node can be the physical substrate for distinct VN nodes, each one being configured according to the context and resource needs of its associated VN. On the other hand, the level of isolation and transparency introduced by network virtualization can be exploited to differentiate the protocols and mechanisms running at each context-aware VN.

The use of network virtualization as a mean to split a physical WMN infrastructure into a set of context-aware logical networks leads, however, to several issues that were still not addressed or solved by the research community. The introduction of wireless virtualization imposes the need of a dynamic mapping, scheduling and switching of mesh routers, interfaces and channels to serve the virtual wireless networks. Moreover, an intelligent mechanism is required to disseminate specific VN and context information in the WMN, in order to quickly discover and select the best fitting VN to assign to an user (or even to move the user from a VN to a better fitting one in case of context change). It is also important to automatically create, configure, update, extend or remove VNs on-demand to react to any type of context change.

These VN control and management mechanisms can be performed by a single entity that controls the entire architecture. Nevertheless, centralized approaches have well-known limitations, such as limited scalability or singles points of failure. Therefore, it may worth distributing several control knowledge and functionalities along the different architectural entities, which can then dynamically cooperate to autonomously control and manage the context-aware VNs on-demand.

In summary, several challenges have arisen in order to effectively understand and deploy new communication and architectural paradigms that deal with the multi-context availability brought by the emergent digital, multimedia and technological evolution. Due to the attractive features and economic conditions of WMNs, they have to be considered a relevant part of the wireless segment of future network architectures. Therefore, we must carefully and deeply understand the context dimensions that are becoming part of WMNs, and shape this context information towards deploying a more comprehensive and flexible context-aware approach to enrich wireless mesh networking. In such approach, an important role needs to be assigned to network virtualization, which is widely considered as one of the key enabling technologies of the future Internet.

1.3 Thesis Approach & Objectives

During the last decade, a plethora of new devices, applications and technologies with sophisticated capabilities have increasingly collided with the resistance of the Internet to adapt its structure accordingly. In addition, the Internet usage is becoming more and more mobile, with users desiring Internet connectivity everywhere and anytime, which is bringing the wireless world closer to our daily life. Fully aligned with the future Internet research, new communication and architectural paradigms need then to be designed to deal with the heterogeneity and complexity of wireless networks. In the context of wireless networks, the high level of adaptation, self-configuration and self-organization of WMNs will indeed assist the definition and deployment of these paradigms.

In the light of the outlined motivation, this Thesis defines an user-centric context-aware paradigm to be applied to WMNs through network virtualization. By taking care of the context of mesh clients (or users), which can span from special security, price and mobility preferences, capabilities of their devices or requirements of their services, we make use of network virtualization to split a WMN into a number of context-aware VNs. Each VN can be instantiated by a specific VN provider or network operator, and it is specialized to meet the same (or similar) context requirements of a group of users, which can be assigned to VNs matching their context. Each VN runs its context-aware services and protocols with the level of transparency and isolation required. In order to make the VNs usable, scalable and adaptive, they need to be discovered, created, adapted and removed on-demand to dynamically react to any type of context change.

In summary, this Thesis argues for a general context-aware WMN architecture built through network virtualization, and capable to provide support for highly diverse users, devices, applications, technologies, network protocols or providers. Our main goal is to personalize WMNs by ensuring that multiple services are provided under specific constraints, device requirements are fulfilled in the best possible way, and users perceive a high Quality-of-Experience (QoE). At the same time, we aim to guarantee an effective WMN resource usage in order to maximize the revenues for operators, which can continue to meet scalability, reliability and security requirements.

Bringing context-awareness and network virtualization together can unveil several challenges that require a deeper analysis. The first question concerns in the definition of a model to relate the distinct variants of user context, which may be highly diverse with unspecified compatibility requirements, and to map such variants into proper VN features. Further, we must realize what is the best mechanism to select the VN to assign to a particular user. Such mechanism cannot be properly handled unless considering a combined view of the architecture, which can lead to a compromise among the user context needs and expected QoE, the features and conditions of the candidate VNs, and the WMN resource availability. Finally, as a driving force of a distributed and flexible VN control and management that we aim for, the different architectural entities must cooperate to enable the autonomous discovery, selection, creation, adaptation and remotion of VNs on-demand to cope with context changes.

Focusing on the presented challenges, the main objectives of this Thesis are the following:

- As a starting point, we must understand the main guidelines, goals, scenarios and applications of a context-driven architecture for WMNs, built through network virtualization. This will inevitably lead to the definition of the key functionalities of the data and control planes of the architecture, along with the stipulation of the main important roles of each network entity in the overall architectural functionality;

- Concerning the architectural data plane, it is first required to quantify and model the context information that characterizes mobile mesh clients, which can span from security, price and mobility preferences, QoS requirements of their services, or QoE expectations. Since each VN is created and configured to meet specific levels of user context needs, we must then define the mechanisms to automatically map such needs into proper context-aware VN structures, running protocols or assigned resources. With regard to the routing protocol running at each VN, we aim to provide a strategy to dynamically select the routes according to the specific context and QoE feedbacks of ongoing VN users. Finally, it is important to evaluate the impact of applying network virtualization to enable non-interfering and personalized WMN data communications;

- Focusing on the context-aware architectural control, we must consider certain levels of flexibility in the context requirements of users, in order to allow users' associations to VNs that do not exactly fit all their requirements, but may incorporate these users if their context flexibility allows to. This way, we can increase the usability of VNs already available in the WMN, while avoiding the constant creation of new VNs to serve single users, which cannot be always performed due to its complexity or unavailability of resources. Further, the context-aware metrics to select the best fitting VN to assign to an user, or the best WMN path to adapt or create a specific VN, have to meet a trade-off among the: *(i)* context needs and QoE expectations of users; *(ii)* context features and the overall quality level of the candidate VNs for the user; *(iii)* resource constraints posed by the WMN infrastructure in order to increase the WMN capacity;

- In the scope of an autonomous VN control and management, we must define a mechanism to distribute the control knowledge and functionalities along the WMN nodes, and their supported virtual nodes. With such control distribution, an user can be automatically assigned to any fitting VN available in the WMN node where he/she is connected to, or it can be triggered the smooth reconfiguration of any fitting VN available in the user neighborhood. Both these local mechanisms are envisaged to enable the fast user association to any fitting VN. Beyond these local VN control and management processes, the architectural entities must be able to cooperate in the global and context-aware discovery, creation, adaptation and remotion of VNs on-demand;

- Since the VNs can be instantiated by specific VN providers or network operators with their own administration, we must define a scheme to enable the cooperation among VNs characterized by similar context purposes. Such approach has a two-fold objective: *(i)* in case of a small change in a specific user need, the re-association of the user to another VN that better meets the new user context needs, can be quickly performed by inspecting the information about the location of similar VNs; *(ii)* in case of the need to start a global VN discovery in the WMN, the cooperation among VNs with similar context patterns, can lead to the fast and context-driven redirection of the VN discovery process.

Beyond the technical objectives, this Thesis is tightly motivated by the ambition to stimulate the integration of WMNs as a high quality network part of the future Internet architectures and virtual operator environments. This way, we aim to make use of an user-centric context model to enrich the data and control operations of WMNs through network virtualization. This Thesis is an important step to empower new paradigms, that currently do not exist or are still in their infancy, in order to be integrated in future network architectures.

1.4 Thesis Contributions

The contributions of this Thesis are not restricted to a single issue or thematic, but span from the definition, modeling and evaluation of a new context-aware architecture to be applied to WMNs, along with its intrinsic network discovery, selection, control and management functions. Table 1.1 summarizes the publication contributions of this Thesis.

The most substantial contribution of this Thesis lies in the identification and understanding of the manifold context dimensions that characterize mobile mesh clients, which led to the definition of a context-aware architectural paradigm to be applied to WMNs. Such paradigm is driven by a general modeling of user-centric context, and by the flexibility and programmability provided by network virtualization. The main guidelines and control challenges of this architecture were first presented at *ICST MONAMI'09*, as "*Context-Aware Connectivity and Mobility in Wireless Mesh Networks*" [8], and later at the *Workshop on Future Internet Architectures* held at *ICT MobileSummit'09*, as "*Context-Based Wireless Mesh Networks: A Case for Network Virtualization*" [9]. This last contribution was then consolidated and extended in [10], which was published in the *Springer Telecommunications Systems* journal.

The use of network virtualization imposed the need to define an analytical model able to measure its impact on the provision of personalized and isolated WMN data communications. The outputs of such model, presented as the limits of the multi-VN architecture in order to accomplish feasible delays for different data communication requirements, resulted in the publication entitled "*Data Communications over Context-Based WMNs: Delay Performance Evaluation*" [11], presented at the *BoD Workshop* hosted at *IEEE NOMS'10*. The enhancement of the model, as a mean to evaluate the delay performance of a basic distributed VN control and management mechanism, led to the contribution entitled "*Context-Based Connectivity over Multi-Virtual Wireless Mesh Networks: Analytical Study*" [12], published in *IEEE ISCC'10*. This distributed VN control and management mechanism included: *(i)* the processes required to locally or globally perform the discovery of exactly fitting VNs for users, and the extension of such VNs up to the nodes where users are connected to; *(ii)* a distributed control structure, based on the context-aware cooperation among distinct VNs, in order to accelerate the global VN discovery in the WMN. Later, the whole analytical

Table 1.1: Publication Contributions related to the presented Thesis.

Type	**Year**	**Title**	**Venue**
Workshop	2009	Context-Based Wireless Mesh Networks: A Case for Network Virtualization	Workshop on Future Internet Architectures @ ICT MobileSummit
	2010	Data Communications over Context-Based WMNs: Delay Performance Evaluation	BoD @ IEEE NOMS
	2010	Context-Based Connectivity and Characterization of Wireless Mesh Networks: Simulation Study	MCECN @ IEEE GLOBECOM
	2012	Quality of Experience-Based Routing in Multi-Service Wireless Mesh Networks	ViOPT @ IEEE ICC
Conference	2009	Context-Aware Connectivity and Mobility in Wireless Mesh Networks	ICST MONAMI
	2010	Context-Based Connectivity over Multi-Virtual Wireless Mesh Networks: Analytical Study	IEEE ISCC
	2011	Context-Based Connectivity in Wireless Mesh Networks: Analytical vs Simulation Studies	IEEE EUROCON
	2011	Distributed Control and Management of Context-Based Wireless Mesh Networks	ICST MONAMI
	2012	On-Demand Selection, Control and Management of Context-Based Wireless Mesh Networks	IEEE ICC
Journal	2009	Context-Based Wireless Mesh Networks: A Case for Network Virtualization	Telecommunication Systems @ Springer
	2011	Analytical Modeling of Context-Based Multi-Virtual Wireless Mesh Networks	Ad Hoc Networks @ Elsevier
	2012	Distributed Approach to Control and Manage Context-Based Multi-Virtual Networks	MONET @ ACM/Springer
	2013	Distributed Architecture to Build and Control Context-Based Wireless Networks Through Virtualization	Under Preparation
Book Chapter	2011	User-Driven Multi-Access Communications Through Virtualization	Wireless Multi-Access Environments and QoS Provisioning @ IGI Global

study, together with an optimization approach for VN embedding, motivated the joint chapter "*User-Driven Multi-Access Communications Through Virtualization*" [13], published in the *IGI Global Wireless Multi-Access Environments and QoS Provisioning* book.

The potential and usefulness of the proposed paradigm for WMNs, along with the promising analytical results achieved, paved the way to the development of a simulation platform to implement interesting architectural functionalities and evaluate their delay and overhead performance. The simulation results, describing the impact of network virtualization on the delay and throughput of data communications and the delay and overhead of the proposed control mechanism to associate users to exactly fitting VNs, were presented at the *MCENC Workshop* co-located with *IEEE GLOBECOM'10*, as "*Context-Based Connectivity and Characterization of Wireless Mesh Networks: Simulation Study*" [14].

The comparison between the modeling and simulation efforts, which allowed us to validate the analytical model, led to the paper entitled "*Context-Based Connectivity in Wireless Mesh Networks: Analytical vs Simulation Studies*" [15], published in *IEEE EUROCON'11*, and then to an extended article "*Analytical Modeling of Context-Based Multi-Virtual Wireless Mesh Networks*" [16], published in the *Elsevier Ad Hoc Networks* journal. In such journal article, we evaluated the data and control architectural planes, through analytical models and simulation platform, even in the presence of mobility of mesh clients or using real WMN topologies (e.g., the Funkfeuer Vienna WMN topology [17]).

The continuous definition of rules to map user context into proper context-aware VN structures, running protocols or assigned resources, along with the exploration of how the user context flexibility can increase the usability of VNs already available in the WMN, encouraged the publication entitled "*On-Demand Selection, Control and Management of Context-Based Wireless Mesh Networks*" [18], presented at *IEEE ICC'12*. In such paper, we proposed a context-aware metric for VN selection in order to reduce the number of user re-associations to other fitting VNs in case of user context change. Moreover, we evolved the basic distributed VN control and management mechanism, previously defined, to be able to create and remove VNs.

Tightly coupled with the context-aware VN configuration, we designed a reinforcement learning scheme to optimize the VN route selection on-demand based on QoE feedbacks of ongoing VN users. This work was presented at the *ViOPT Workshop* held at *IEEE ICC'12*, as "*Quality of Experience-Based Routing in Multi-Service Wireless Mesh Networks*" [19].

The results of the aforementioned basic distributed VN control and management mechanism pointed to a set of improvements brought by distributed control, but also suggested that we needed greater insight on designing a more intelligent distributed mechanism to control and manage VNs on-demand. Therefore, we presented at *ICST MONAMI'11*, as "*Distributed Control and Management of Context-Based Wireless Mesh Networks*" [20], the mechanisms to distribute several control knowledge and functionalities among the distinct WMN nodes. We also enhanced the distributed context-aware control structure to enable the cooperation among VNs characterized by similar context purposes. Finally, we proposed a context-aware metric for VN and WMN path selection that is able to cope with user context change and WMN resource availability.

This last contribution was then extended, resulting in the publication with title "*Distributed Approach to Control and Manage Context-Based Multi-Virtual Networks*" [21], published in the *ACM/Springer MONET* journal. In such journal article, we detailed the overall framework to enable the distributed cooperation among the distinct architectural entities in order to autonomously discover, select, create, adapt and remove VNs on-demand to react to any type of user context change or mobility. Further, we defined and validated an analytical model to evaluate several performance aspects of the distributed VN control and management framework when applied to specific WMN scenarios, and compared it against centralized and decentralized schemes.

In the following, we performed an extensive evaluation of the data and control planes of the multi-VN architecture through simulation tools. The obtained results were compared against other possibilities to control and manage the architecture in a centralized or distributed way. This final study led to the journal publication "*Distributed Architecture to Build and Control Context-Based Wireless Networks Through Virtualization*" [22].

Part of the research presented in this Thesis was performed in the scope of two internal projects of the "*FP7 EURO-NF (European Network of the Future)*" Network of Excellence [23] funded by the European Commission. Both projects enabled fruitful discussions and suggestions towards the definition of the architecture that guides this Thesis. The first project, "*Context-Aware Mobility in Wireless Mesh Networks (CAMYN)*" (2008-2009) [24], aimed tot build an optimized context- and mobility-driven architecture for WMNs, by exploiting Peer-to-Peer (P2P), self-organized and mobility prediction concepts, in order to solve several inter-domain and scalability shortcomings of WMNs. The second project, "*Multi-Overlays for Multi-Homing: A Wireless Mesh Network Use Case (MOMO)*" (2009-2011) [25], made use of economic pricing and accounting strategies for network and services, in order to investigate the potential of multi-homing and multi-path in WMNs to increase the overall user QoE.

Finally, the key achievements and conclusions of this Thesis were also used as important inputs for: *(i)* the discussions performed in the projects "*Cross-Layer Optimization in Multiple Mesh Ubiquitous Networks (Ubiquimesh)*", PTDC/EEA-TEL/105709/2008 (2010-2012) [26], and "*User-centric Mobility Management (UMM)*", PTDC/EEA-TEL/105472/2008 (2010-2013) [27]; *(ii)* the definition of the goals, plans and methods of the project "*Multiple Context-based Wireless Mesh Networks (MC-WMNs)*", PTDC/EEA-TEL/120176/2010 (2012-2014) [28]. These three projects are funded by the Fundação para a Ciência e Tecnologia (FCT).

1.5 Thesis Overview

The work presented in this Thesis is structured around the modeling and use of context information to enrich WMNs through network virtualization, along with the definition of a distributed VN control and management mechanism to react to context dynamics. The structure of this Thesis consists in the introduction and related work chapters where the concepts explored in this Thesis are framed, four central chapters in which the most important contributions of this Thesis are described, and the final conclusion chapter.

The introductory Chapter defines the research context that has driven the Thesis, along with the motivation to design novel context-aware paradigms to be applied to WMNs. Chapter 1 also outlines the approach and goals that guided the work performed under the Thesis scope, and highlights the contributions stemming from the explored concepts.

In Chapter 2, we first provide a brief overview on WMNs, comprising the description of their key characteristics, scenarios, applications and control approaches, along with the survey of several real world deployments, testbeds and future research directions. Then, this Chapter provides an insight on the added value brought by network virtualization, showing its most important techniques and current trends; and, within the scope of this Thesis, it gives a special emphasis to wireless network virtualization. In the following, Chapter 2 dives into the context and context-aware thematic, presenting a literature review of distinct WMN context-aware approaches. With regard to QoE, which is a particular component of the user context, it surveys the most important approaches for QoE-aware wireless mesh networking. Finally, this Chapter summarizes several solutions and architectures to decentralize or distribute the control and management functionalities in the network, especially the ones that are applied to WMNs or are driven by context information. As conclusion, Chapter 2 shows that there is still room for innovation in the context-aware configuration and control of WMNs, and such innovation can be accomplished through the use of network virtualization.

Chapter 3 proposes an architecture to split a WMN into a set of personalized VNs, in order to become feasible to form purpose-oriented networks connecting mesh clients that share the same (or similar) context. It then provides the first introduction of the control and management challenges that are under the vision of this new architectural paradigm, which will be revisited in the following three Chapters. Subsequently, Chapter 3 defines an analytical model for the architecture, which allows us to evaluate the impact of network virtualization on the End-to-End (E2E) delay of WMN context-based communications. This Chapter finishes by validating such model against a simulation study, and by evaluating the impact of using the node positions of a real WMN topology to implement the multi-VN architecture.

Chapter 4 proposes a basic distributed control mechanism to locally or globally discover and extend exactly fitting VNs to assign to users. Closely following the analytical model previously defined, this Chapter enhances such model to assess the delay performance of this distributed control mechanism. Finally, the outputs of the analytical study are compared against the ones obtained through simulation, being also evaluated the architectural control under the influence of mobility of mesh clients.

In Chapters 3 and 4, the concepts of context-aware configuration and selection of VNs frequently reappeared, fostering the need to define more intelligent context-aware mechanisms to be part of the proposed WMN architecture. Therefore, in Chapter 5, we first define several rules to quantify and map user context into proper context-aware VN structures, running protocols or assigned resources. Chapter 5 then proposes a VN selection metric that takes into account the user context flexibility to increase the usability of VNs, while decreasing the need to create new VNs and its associated complexity. The impact of this VN selection metric on the decrease of the number of VN creations is evaluated through both analytical and simulation studies. Tightly coupled with the context-aware VN configuration, Chapter 5 closes with the definition and evaluation of a hop-by-hop reinforcement learning VN routing mechanism based on QoE feedbacks of ongoing VN users.

Due to the appealing conclusions of Chapter 5, Chapter 6 enhances the distributed VN control and management mechanism presented in Chapter 4. Here, we present an overall framework to distribute control knowledge (e.g., VN topology information and indicators of WMN resource availability) and functionalities along the distinct WMN and VN nodes. These network entities can then cooperate to autonomously discover, select, create, adapt and remove VNs on-demand in order to cope with user, VN and WMN context change. The distributed framework for VN control and management also resorts to the flexible cooperation among VNs characterized by similar context purposes in order to accelerate the global VN discovery. In the following, this Chapter defines and validates an analytical model to compare the delay and overhead performance of this distributed control framework against other centralized or decentralized solutions for VN control and management. At the end, Chapter 6 presents extensive evaluations of the data and control planes of the architecture, comparing them with other possibilities to implement the architecture.

As conclusion, Chapter 7 summarizes the overall outcomes of this Thesis in the scope of the context-aware paradigm to be applied to WMNs through network virtualization. Beyond pulling together all the results and achievements in relation to the research goals, this Chapter offers a map for future research that can directly complement or enhance the solutions and mechanisms provided along this Thesis. Finally, it takes a look at several aspects that are intrinsically related to the evolving of the proposed context-aware multi-VN architecture, which can be a good starting point towards its integration in emerging and future network architectures.

Chapter 2

Background & Related Work

This Chapter first presents an overview on WMN and network virtualization concepts, which are, respectively, the physical network infrastructure and the key enabling technology of the architecture proposed and evaluated in the later Chapters. It then focuses on the context and context-awareness thematic, summarizing the drawbacks of several approaches available in the literature in order to encourage the design of a context-aware WMN paradigm that goes beyond these approaches. Finally, it analyzes distinct solutions to decentralize or distribute the control and management functionalities in the network, especially the ones that are relevant to the work presented in this Thesis.

2.1 Introduction

The work of the presented Thesis has a strong focus on WMNs, which have been considered a relevant part of the wireless segment of current and future networking environments. The high flexibility, self-configuration and self-organization of WMNs, along with their attractive economic conditions, make them suitable to deploy and test novel communication and architectural paradigms that are aware of the growing list of strict requirements of mobile users, devices, services and other network elements. In the scope of this Thesis, WMNs are envisaged to provide support for an user-oriented context-aware architectural paradigm. It is then important to frame the reader with the most important features, current trends and open issues of WMNs. This way, Section 2.2 describes the key WMN characteristics, scenarios and real world deployments. This Section also presents several WMN control and management solutions that deal with routing, channel assignment, cross-layer design and mobility-related aspects, since these solutions are relevant to understand the future research directions in the context of WMNs, which are summarized at the end of this Section.

The development and materialization of new Internet-based architectures and protocols that share the same physical network infrastructure, while providing enough flexibility for end-users and profits for network operators, cannot be properly handled without resorting to the highly attractive properties of network virtualization. Network virtualization has often been advocated as a mean to deploy and test independent network architectures and services towards the future Internet, while enabling a flexible, programable and dynamic control and management of the network infrastructure. As this Thesis aims to define an architectural paradigm to be applied to WMNs, an important role then needs to be assigned to network virtualization. To engage the reader with the added value brought by network virtualization, Section 2.3 provides an insight on the most interesting features, concepts and challenges related to network virtualization. Within the scope of this Thesis, this Section also gives a special emphasis to wireless network virtualization.

After presenting the key enabling technology of the context-aware architectural paradigm that we aim to propose in this Thesis, we need to effectively understand the context information available in WMN environments, and how we can use such information to personalize wireless mesh networking. These two actions cannot be handled without performing a deep review of the works and proposals available in the literature that deal with the context-awareness thematic in the scope of WMNs. Following this strategy, we can detect what are the open issues or pitfalls of the works and proposals that consider the WMN context availability as an opportunity to enrich wireless mesh networking, instead of only a challenge that raises the WMN complexity. Section 2.4 gives an overview of related work concerning the context-aware wireless mesh networking, and motivates the need to define more intelligent and comprehensive context-aware architectures or mechanisms to be applied to WMNs. With regard to QoE, which is a particular component of the context characterizing users, Section 2.4 surveys several approaches for QoE-aware wireless mesh networking, and states the main issues that can be addressed in order to go beyond these approaches.

This Thesis aims to leave behind the scalability limitations of centralized architectures for the control and management of nodes and networks. Therefore, Section 2.5 summarizes several literature approaches that can be used to decentralize or distribute the control knowledge and functionalities along distinct network entities. This is performed in the scope of several network control and management mechanisms, especially the ones that are applied to WMNs or can use context information to optimize such mechanisms.

We finish the Chapter in Section 2.6 by providing a summary of the discussed technologies, concepts and trends.

2.2 Wireless Mesh Networks

This Section provides a brief overview on WMNs [7], comprising the description of their key characteristics, scenarios, control and management approaches at different layers, real world deployments and testbeds. At the end, we present several future WMN research directions.

2.2.1 Concept Overview

WMNs are composed by a set of mesh routers and mesh clients. Mesh routers automatically establish a relatively stable ad hoc network, and maintain the mesh connectivity among them in order to form a wireless broadband backbone. In the most common form of WMNs, every mesh router performs relaying of data for other mesh routers (a typical ad hoc networking paradigm), and certain mesh routers also have the additional capability of being Internet gateways. Mesh clients connect to mesh routers using a wireless or a wired link. This way, a WMN can not only provide ubiquitous wireless access for mesh clients, but also wireless infrastructure and gateway/bridge functionalities for connections to other networks (both wired and wireless).

WMNs are meant to be self-configured, self-formed and self-organized. In this sense, mesh routers can be added and removed from the network when needed (or due to hardware failure), and the WMN can still operate without the intervention of any centralized administrative entity. Moreover, WMNs have become more and more popular due to the increasing usage of multiple wireless interfaces, built on either the same or different technologies, and of virtual wireless interfacing techniques.

WMNs extend the coverage range of current wireless networks by using multi-hop paths through shorter link distances and offering a better frequency re-use. Thus, nodes tend to connect to each other through intermediate nodes rather than making a direct connection. This means that the data is forwarded from one device to another until it reaches its destination. As a result, WMNs can achieve a higher throughput without sacrificing their effective radio range, and can cover the same area with less transmission power, which can lead to less wireless interference.

Due to their attractive features, WMNs are gaining a huge attention by the academia, industry and standard organizations. Instead of being another type of ad hoc network, WMNs diversify the capabilities of ad hoc networks, presenting a low up-front deployment cost, easy network maintenance and expansion, limited mobility and low energy consumption by mesh routers. In addition to be widely accepted in the traditional application sectors of ad hoc networks, WMNs are undergoing rapid commercialization in many other scenarios and inspiring numerous applications, such as broadband home networking, community networking, building automation, high-speed metropolitan area networks, enterprise networking and Internet access (particularly in rural areas). At the same time, WMNs are already being used in free wireless access initiatives and testbeds.

The majority of current WMN deployments are based on the IEEE 802.11 standard [29], mainly due to the cheap availability of 802.11 hardware. Since the 802.11 software stack was originally designed for infrastructure Wireless Local Area Networks (WLANs), various modifications are required to use such software in WMNs. These modifications have been investigated in the scope of the IEEE 802.11s [30]. Meanwhile, the knowledge gained by the research and development of IEEE 802.11-based WMNs has lead to the emergence of other standardization groups, such as IEEE 802.15.5 for mesh networking of Wireless Personal Area Networks (WPANs) [31] and IEEE 802.16j involving 802.16 multi-hop relaying [32].

2.2.2 WMN Control & Management

The particular characteristics of WMNs, setting them apart from traditional wireless networks, bring up new challenges and open issues that influence the performance of the WMN communication protocols, network management or reliability assurance. In this Section, we present several WMN mechanisms and approaches available in the literature that deal with routing, channel assignment, cross-layer design and mobility-related aspects, which are relevant topics in the context of the WMN research.

2.2.2.1 Routing Protocols & Metrics

Routing protocols, together with their routing metrics and path computation algorithms, are important components of WMNs to support the requirements of distinct users, services and applications, as well as to improve the WMN operation and performance [33].

Despite the fact that mesh routers are generally stationary, the WMN routes are expected to be unstable, because the quality of WMN links varies over time due to external and internal interference, weather conditions and traffic patterns. This means that frequent changes of WMN links' conditions can cause variations in the quality of WMN routes, and thus, can lead to oscillations that have an adverse effect on the overall WMN performance. These aspects, together with the multi-hop and multi-path nature of WMNs, have motivated the design and development of a variety of new routing protocols and metrics for WMNs.

- **Routing Protocols**

Routing protocols may use different methods to compute the path length, distribute the routing information and coordinate the mesh routers, and so, they play a key role in the formation, configuration and maintenance of the WMN topology.

Many WMN routing protocols use similar strategies than the ad hoc routing protocols, which can be split in three distinct groups: proactive, reactive or hybrid.

In a proactive (or table-driven) routing protocol (e.g., Optimized Link State Routing - OLSR [34], or Destination-Sequenced Distance Vector - DSDV [35]), each node maintains updated information about the network topology in the routing tables through a constant exchange of routing information, which is transmitted by flooding. Since the routing information is disseminated periodically, this strategy causes a large overhead in the network. However, every node has information about the whole network, and so, a proactive approach allows the source nodes to obtain the required routing information to quickly establish a route to reach a specific destination. In a reactive (or on-demand) routing protocol (e.g., Ad hoc On-Demand Distance Vector - AODV [36], or Dynamic Source Routing - DSR [37]), the network nodes do not exchange routing information periodically, but instead they gather routing information on-demand, which leads to less overhead than a proactive strategy. However, the source node may have to wait for a considerable amount of time before a route to the destination is discovered, which can increase the network response time.

The WMN backbone significantly differs from the conventional ad hoc networks due to the lack of mobility and energy constraints of mesh routers. Moreover, most of the WMN traffic is typically directed to/from the WMN gateway. Due to these two factors, there have been proposed many extensions to the traditional proactive and reactive ad hoc routing protocols in order to deal with the WMN link-quality variations. For instance, SrcRR [38] is a variation of DSR that uses the expected link transmission time as a routing metric instead of the number of hops. AODV-Spanning Tree (AODV-ST) [39] adapts AODV by using spanning tree concepts. Here, the WMN gateway periodically requests routes to every WMN node in order to update its routing table based on the expected number of transmissions required to successfully transmit a packet, or on the expected transmission time to deliver the packet. Moreover, the WMN gateway is the root of the tree, and the communications that do not include the gateway use the original AODV protocol. In [40], it is presented a solution that enhances DSDV by using a back-tracing technique to minimize the disconnection time and packet loss rate when mobility occurs in WMNs.

A hybrid routing protocol (e.g., Hybrid Wireless Mesh Protocol - HWMP [41], or Mesh Routing Protocol - MRP [42]) combines the advantages of proactive and reactive strategies by adapting the routing scheme to the characteristics of the WMN, such as topology, size, mobility or traffic patterns. For instance, a hybrid routing strategy can be used in a WMN that is divided into zones or clusters. In this case, the routing protocol can employ a proactive strategy for intra-cluster communications and a reactive strategy for inter-cluster communications, since intra-cluster communications tend to occur more frequently than inter-cluster communications. However, this strategy is complex, since it depends on a trade-off between the proactive and reactive strategies.

Recently, due to the limitations of the ad hoc based routing protocols when applied to WMNs, other routing protocols have been proposed, such as opportunistic, multi-path or geographic routing protocols. Opportunistic routing protocols aim to exploit the unpredictable nature of the wireless medium by deferring the next hop selection after the packet has been transmitted (e.g., Opportunistic Multi-hop Routing for Wireless Networks - ExOR [43], or Simple Opportunistic Adaptive Routing - SOAR [44]). Although these protocols can lead to many advantages like the faster packet progress towards the destination, they require complex coordination between the transmitters regarding such progress. In multi-path routing protocols (e.g., Multi-path routing in wireless MESH networks - MMESH [45]), every mesh router stores multiple WMN paths to reach a specific destination, and it can then perform load-balancing by selecting one of the least loaded paths. Geographic routing protocols (e.g., [46]) deal with the availability of location information at mesh routers in order to forward packets towards the neighbor closest to the destination.

- **Routing Metrics**

Routing metrics, used by the routing protocols, usually consist of a set of measures that are mathematically combined to characterize the WMN link quality.

Since the exclusive use of the hop-count as routing metric in WMNs has proven to be inefficient [47], the measures to derive a WMN routing metric are generally obtained from the Physical (PHY) and Medium Access Control (MAC) layers in order to take into consideration the dynamic characteristics of the wireless medium, such as link capacity, link quality, channel diversity or interference.

In [47], it is provided a detailed explanation of the characteristics that a WMN routing metric should possess. Some criteria used to compare the WMN routing metrics are: *(i)* level of intra-flow, inter-flow or external interference; *(ii)* locality of the information; *(iii)* computational cost; *(iv)* ability to quickly respond to topology or load changes; *(v)* ability to preserve the order of the weights of two paths if they are appended or prefixed by a common third path; *(vi)* the metric of a link should not vary too fast over time; *(vii)* a metric should be able to consistently select routes with high throughput or low delay.

Different metrics have been proposed in the literature, each of them enhancing the last one: Expected Transmission Count (ETX) [48], Expected Transmission Time (ETT) and Weighted Cumulative Expected Transmission Time (WCETT) [49], Metric of Interference and Channel Switching (MIC) [50], Interference Aware Routing Metric (iAWARE) [51], Interference-Load Aware Routing Metric (ILA) [52]. A brief overview of these metrics is presented below.

ETX is defined as the expected number of transmissions a node requires to successfully transmit a packet to a neighbor (which is directly related to the packet loss rate), being the ETX of a route defined as the sum of the ETX of all the links along the route. Since ETX does not consider the data rate at which packets are transmitted over each WMN link, ETT improves ETX by capturing the data rate used by each WMN link. The ETT of link i is then defined by the product between ETX and the average time a single data packet requires to be transmitted, $ETT_i = ETX_i \times (S/B_i)$, where S is the packet size and B_i is the bandwidth of the link i. Both ETX and ETT metrics do not consider the intra-flow and inter-flow WMN interference and channel diversity. WCETT was proposed in order to enhance ETT, and it considers two components: *(i)* the sum of transmission times along a route; *(ii)* the number of channels available in a route to avoid intra-flow interference. Despite taking into account the bandwidth, error rate and channel diversity in a path, WCETT neither guarantees shortest paths nor avoids inter-flow interference, which are issues addressed by MIC. Using MIC, each node takes into account the number of interfering nodes in its neighborhood to estimate inter-flow interference. In addition, MIC guarantees the minimum-cost routes computation.

All of the presented metrics (ETX, ETT, WCETT and MIC) are based on average values computed on a time-window interval, and they may not follow the link-quality variations or may produce prohibitive control overhead. Instead, iAWARE uses Signal-to-Noise-Ratio (SNR) and Signal-to-Interference-and-Noise-Ratio (SINR) to continuously reproduce neighboring interference variations onto routing metrics, since it estimates the average time the medium is busy because of transmissions from each interfering neighbor. Finally, ILA is also able to find paths based on values of intra-flow and inter-flow interference, being easily adaptable to variations in transmission rates, loss ratio and congested areas.

In [53], the authors compare single-radio WMN metrics by means of testbed measurements. In [47], the authors discuss various single- and multi-radio WMN metrics, and compare performance results through simulations. In [33][54], it is presented an extensive qualitative comparison of the most relevant routing metrics for multi-radio WMNs.

2.2.2.2 Channel Assignment Schemes

Several works available in the literature have proposed novel protocols for WMN channel assignment in order to take advantage of the availability of multiple radios per mesh router, possibly working in distinct wireless channels. By assigning different non-interfering channels to possible interfering WMN links, these protocols can decrease the wireless interference and increase the overall spatial re-use. However, the design of multi-channel assignment protocols for WMNs raises important issues and challenges [55][56]. Channel assignment protocols can be broadly classified in static, dynamic and hybrid schemes.

Static channel assignment consists in a fixed assignment of channels to the radios of mesh routers, which remains unchanged over the course of the WMN operation. Such mechanisms (e.g., Multi-radio Unification Protocol - MUP [57], or Connected Low Interference Channel Assignment - CLICA [58]) are simpler and do not impose channel switching delays. However, they are less adaptive to changing conditions, such as interference or traffic demands.

In a dynamic channel assignment protocol, the channels assigned to each radio of each mesh router are dynamically changed based on several metrics, such as current interference, traffic demands or power allocation. In addition to the overhead of channel switching, mesh routers often require tighter coordination between them to avoid disconnections, deafness problems and multi-channel hidden terminal problem, which make the dynamic channel assignment protocols extremely complex. Despite its challenging design problems, a dynamic channel assignment protocol has potential to increase the WMN capacity. To solve the multi-channel hidden terminal problem, the Multi-channel MAC (MMAC) protocol [59] aims to coordinate all the mesh routers in order to allow the selection of different non-interfering channels for data communications. The Slotted Seeded Channel Hopping (SSCH) protocol [60] improves the MMAC protocol, by eliminating the need of the existence of a separate control channel to synchronize the mesh routers. Both MMAC and SSCH protocols require tight time synchronization between the mesh routers, which is decreased by the Extended Receiver Directed Transmission (xRDT) protocol, proposed in [61].

In hybrid channel assignment protocols, some of the radios of each mesh router have assigned fixed channels, while others switch their channels dynamically. These protocols benefit from their partial dynamic design, while inheriting the simplicity of static mechanisms. As shown in [62], in a hybrid channel assignment protocol, all mesh routers try to assign a different channel to their fixed radio. When a mesh router wishes to communicate, it switches its switchable radio to the channel of the fixed radio of the destination.

2.2.2.3 Cross-Layer Design

In order to improve the performance of several WMN protocols, cross-layer design has recently gained a special attention by the WMN research community [63].

The layered-protocol design carries several advantages, since protocols in one layer can be designed, enhanced or even replaced without any impact on other protocol layers. However, such a methodology does not provide a mechanism for the performance optimization among protocol layers, which can significantly compromise the overall WMN performance.

In the cross-layer design, parameters of other protocol layers are taken into account in order to improve the performance of a specific protocol layer. This way, information is exchanged among different layers, so that a better representation of the network resources and characteristics can be achieved. This is particularly true for WMNs because they demand for a scalable network performance, but are exposed to many challenging problems related to distinct protocol layers, such as heterogeneous QoS constraints, multi-hop wireless communications and variable link capacity. In addition, the cross-layer design provides a low-cost means of improving the overall performance, since it does not require additional hardware.

In the following, we survey several proposals that use a cross-layer design methodology to optimize two or more WMN management functions. Proposals in [64][65] present solutions to maximize the throughput of WMN links by adjusting the channel scheduling decisions, while minimizing the power consumption of WMN nodes. In such proposals, the scheduling algorithms take into consideration the interference relationships between the WMN links, which in turn is decided by the power assignments at WMN nodes, since the WMN nodes transmitting at high power level create higher interference WMN links. Several approaches (e.g., [66][67]) provide a joint solution to assign routing paths and schedule links in order to achieve a better overall throughput. Here, once the traffic demands are routed on specific routing paths, the scheduling algorithm tries to achieve a conflict-free schedule for links on these routing paths. In [68], it is presented a scheme that considers the estimated traffic demand and channel/radio information, in order to simultaneously find routing paths and assign the corresponding wireless channels until the estimated traffic requirement is satisfied. In [69], it is presented a metric that derives the expected number of transmissions required for delivering a packet over a path by taking into account the effect of the relative position of the link in which a packet is dropped, as well as the number of links and the link quality.

Despite the advantages of the cross-layer design, critical issues must be considered due to loss of protocol-layer abstraction, incompatibility with existing protocols, unforeseen impact on the future design of the network, and difficulty in maintenance and management [63].

2.2.2.4 Mobility Management Approaches

In a dynamic WMN environment, mesh clients can run services or access information anytime and anywhere, information flows can be redirected to different places, and mobile equipments can be moving while maintaining the connectivity.

This way, several mechanisms have been proposed to integrate mobility management in WMNs. In order to compare these mechanisms, some key requirements have to be considered: *(i)* Handoff (HO) management (may concern only on link-layer (L2) HO, only on network-layer (L3) HO, or even on both); *(ii)* location management (tracking and update of the location of mesh clients); *(iii)* addressing schemes (mobile equipments may need to acquire a Care-of-Address (CoA) in case of mobility); *(iv)* macro-mobility aspects. In the following, we describe the advantages and drawbacks of several solutions for WMN mobility management, which are summarized in Table 2.1.

- **Handoff Management**

When a mesh client moves from the range of one mesh router to that of another, HO is indispensable for connection continuity. Ideally, the HO should be completely transparent to mesh clients, and should be characterized by low packet loss, minimal latency, low signaling overhead, limited rate of failure, and similar QoS provided by the source and target systems.

Some proposals use gratuitous ARP (gARP) messages sent by mesh routers to provide an instantaneous link-layer HO [71][76][79][82][91]. The link quality between a client and a possible new mesh router is monitored (most of the times based on the SINR value of the messages received). When a possible mesh router believes that it has the best connectivity with the client, it sends a gARP message. Despite reducing the link-layer HO delay, gARP messages use ARP, which is a low-level protocol that cannot be properly secured.

Some approaches use a distributed MAC-to-IP association between the mesh routers and their associated clients to trigger the network-layer HO, reducing the overall HO delay [70][73][75][77][78][80][74]. This strategy is a trade-off between the risk of address conflict, which has to pay the global-state-routing update, and the requirement of a central location server or complex interaction between mesh routers to implement a complete-free strategy.

Several proposals assume that the link-layer HO follows the standard IEEE 802.11 procedures, and are only focused on the network-layer HO [95][85][97][93]. In [95], the IP header

Table 2.1: Mobility Management in Wireless Mesh Networks

	Addressing Scheme (Micro-Mobility Scenarios)	
	Addressing change inside the WMN	**No addressing change inside the WMN**
L2 Solution		iMesh [70] SMesh [71][72] [a] WAPL [73]
L2/L3 Solution	Graphs and Buffering [74]	LCMP [75] LIHP [76] Ant [77] MobiMESH [78] LCMIM [79] MEMO [80] [a] BASH [81] Integrating L2 and L3 Routing [82]
L3 Solution	AODV-PRD [83] [b] OLSR-FastSync [84] [b] Caching [85] MAPs Placement [86] [b] Macro Mesh Mobility [88] [a] Geo-Mobility Solution [90] Mobile Party [92] [b] M^3 [a] [94] Mobile Agent-Based Solution [96] Caching with a Circular Buffer [97]	MeshCluster [39] MeshDV [87] [a] MCIP [89] [b] QMesh [91] [a] MAMP [93] WMM [95]

[a] Includes Gateway Assignment to handle Macro-Mobility
[b] Requires Mobile IP (MIP) to handle Macro-Mobility

is used to update the routing paths. In [85][97], the intrinsic WMN characteristics, such as multi-hop data forwarding and wireless broadcast medium, are combined to allow the buffering of packets in neighboring mesh routers (in a promiscuous mode), which can reduce the HO packet loss. In [93], the post-HO routing optimization makes use of the WMN multi-path availability. Despite the potential of these approaches, their gains need to be carefully evaluated due to the costs introduced by: *(i)* the signaling required to update the location of mesh clients [95]; *(ii)* the high requirements of memory capacity of mesh routers [85][97]; *(iii)* the control and management of multi-paths [93].

Some network-layer HO solutions create a tunnel between the source and the target mesh router during the HO, maintaining the ongoing communications during the update of routing tables [75][77][87]. Others make use of make-before-break HO solutions in order to transfer the context of mesh clients to a new mesh router on behalf of mesh clients, which will have to be informed to execute the HO [81][83][96]. Finally, several approaches maintain a fixed tunnel between mesh gateways and mesh routers [76][94], which lead to the constant need of packet encapsulation/decapsulation.

Geo-mobility solutions [90] attempt to reduce the HO latency by assuming that mesh clients move over short distances between geographical neighboring nodes. However, they have the drawback of the global-state-routing update. There are also some proposals that try to solve some intrinsic problems of MIP-based solutions, through the enhancement of MIP-based micro-mobility protocols developed for wired and deterministic networks [86][89], or the adaptation of the schemes designed for Mobile Ad Hoc Networks (MANETs) [83][84].

- **Location Management**

It is important to minimize the signaling overhead, implementation and processing costs of the functions to locate and update the position of mesh clients in the WMN [98].

The basis of location management is the maintenance of location tables by mesh routers, where it is stored the: *(i)* addressing information about their current mesh clients (MAC-to-IP association [70][78]); *(ii)* current best paths for local communications; *(iii)* characteristics (source, destination and other important parameters) of vicinity communications (in a promiscuous mode [73]) or of communications that pass through them [75]; *(iv)* current mesh clients in neighboring mesh routers [77][93]. These location tables may exist in all mesh routers [79][70][95], only in the gateway [76][85][89], or in a central location server [77][82]. It is well-known that a central location server has to pay the location update cost, whereas a distributed solution can introduce more signaling overhead in the global-state-routing update. In [94], it is proposed a solution where the database of each mesh router only stores information of its attached mesh clients, and the databases of superior mesh routers store the information of the mesh clients that reside in subordinate mesh routers.

Concerning the mechanisms to distribute the location information in the WMN, there can be used different approaches. In [95], it is proposed a solution where the IP header (*Options* field) is used to carry location information. A flooding scheme to distribute the location information in the WMN is adopted in [70][39], but it can suffer from the corruption of the broadcast messages. In [71], it is presented a multicast solution to coordinate the mesh routers, in order to guarantee that at least one manages each mesh client at anytime, which can impose a high overhead in the management of multicast groups. In [74], dynamic neighboring graphs are built for context transference among mesh routers.

As can be observed, there are many approaches to locate, distribute and update the location information of mesh clients in the WMN. The envisioned scheme requires to have in mind that, in the one hand, the capacity of memory and processing of mesh routers are limited, and on the other hand, the overhead and signaling costs can be significantly increased.

- **Addressing Schemes**

In the traditional (MIP-based) solutions [74][83][84][85][86][88][94][96][97], it is required to acquire a new address (CoA) and register it with a mobility agent when a mesh client is moving (even in micro-mobility cases). These schemes can be improved through the separation between End-point-IDentifiers (EIDs) and addresses of mesh clients, by adopting geographical addresses [90] or tree-based address structures [84]. However, these non-standard-IP addressing mechanisms can introduce more overhead in the global-state-routing update.

Most of the other addressing schemes make use of a distributed MAC-to-IP association between the mesh routers and their associated mesh clients. They get an IP address from a MAC address by using a simple hash function, which can enable a client to keep the same IP address after HO. However, getting an IP address from a MAC address through hash functions may cause address conflict.

- **Macro-Mobility Aspects**

Besides the macro-mobility integration through MIP-based solutions [83][84][86][89][92], several solutions define novel algorithms to select the best gateway at the HO time.

In [91], it is proposed a gateway assignment algorithm that considers several parameters that incur QoS degradation and additional costs (e.g., network distance and congestion, or server loads), in order to achieve the best QoS while reducing the number of gateway HOs. In [72], macro-mobility is incorporated by assuming that new connections always use the closest gateway, while existing connections are forwarded through the wired infrastructure to their originally gateways. In [88], it is proposed an algorithm to determine when a mesh client is using another gateway, and such algorithm supports gateway assignment and multihoming. In [80], an effective gateway function has been implemented, but it is based on flooding mechanisms. Finally, in [94], the macro-mobility is integrated by placing the Foreign Agent/Home Agent (FA/HA) at the intersection of the gateways, and using different IP address pools for each gateway.

2.2.3 Initiatives & Testbeds

In the last few years, increasingly cheaper and more accessible WMN technology has lead to the development of a plethora of WMN testbeds with distinct application scenarios.

Examples of academic WMN testbeds include Massachusetts Institute of Technology (MIT) Roofnet [99], Mesh Purdue [100], MeshNet [101] and MeshCluster [39]. The MIT Roofnet [99] is deployed at MIT, being composed by 37 WMN nodes that cover an area of approximately $4km^2$. Each WMN node is equipped with a single radio and uses standard 802.11b hardware with an omnidirectional antenna. The MIT Roofnet uses source routing with the ETT link cost metric, and it also has its own algorithm to adaptively change the coding and transmission rate. Mesh Purdue [100] is another testbed with 32 WMN nodes equipped with multiple radio interfaces, using both omnidirectional and directional antennas. The routing algorithm is based on the OLSR protocol with the ETX cost metric. MeshNet [101] is a testbed composed by 30 WMN nodes, at the University of California at Santa Barbara, where all WMN nodes have a single radio interface and use the AODV routing protocol. MeshCluster architecture [39] extends the original architecture of MeshNet, so that WMN nodes may have multiple radio interfaces, and the routing protocol is modified to include more appropriate cost metrics, such as ETT or ETX.

Some WMN testbeds like Orbit [102] and Emulab [103] provide a flexible platform to allow WMN researchers to deploy and test their own architectures, ideas and protocols. Although testbed experimentations result in precise evaluations, they are often time-consuming, costly and inflexible. To overcome such issues, scaled-down and smaller transmission range versions of actual testbeds, such as ScaleMesh [104] and IvyNet [105], can also be used. Sometimes a combination of simulation, emulation and real world testbed experiments are used [106], or testbeds are deployed with advanced operating system virtualization techniques to improve the testbed control and management [107][108].

There is a diverse range of application scenarios for WMN deployment. In [109][110][111], users contribute with their own mesh routers to build community-oriented WMNs. TFA Rice Mesh [112], Heraklion Mesh [113] and Google-Meraki Mesh [114] are examples of WMN deployments to provide low-cost Internet access. With the recent awareness about using alternative sources of energy, [115] is a testbed where many of the WMN nodes can run with solar energy and rechargeable batteries. CalMesh [116] and RescueMesh [117] were designed for investigating communications in disaster scenarios and emergency situations, where a WMN can configure itself entirely automatically without any need for prior infrastructure. Other applications considered for WMNs include remote medical care [118], traffic control system [119], public services [120], integration with sensor monitoring systems [121].

Due to their attractive advantages and plethora of applications, many companies have started providing their own WMN solutions. Strix systems [122], Cisco systems [123], Meshdynamics [124] or BelAir [125] are some examples of commercial WMN vendors.

2.2.4 Future Directions

The simplicity, robustness, ease of setup/maintenance and self-organizing nature of WMNs have attracted an impressive amount of efforts on WMN research. However, there are still several issues that need to be addressed to increase the WMN performance [7][126], and some of these issues were already described along this Section.

At the physical layer, it is important to improve the transmission rate and the performance of the physical layer techniques due to the scarce wireless bandwidth availability. The use of directional antennas, in order to concentrate signal power in particular WMN directions, can also help in a better spectrum utilization and reduction of wireless interference.

At the MAC layer, novel MAC protocols need to be designed to deal with the exposed and hidden node problems in wireless multi-hop environments, and to provide QoS support. It is still required to reduce the complexity of the channel allocation and scheduling schemes, in order to take advantage of the full potential of multi-radio and multi-channel availability.

At the routing layer, the protocols and metrics should be enhanced to consider the WMN multi-path availability and to provide E2E QoS support. Moreover, both the physical, MAC and routing protocols need to consider the heterogeneity of context information available in WMN environments, and address security and scalability issues.

Finally, signaling mechanisms are required for: *(i)* network monitoring; *(ii)* energy management; *(iii)* multicast support; *(iv)* integration of WMNs with other networks, such as sensor networks, delay tolerant networks, vehicular networks or wired networks; *(v)* cooperation among the multiple wireless interfaces (and technologies) of a mesh router.

Tightly coupled with the aforementioned lines of WMN research, this Thesis aims to take advantage of the context information available in WMN environments in order to improve the WMN operation. Such context information can span from the requirements and features that characterize the WMN clients, terminals, routers, services, technologies or providers. This way, we aim to deploy an architecture that is able to personalize and enrich WMN communications and several control functionalities according to the preferences and expectations of mesh clients, and the requirements of their devices and services.

More specifically, we aim to select, configure and adapt the WMN nodes, interfaces and channels on-demand, in order to allow the automatic establishment and reconfiguration of WMN communications based on any type of context requirement, mobility and context change of mesh clients. This will imply the use of mechanisms for the context-aware WMN topology control and resource management, since a variety of communications can take place over the same WMN, each one requiring the use of distinct nodes or imposing different resource demands. Moreover, we envision to define a strategy to dynamically select the best WMN routes and paths according to the context and QoE feedbacks of mesh clients. This strategy can be interesting to leave behind the limitations of the shortest path, ad hoc and QoS-aware routing protocols when applied to WMNs. Please note that the context-aware wireless mesh networking thematic will be revisited in Section 2.4.

2.3 Network Virtualization

Network virtualization has been identified as one of the key technologies to overcome the gradual ossification problem faced by the existing Internet. It is also proposed to be an integrant part of a future Internet paradigm [4], since it enables the flexible and programmable coexistence of multiple, heterogeneous and isolated network architectures and protocols over the same physical network infrastructure. This Section presents the key features and applications of network virtualization, along with its most important techniques and research challenges.

2.3.1 Definition & Perspectives

Originally, virtualization appeared to stimulate the re-use of the physical hardware (e.g., servers and data-centers) in order to improve its overall utilization [127][128]. To achieve this, the resources of a physical machine (e.g., bandwidth, CPU-power, memory or storage space) are shared and multiplexed among distinct virtual machines. The control and management processes by which these virtual machines are dynamically created and terminated, or even cloned and moved to other physical machines, are usually performed by a Virtual Machine Monitor (VMM). A VMM provides an abstraction-layer between hardware and software, allowing multiple Operating System (OS) images to transparently use the same physical machine. From this perspective, virtualization enables a flexible hardware usage and software isolation, which reduces the complexity of the underlying network topology and the costs for network operators (CAPital EXpenditures - CAPEX, and OPerational EXpenditures - OPEX).

Recently, the research community is applying these concepts to virtualize distinct network elements (e.g., switches, routers or interfaces) in order to build logical networks (or Virtual Networks - VNs) on top of the same physical substrate, which can be provided by different infrastructure providers [129][130][131]. The first examples of network virtualization were the virtual isolation of different Local Area Networks (LANs) (VLAN concept) and the use of Virtual Private Networks (VPNs) to connect multiple distributed sites through tunnels over public networks. Network virtualization enables the creation of multiple, heterogeneous and isolated VNs in an easy programmable manner, since it introduces the required flexibility to share and multiplex the resources of network elements among distinct virtual instances, which can then be interconnected among them. Each VN can have its particular topology, can run its own services, or can have assigned specific addressing and forwarding mechanisms. This high level of transparency and isolation can be used by the research community to separate experimental from real traffic, in order to deploy and evaluate new network architectures and mechanisms for the future Internet. Beyond the physical resource sharing, network virtualization also allows the aggregation of a set of physical network resources that can then be managed by a single virtual instance, in order to optimize the traffic balancing or intrusion detection. In addition, network virtualization can provide a clean separation between service and infrastructure providers, which can stimulate new business models for operators and service providers. From this perspective, virtualization increases the resource manageability, promotes the network diversity and improves the communication reliability.

2.3.2 Network Virtualization Techniques

In this Section, we present the key features and open issues of several solutions available in the literature that address distinct topics related to network virtualization.

2.3.2.1 Virtual Switches

Several literature approaches enable the separation between the software and hardware functionalities of traditional switches.

In [132], it is presented OpenFlow, which is an open protocol to program a switch flow-table. In the one hand, commercial approaches provide the required speed, but vendors do not provide an open, programmable and virtual platform of their switches. On the other hand, open research platforms, such as a simple PC with several interfaces and operation systems, provide the flexibility, but not the necessary performance. With OpenFlow, researchers can run and control their experimental protocols using low cost programmable switches at line-rate and with high port-density, isolating experimental from real traffic. Thus, one of the major aims of OpenFlow is to encourage network-vendors to add OpenFlow to their switches for deployment in college campus backbones and wiring closets, not exposing their internal operation. An OpenFlow-based switch flow-table can be remotely controlled by an external controller (OpenFlow controller), which can communicate with the switch through a secure channel by using the OpenFlow protocol. In [133], it is presented a network virtualization layer, the FlowVisor, that enables the slicing of a physical switch into a set of isolated virtual switches. FlowVisor hosts multiple OpenFlow controllers, each one responsible to control the flow-table of a single virtual switch.

A very close approach to OpenFlow is NetFPGA [134], which proposes a simple methodology for the fast design and prototyping of network hardware. NetFPGA is a line-rate interface that directly translates the way of processing packets into a simple clean pipeline among hardware modules. This way, NetFPGA aims to ease the process of adding new modules in a pre-built NetFPGA design, which can make the hardware design more re-usable for professors and researchers. However, there are several open issues, such as how to assure line-rate performance when there are distinct NetFPGA modules working in parallel.

2.3.2.2 Virtual Routers

There are several approaches that deal with the design of flexible and configurable routers, in order to leave behind the conventional router architectures.

Click modular router [135] is an open and extensible router framework, in which the router building blocks are packet processing modules, called elements. These elements are able to implement simple functions, such as packet classification, queuing, dropping policies, scheduling and interfaces with network devices, and they are connected into a graph to build a router configuration. Packets move from element to element along the graph edges, called connections. The Click modular router includes two specific features that add power to this simple architecture: *(i)* pull processing, which models packet motion driven by transmitting interfaces and makes packet schedulers easy to compose; *(ii)* flow-based router context, which helps an element to locate other interesting elements. Moreover, Click allows the building of routers from small and modular elements, while maintaining a good forwarding performance for PC hardware. The user can write new elements or compose the existing ones in new ways, in order to define a new router configuration. However, Click does not consider distributed heterogeneous environments.

In [136], it is proposed an extensible virtual router architecture, called VERA. VERA includes a router abstraction-layer, which must be rich enough to support the requirements for IPv4 routers, along with new forwarding functions. VERA also includes a hardware abstraction-layer which supports the range of hardware of interest, exposing only enough hardware details to allow improved router implementations. In the middle of these two abstraction-layers, it is included a distributed router operating system, which ties together the router abstraction and the hardware abstraction in a clean manner.

In [137], it is presented XORP. XORP uses the Click modular router for packet forwarding, and provides a flexible way to control the router forwarding plane. In such approach, each routing protocol and forwarding management function is implemented in a separate process, adopting a single-threaded event-driven programming model. In such model, events are generated by timers, and file descriptors and call-backs are dispatched whenever an event occurs.

The Virtual Router Project [138] also explores the idea of deploying a re-programmable router forwarding plane in order to run multiple virtual forwarding paths in parallel. As concluded by the Virtual Router Project, the mechanisms for the control and maintenance of multiple virtual routers, each one characterized by its specific forwarding path, still have several open issues: *(i)* the correct classification of the packets that belong to each virtual router; *(ii)* the queuing complexity introduced by novel scheduling mechanisms to achieve higher fairness and isolation; *(iii)* the large memory requirement to maintain separated forwarding tables, and to merge them into a single table residing in the physical forwarding plane. By evaluating the performance of virtual routers, the Virtual Router Project also concluded that modern commodity x86 hardware (with multi-core CPUs, or network interface cards with hardware multi-queuing) may constitute a viable platform to support high-performing and cost-effective virtual routers. However, the global scheduling of memory access and the dynamic allocation of forwarding paths to the available cores, still need to be solved.

In [121], it is presented VROOM, which is a solution that virtualizes routers and allows their live migration from one physical node to another one. The live migration of virtual routers between two different physical nodes (A→B) includes several steps: *(1)* setup of a tunnel to redirect routing messages to B; *(2)* transference of a memory image snapshot of A, which causes a short handover outage (at the end of this step, B starts to remotely control the control plane of A, but the data plane of A is still active); *(3)* migration of data flows, with the possibility of the simultaneous existence of two parallel data planes; *(4)* remotion of the data plane of A, and redirection of tunnels. VROOM can have different use cases, such as the flexible device and configuration maintenance, and the energy-saving and load-balancing by turning on/off virtual routers.

2.3.2.3 I/O Virtualization

The concept of virtualization of Input/Output (I/O) devices, which is commonly associated to the virtualization of network interfaces, has a higher impact on the virtualization of switches and routers than on the virtualization of servers or data-centers [139][140][141].

There are several possibilities to share a physical I/O device among multiple virtual instances: *(i)* emulate the physical I/O device in every virtual instance, which has considerable overhead; *(ii)* apply para-virtualization, where a special virtual instance has to control and manage all I/O devices, and connect them to the other virtual instances through a virtual bus; *(iii)* full-virtualization, where virtual instances directly access to the physical I/O devices.

The physical I/O devices should be prepared to be accessed by multiple virtual instances. However, it is required that a specific VMM (or other abstraction-layer) to have good I/O performance and dedicated control of the physical I/O devices. Moreover, there are other open issues of I/O virtualization that may increase the CPU overhead and decrease the network throughput, such as: *(i)* address, memory and interruption transference, translation and re-mapping; *(ii)* traffic scheduling and multiplexing; *(iii)* hot plugging and swapping; *(iv)* congestion control; etc.

2.3.2.4 Virtual Networks

There are several approaches that deal with the building-up of several VNs over the same physical network infrastructure. These approaches split physical routers, switches and interfaces into distinct virtual instances, which are then interconnected to build isolated VNs. Each of these VNs can have its particular topology, can run its own services, or can have assigned specific addressing and forwarding mechanisms. In the following, we describe some multi-VN architectures developed in the scope of several research projects [4].

Among other projects, PlanetLab [142] allows the slicing of a large scale overlay testbed in different VNs, in order to design, evaluate and deploy geographically distributed network services. Each physical node has a VMM for the scheduling and allocation of its resources among the distinct slices that it supports. To obtain a virtual slice, an user first contacts a resource broker, then goes through an admission control process in each of the nodes assigned by the broker, and finally launches its service in the resulting slice.

VINI [143] is an extension of PlanetLab that uses its infrastructure. It can be considered as a specific instantiation of an overlay network that runs software routers, and allows multiple VNs to exist in parallel. The incorporation of programmability in PlanetLab allows users to evaluate their own protocols and services in real networking environments.

GENI [144] is a US initiative with many similarities to PlanetLab. It provides an open and large scale testbed to evaluate new architectures by creating customized VNs, carrying real traffic on behalf of end-users, and connecting to the existing Internet to reach external sites. Similar to PlanetLab, it includes sliceability to share and isolate the resources of the physical network infrastructure, but this time in a space- and time-manner. Moreover, it provides a flexible platform for user access, in order to test and evaluate its own proposals.

CABO [131] is a design of the next generation Internet that aims to de-couple the infrastructure from the service providers. In CABO, infrastructure providers are expected to lease their infrastructure entities over which service providers could deploy their own specific protocols, and run their own network services optimized to specific service parameters, such as low latency or real-time support. The infrastructure providers may virtualize their physical network infrastructure, and thus, allow the isolated coexistence of multiple service providers.

4WARD [145] enables the co-existence of multiple networks on common platforms through carrier-grade virtualization of network resources, embracing a full range of technologies from fiber backbones to wireless and sensor networks. It enhances the utility of networks by making them self-managing, increasing their robustness by leveraging diversity.

In a multi-VN environment, multiple infrastructure providers can offer a set of virtual components to build distinct VNs. Further, there can be many possible ways to map a given VN link in the nodes and links of a physical network infrastructure, since such VN link can span multiple physical hops. In [146][147], several approaches are proposed to solve the VN mapping problem by considering distinct resource constraints of a VN request (e.g., CPU utilization and location of VN nodes, or bandwidth capacity of VN links).

2.3.2.5 Wireless Network Virtualization

The broadcast nature of the wireless medium, which leads to radio interference among the various mobile nodes that share the same wireless spectrum, imposes some unique and interesting challenges in the virtualization of wireless networks that are not observed in its wired counterpart [148]. In the literature, there are several approaches to slice wireless network resources in order to allow multiple concurrent and isolated experiments in wireless testbeds. However, it is still unsolved the issues of how to dynamically share and isolate the wireless medium in frequency, time, space or code dimensions, and its impact on different types of applications. The mechanisms for applying wireless network virtualization based on Frequency Division Multiple Access (FDMA) or Time Division Multiple Access (TDMA) allow the possibility to slice different wireless experiments along the frequency or time dimensions, respectively. However, the FDMA- and TDMA-based mechanisms present high channel switching times taken by the wireless nodes to switch between channel frequencies or between non-overlapping channel time-slots. Moreover, many experiments in wireless environments are focused on the definition and evaluation of novel PHY and MAC protocols, and so, the sharing of wireless nodes among multiple virtual experiments becomes impossible. In addition, some emerging applications have very strict latency requirements, which are lower than the channel switching times (e.g., vehicular networking). In the following, we summarize several works available in the literature that allow the virtualization of IEEE 802.11 Access Points (APs), and enable the slicing of a IEEE 802.11 network infrastructure into a set of virtual networks.

In [149], it is presented WiSwitcher, which is a wireless client able to connect to multiple APs. WiSwitcher reduces the cost of switching between APs down to the hardware switching time, increases the stability of the percentage of time assigned by schedulers, achieves high aggregated throughput over the connecting APs and seamlessly transmits data traffic under controlled scenarios (a very similar approach to WiSwitcher is MadWiFi [150]). In [151], it is proposed the SplitAP architecture that allows a network operator to deploy a shared physical AP, which is capable to run different algorithms that control the uplink airtime across distinct client groups. In [152], it is presented a hybrid wireless virtualization approach, named Virtual WiFi, which combines elements of software and hardware for virtualization support. Virtual WiFi aims to: *(i)* expose full wireless functionalities inside virtual machines; *(ii)* provide link level isolation across different virtual machines; *(iii)* achieve an appropriate performance/complexity trade-off with architectural choices specifically suitable for wireless interface virtualization. MultiNet [153] is a driver-based approach to facilitate simultaneous connections to multiple VNs using a single wireless card, which continuously switches the card across multiple VNs. In [154], each wireless device is equipped with at least two physical wireless cards: one of them is switched to a fixed channel for receiving traffic, while the second one is switched to any of the remaining channels for sending, which allows the wireless device to virtually connect to multiple networks simultaneously.

Concerning the slicing of a wireless infrastructure in a set of isolated VNs, ORBIT [155] is a radio grid-based open testbed that is intended to facilitate a broad range of experimental research on next generation wireless network protocols and applications. In the ORBIT testbed, which is composed by 400 wireless nodes with multiple wireless interfaces per node, researchers can remotely: *(i)* download their own code; *(ii)* run experiments in their own slices; *(iii)* monitor the behavior of the network; *(iv)* collect results. A similar approach to ORBIT is presented in [156] in order to allow multiple and isolated experiments to coexist on a shared wireless experimental facility. In [157], it is proposed a radio resource sharing framework, which enables different virtual radio networks to operate on top of a common physical infrastructure and share the same radio resources without interfering with each other.

In the scope of WMNs, which is the network infrastructure addressed along this Thesis, [158] presents the key mechanisms to design a system that virtualizes a WMN into a different number of isolated VNs, but without providing any type of system instantiation or evaluation. In [159], it is proposed a WMN architecture, called OverMesh, that aims to slice a WMN into a distinct set of overlay networks in order to deploy distributed services across WMN nodes. Concerning the performance improvement of WMN communications through network virtualization, [160] describes a joint approach to optimize streaming rate allocation of traffic flows and power consumption of links in multicast overlays. In [161], a wireless ring over a regular WMN is proposed in order to carry high bandwidth data. Moreover, in [162], the benefits of applying wireless interface virtualization are evaluated in the context of energy-efficiency and seamless mobility in WMNs.

The related wireless virtualization approaches allow the instantiation of different VNs in the same physical WMN, and provide basic methods for wireless resource sharing among these VNs. They also show that wireless virtualization can indeed be implemented in real environments. However, these approaches leave behind the mechanisms to dynamically create, setup, adapt and remove the VNs on-demand, in dynamic WMN environments, which is an aspect that this Thesis aims to go beyond the works available in the literature.

2.3.3 Future Directions

Beyond the description of the potential of network virtualization techniques, this Section has already focused on several aspects of such techniques that need further research. First, virtualized servers have to access and switch between the physical bridging hardware in order to redirect processes and swap memory among the distinct virtual machines, which lead to distinct security and compliance problems. Second, in the virtualization of routers and switches, it is required to develop more sophisticated methods to schedule their interfaces and other resources on-demand, which can increase the scope and complexity of management overhead. Third, the application of virtualization in wireless networks has to deal with the lack of wireless resources, since the control protocols of different layers are fighting to access them, and data flows are competing for increased throughput; hereby, wireless interference is a major problem and has to be decreased, while throughput has to be increased on the wireless links. All of these aspects impose the need to effectively understand what are the real benefits of applying network virtualization in a specific scenario or environment, and what are the network elements that need to be virtualized. This action is important to reduce the overhead and cost, and increase the security and performance of network virtualization.

More challenging issues related to the provision, control and management of virtual resources and networks have been raised. A flexible control and management of a multi-VN environment, where each VN can be specialized to run distinct services or applications, requires the definition of intelligent mechanisms to maximize the number of coexistent VNs, without degrading the performance of any of them [163]. These mechanisms can span from the: *(i)* dynamic monitoring, description, discovery, provisioning, polling and failure handling of (virtual and physical) resources in order to quickly map or adapt the VN elements to the physical ones; *(ii)* admission control of VN requests in order to assure that the physical capacity of the underlying physical infrastructure is not exceeded; *(iii)* sharing of physical and virtual resources under new constraints (e.g., energy or security) in order to optimize the performance that a VN service or application expects [164], or to improve the overall energy-efficiency by idling or turning-off virtual resources [165].

Network virtualization may help in the traffic management, enabling multi-homing and multi-path optimization over different VNs, virtual paths and infrastructure providers. This can increase the network resilience, load-balancing and fault tolerance, and the reliability and availability of communications [166]. However, it is still required to define standard interfaces and signaling mechanisms to allow the effective collaboration among distinct infrastructure providers, or the secure inter-operability among VNs with comparable functions [167][168].

Fully aligned with the current topics of research in the network virtualization area, this Thesis aims to: *(i)* define a multi-VN architecture to be applied to WMNs, where each VN is specialized to meet the preferences and QoE expectations of a subset of mesh clients, along with the QoS requirements of their services; *(ii)* provide intelligent and context-aware mechanisms to dynamically create, setup, adapt and remove the VNs on-demand. This way, this Thesis also aims to stimulate the extension of the virtual platforms of network operators up to the WMN environments.

2.4 Context & Context-Awareness

Context, as it was defined by [2], is any objective or subjective type of information that can be used to characterize the situation of network elements, such as users, devices, services, switches, routers or operators. Recently, context information has been used by the research community as a mean to personalize or optimize several network operations or processes, which can then be dynamically adapted to react to context change.

In this Section, we first summarize several context-aware approaches for WMNs available in the literature, and describe how we aim to go beyond these approaches. As the QoE perceived by an user is an important subjective component of the user context, we also give a special emphasis to QoE-aware wireless mesh networking, and state what is the main goal of this Thesis with regard to such thematic.

2.4.1 Context Definition & Modeling

There are several context features with high importance in networking environments that, if correctly considered or integrated in the definition of the network mechanisms and protocols, can enhance the interaction among users or even improve the performance of their devices and services. These features can span from: *(i)* location, mobility patterns, QoE expectations, price and privacy preferences, profile and personal information of users; *(ii)* processing power, battery and memory capabilities of the user devices; *(iii)* QoS, availability and reliability required by the user services; *(iv)* CPU-power, memory, storage space of network nodes; *(v)* bandwidth, interface/channel availability, quality and interference conditions of network links; *(vi)* trust and security policies of Internet Service Providers (ISPs); *(vii)* lightning, noise levels, time and temperature characterizing the networking environment.

The dynamic adaptation of network operations and processes to meet changing context requirements is a undeniable goal of the research community towards the future Internet. The European projects C-Cast [169] and SENSEI [170] are two major recent projects in this field that demonstrate the timeliness of context-aware networking. While the goal of C-Cast was to provide mobile multimedia multicasting in an E2E context-aware communication framework deployed over a sensor-enriched mobile environment, SENSEI focuses on the building of context-aware services over a mixed mobile and fixed environment.

One of the key mechanisms in context-aware networking is the modeling of context information in a way that simplifies the processes by which such information is stored, transported and used in the network [171][172]. The context modeling mechanisms should be extensible and flexible enough to accommodate existing and future aspects of context information. These mechanisms can span from key-value, graphical, object-oriented, logic-based or ontology-based models to the simple use of context levels. Moreover, they should be able to directly map context information in proper network rules or policies, and to associate data with a specific set of context features. Despite making use of a basic context model to design a context-aware architecture for WMNs, the work of this Thesis will not focus on the open issues and challenges of the context modeling techniques available in the literature.

2.4.2 Context-Aware Wireless Mesh Networking

The use of context information to enhance the WMN performance is manifold. In this Section, we present several WMN mechanisms and protocols that dynamically adapt their functionalities by being aware of information related to security, energy, user mobility, access technologies, user preferences or service requirements. At the end, this Section describes how this Thesis aims to go beyond these context-aware WMN approaches.

2.4.2.1 Security Issues

Security is an important context parameter that has to be considered in any network paradigm, and it can encompass aspects that span from data confidentiality and integrity, access control, secure routing, intrusion detection, secure key exchange or authentication. In the scope of WMNs, the security vulnerabilities have become crucial problems due to the wireless and multi-hop nature of this type of networks [173], which have led to a large amount of research to study the impact of distinct security constraints and considerations on WMNs.

In [174], it is proposed WMNSec that is an adaptation of the IEEE 802.11i security standard, specifically targeted at WMNs. WMNSec reduces the authentication time by up to a factor of 3, when compared to the IEEE 802.11i standard. [175] presents a light-weight multi-hop authentication scheme that could be used as an upper-layer mechanism on top of WMNSec or 802.11i, in order to authenticate mobile clients without using a shared secret. In addition, [176] describes an approach to be applied in a scenario where several WMN nodes are operated by different parties, posing another set of security issues like billing, which is a topic addressed in [177]. In [178], it is proposed a WMN security infrastructure that incorporates a novel client-router authentication and key agreement protocol, which can be applied in both communication and computation, being resistant to some common attacks with low cost of maintenance.

Several secure routing protocols have been proposed for WMNs in order to identify and discard false routing information, thus eliminating the probability of attacks. For instance, Security Enhanced AODV (SEAODV) [179] improves the traditional AODV routing protocol for security in WMNs, and it uses a key pre-distribution scheme to establish keys in the WMN. Each WMN node possesses two types of keys, Pairwise Transient Key (PTK) and Group Transient Key (GTK), which authenticate unicast and broadcast routing messages respectively. A unique PTK is shared by each pair of nodes, while GTK is shared secretly between the node and all its one-hop neighbors. Depending on the type of the routing message, MAC is computed using either PTK or GTK, and it is attached as an extension of the original AODV routing message to guarantee the message authenticity and integrity in a hop-by-hop fashion. Moreover, [180] designed and evaluated a secure infrastructure-based WMN routing protocol that uses distance vector routing. In this protocol, each node keeps the information of its two-hop neighborhood, and it is introduced a new routing metric to select the best shortest and secure path by using a two-hop passive acknowledgment scheme.

2.4.2.2 Energy Constraints

WMNs are very flexible since they can easily change their structures, configurations and resource distributions, while maintaining the required quality for users and services. For instance: *(i)* the WMN topology allows connectivity options to be selected as demanded; *(ii)* the presence of multiple wireless interfaces allows to connect via different wireless technologies; *(iii)* the wireless medium itself allows to dynamically adapt data rates or transmission ranges. Since the energy consumption of communication systems is becoming a fundamental issue, and the wireless environments are largely responsible for the increase in consumption, the WMN flexibility supports that energy-efficiency in WMNs can be significantly improved [181]. In the following, we present several proposals that monitor, measure or estimate the energy consumption of WMN nodes in order to reduce the WMN energy expenditures.

In [182], it is proposed the concept of solar-powered WMNs with sleep management and connection admission control, which aims to optimize the selection of the sleep/wake-up parameters or the connection admission control threshold in order to meet a set of desired QoS constraints. In [183], it is presented an analytical model to study the problem of energy wasting in large scale and high density WLANs detected in [184]. Such model is used to study two simple on-demand policies that, based on instantaneous WLAN parameters, can select the appropriate number of APs to activate, thus avoiding the energy wasting on under-utilized APs. The model of [185] aims to minimize the power consumption of a wireless access network by selecting the optimum network configuration in terms of instantaneous power consumption, while ensuring enough capacity and coverage for active users in a specific service area. The work presented in [186] introduces a context-aware energy management system to optimize the operation of self-sufficient WMN nodes. [186] proposes several context-aware rules to reduce the energy consumption of WMN nodes and to balance the overall energy budget of the WMN, by assuming that WMN nodes can be equipped with technologies for energy generation from renewable energy sources. [187] focuses on the energy-aware path selection for IEEE 802.11s WMNs. To this purpose, [187] proposes a path selection algorithm that provides a balanced energy spending among WMN nodes, which then reduces the amount of workload that is usually given to other nodes, and so, maximizes the WMN lifetime. Finally, [188] presents a novel approach for the dynamic WMN energy management by jointly optimizing the issues of wireless coverage, for the wireless access segment, and routing, for the wireless backhaul network.

2.4.2.3 Mobility Management

Due to the proliferation of the mobile broadband access, users can now run a variety of services anytime and anywhere, and user terminals can be moving while maintaining the Internet connectivity.

The movement patterns of users can be characterized by different features, such as speed, direction, frequency of direction changes. Moreover, they can be used to optimize several network functionalities, e.g., connectivity, handover or topology management. In the following, we summarize a set of approaches that are aware of mobility of mesh clients to enhance several WMN processes [98].

In [189], an adaptive routing algorithm is proposed to react to the intensity of mobility within the WMN, which is assumed to be high if WMN link breakages are detected. Depending on the mobility assumed, either a reactive or a proactive routing algorithm can be applied.

The current position of mesh clients is taken into account by the channel assignment and router selection algorithm proposed in [190]. In this work, the information of the position of mesh clients is used to configure the routing order selection, so that users in low congested areas are routed first, which can lead to a higher WMN throughput.

In [191], the authors present a clustering scheme that aims to improve the radio resource utilization in WMNs by considering both the mobility of users and the interference effect among neighboring WMN links. The WMN is divided into a set of virtual clusters covering all the nodes in the WMN. In each cluster, one WMN node operates as an intermediate node between the gateway and the APs inside the cluster, and manages the mobility of local users inside the cluster. Following this approach, the major part of the exchanged signaling messages due to user mobility is restricted to a local area (i.e., the cluster), which reduces the resource utilization costs, and thus, improves the WMN performance.

In [192], the authors describe a localized approach to mobility management and optimized post-handover routing in WMNs. In such approach, information of user location is recorded on geographically close nodes, and a multi-path packet forwarding strategy is used to maintain active connections during a handover. In order to support seamless handover in WMNs, [193] presents a mobility-aware extension to proactive routing protocols that makes use of neighbor client tables and a set of control messages to avoid packet loss during user mobility.

2.4.2.4 Access Technology & User Preferences

The recent advances in wireless devices allow multiple interfaces and technologies to be supported by a single device. Mobile users can then be simultaneously connected to diverse wireless access technologies. This leads to the possible definition of new mechanisms to dynamically select the best available technology that not only satisfies the user service requirements, but also the particular connectivity preferences of users with respect to mobility, price or security, following the "always best connected" paradigm [6]. The dynamic selection and adaptation of the user access technology is also a very important topic in the scope of WMNs, since they form a flexible wireless access infrastructure for a large amount of users.

In [194], it is proposed a solution that uses non-compensatory and compensatory multi-attribute decision to select the optimal access network to assign to an user. By using a non-compensatory algorithm, the access networks that are not suited to meet the user requirements are first removed from the candidate list. Then, a compensatory algorithm is used to select the access network that has the shortest Euclidean distance from the best solution, and the longest distance from the worst solution.

Similarly, [195] suggests an architecture that enables users to select the access networks that best suit their needs. In such work, and beyond the definition of a framework for information collection and access discovery, users can select their personalized best access networks by changing weight factors and constraints in a single objective optimization problem.

The work presented in [196] applies mathematical and computational techniques to model the access selection problem. The authors first use an analytic hierarchy process to decide the relative weights of the criteria set according to the user preferences and service requirements. Then, they adopt grey relational analysis to rank the candidate access networks with faster and simpler implementation than the analytic hierarchy process.

The work in [197] models the multi-constraint dynamic access selection problem as a variant of the bin packing problem, which is then solved by using a set of approximation algorithms. This work aims to map all traffic flows in the available access networks in order to both satisfy the user preferences, maximize the number of traffic flows admitted in the network, and minimize the power consumption cost, while satisfying QoS needs.

Finally, [198] and [199] are solutions to aid the user-centric network selection and decision process in order to optimize the handover across heterogeneous access networks. In the one hand, [198] makes use of a satisfaction degree function to evaluate, according to an user predefined criteria, the available access networks and select the best one(s) according to such criteria. On the other hand, [199] employs a novel multiplicative aggregative utility approach to best evaluate the candidate access networks.

2.4.2.5 Service Requirements

Over the past few years, a huge plethora of multimedia real-time services with high demanding requirements in terms of e.g., quality performance, availability or reliability, have been developed using Internet protocols. Since the Internet was never designed to support the wide range of QoS requirements of these services, several solutions have been proposed to enhance the network control and management processes in order to meet such QoS requirements, while improving the network resource usage. These solutions span from: *(i)* link adaptation in the physical layer; *(ii)* service differentiation in the MAC layer; *(iii)* route adaptation in the network layer; *(iv)* admission control and bandwidth reservation; etc. In the following, we summarize several QoS-aware approaches for WMNs.

In [200], it is addressed the interference-aware topology control and QoS routing in IEEE 802.11-based multi-channel WMNs with dynamic traffic. First, this work proposes a novel definition of co-channel interference to precisely capture the influence of the interference in a multi-channel WMN environment, which is then used by a channel assignment algorithm. Second, it is presented a polynomial time optimal algorithm to solve the bandwidth-aware routing problem, which seeks routes for QoS connection requests with bandwidth requirements on a given network topology induced by the channel assignment algorithm.

The work presented in [201] studies QoS routing in WMNs with cognitive radios. The channel allocation and scheduling scheme is performed by considering the impact of interference and channel heterogeneity. Then, it is presented a distributed routing protocol and resource allocation scheme to satisfy the E2E bandwidth requirements of WMN flows.

In [202], it is proposed a cooperative cross-layer method to jointly design the QoS routing and rate adaptation in IEEE 802.11s-based WMNs. More specifically, this work incorporates the transmission delay, traffic load and bandwidth in order to improve the metric of the hybrid wireless mesh protocol, which is commonly used for best effort traffic. Moreover, the rate adaptation is considered in the routing discovery, being proposed a new link-layer sensing technique to acquire the packet delivery ratio of current links with low overhead.

An architecture which adapts to current WMN conditions for the support of real-time communications is proposed in [203]. Here, performance metrics, such as the path availability, bandwidth and delay, as well as outages based on network monitoring and statistical evaluation, are used to assess the robustness and perform the selection of the best available WMN path for a real-time communication.

In [204], it is presented a bandwidth-constrained routing protocol for multi-radio multi-channel WMNs based on the IEEE 802.11 Distributed Coordination Function (DCF) MAC protocol. First, a threshold-triggered bandwidth estimation scheme is proposed for each WMN node to estimate the free-to-use bandwidth on each channel. Then, a new call admission control algorithm predicts the residual bandwidth of a WMN path by considering the free-to-use bandwidth at each WMN node, and the inter- and intra-flow interference.

In [205], a new QoS-aware routing protocol, named QUORUM, is proposed. QUORUM integrates a novel mechanism that predicts the E2E delay of a flow with good accuracy, which can then be integrated into flow setup to satisfy QoS requirements. Further, it defines a robustness metric for link quality and demonstrates its utility in route selection.

[206] and [207] are solutions that aim to assist the service-oriented wireless mesh networking. In the one hand, [206] proposes a novel protocol for the selection of a set of WMN nodes that facilitate the service matching between the producer nodes' advertisements and the consumer nodes' subscriptions. On the other hand, [207] introduces a novel algorithm to select the best WMN topology that meets a predefined set of QoS requirements.

Finally, the European project CARMEN [208][209] has defined an architecture capable of delivering carrier-grade triple-play services over heterogeneous WMNs. The proposed architecture is intended to meet the quality and reliability requirements of service providers, while reducing their installation and operational costs.

2.4.2.6 Multiple Context Integration

Along this Section, we have summarized several proposals that are aware of specific context information available in WMN environments in order to improve or enhance distinct WMN control and management mechanisms, making them adaptive to react upon context changes. Although the aforementioned proposals have well-known merits, most of them are only focused on a single context parameter, which can be quite misleading when studying the overall WMN functionality or performance. To meet this drawback, several solutions have been proposed by considering a more broad set of context parameters. For instance, [210] proposes a model that considers user mobility, traffic patterns, wireless technologies and QoS requirements, which can be used by the wireless networking community in the simulation of wireless data networks. [211] defines a model to measure the interaction between delay of QoS and security in different application scenarios, where the security level is related to three main features, i.e., authentication, data integrity and confidentiality. In [212], it is presented a context-aware mobility management mechanism that provides fast handover with guaranteed QoS for real-time multimedia services and minimal energy consumption in WLANs. [213] first classifies the energy saving mechanisms for security protocols in wireless networks, and then applies such mechanisms to the existing security protocols in order to reduce their energy consumption. [214] defines a middleware for context-aware routing in WMNs, where routing metrics, such as hop count, retransmission count or link interference, are related to context data, i.e., security, traffic priority or mobility, by means of manually configured rules.

Despite the potential of the presented context-aware WMN approaches, most of them are narrowly defined to improve specific WMN control and management mechanisms, without providing a sufficiently rich descriptiveness to personalize and enrich the connectivity of WMN clients up to a high degree. Therefore, the presented Thesis aims to go beyond the related work by defining a generic model to characterize the user context, which is then used to build user-centric personalized WMNs. While in the related work, the WMN resources and operations are selected according to general network characteristics, in our approach, the user preferences and expectations, along with the requirements of their devices and services, are modeled as multi-variate context data and automatically mapped to corresponding WMN features. These features can include the proper selection and configuration of WMN paths, routes, connections, interfaces, channels or control operations on-demand, in order to be dynamically adapted to user context and its change.

2.4.3 QoE & QoE-Awareness

Since this Thesis aims to define a WMN architecture that is driven by the user context requirements, it is important to observe and measure how well the architecture is satisfying these requirements, and possibly adapt its behavior to better meet such requirements.

Therefore, the concept of QoE, which was defined by [215] as "*The overall acceptability of an application or service, as perceived subjectively by the end-user*", has to be considered in our approach to capture how an user perceives the usability of a service when in use. Moreover, it is important to perform QoE-aware wireless mesh networking, that is, several network selection or management processes need to be adapted according to the QoE feedbacks of users. This can increase the overall user perceived QoE and lead to a fairly distribution of WMN resources among users.

This way, this Section first introduces QoE and summarizes several approaches available in the literature to measure QoE. Then, it presents selected proposals related to QoE-aware wireless mesh networking, stating how we aim to go beyond such approaches.

2.4.3.1 QoE Definition & Measurement

Recently, the proliferated consumption of rich media content, the common usage of personalized services and the increased tendency to request any multimedia content at any time at any device, have led to a strong growth in network traffic and in the market competition among service providers. Moreover, the users' awareness for the quality of the requested services has become more and more important; in this context, the multimedia content, especially video, directly reflects the network quality in the viewable content [216].

Due to these factors, we have observed an important paradigm shift in the service quality research, since the research on QoS, which has been one of the dominating research topics in the last decade, has been replaced by the research on QoE.

QoS was defined by [217] as "*The ability of the network to provide a service at an assured service level*". QoS is related to objective metrics, that is, QoS is captured by purely technical network parameters in order to detect any network problem that affects the service delivery, such as high delays or losses and low throughput [218].

In contrast, users are usually not bothered about technical parameters, but instead, they observe the service required, and then derive their subjective quality measure for the service by taking into consideration the audio and visual outputs of the service. Thus, the users' subjective perception of the service quality is not reflected in QoS, but it is captured in terms of QoE, which is a measurement of how well a service offering satisfies the user expectations in terms of usability, accessibility, retainability and integrity of the service [217]. QoE includes the complete E2E system effects (client, terminal, network, etc.) and it can be influenced by the context of the user (e.g., predispositions, expectations, needs, motivation or mood), the characteristics of the designed system (e.g., complexity, purpose, usability, functionality or relevance) and the environment within which the service is experienced (e.g., organizational/social setting, meaningfulness of the activity or voluntariness of use).

Since there is no standard available to measure and express QoE in a general context, the development of novel tools and techniques for the accurate measurement and assessment of QoE has attracted significant interest from the academia, industry and standardization bodies. In the following, we present several proposals to measure QoE in distinct environments.

The use of Mean Opinion Score (MOS) [219] is one of the most common methods to evaluate QoE. In such approach, the subjective user perception of the service quality is expressed by a number which value ranges from 1 (lowest quality) to 5 (highest quality).

A structured approach to define and measure QoE in relation to QoS is explained in [220], in which QoE is influenced by four essential attributes: the communication situation, service prescription, technical parameters and user experience. The developed QoE measurement scheme includes objective parameters of user performance, thus providing higher validity, and allows industry professionals to compare various services, products and QoS levels.

In [221], it is proposed a generic formula in which QoE and QoS parameters are connected through an exponential relationship. This QoE formula is validated by using three case studies (voice quality, user reactions to download times and throughput limitations) that provide better QoE estimates than the original logarithmic approximations. Other works define new ways of characterizing QoE from network-level QoS parameters by using empirical rules [222], correlation models [223], utility functions [224] and fuzzy logic approaches [225].

The authors of [226] present the Quadrant of Euphoria, which is an user-friendly Web-based platform facilitating QoE assessments. Such platform is said to be low cost and represents participant diversity, which provides meaningful and interpretable QoE scores. Finally, [227] designs a proactive in-service QoE controller service enabler that uses a QoE estimation algorithm based on objective E2E QoE modeling principles, rather than relying on subjective user feedback acquisition or legacy QoS methodologies.

In this Thesis, we do not aim to define a new metric to measure QoE in a general network environment. However, we need to take into account that in the proposed multi-VN architecture, the quality perceived by a VN user is not only related to the state of such VN, which can be measured based on QoS metrics, but it is also related to the VN context purpose. Since a VN is configured to meet a specific set of user context requirements, it is then important to consider the impact of such requirements on the overall VN performance, and consequently, on the perceived QoE of VN users.

2.4.3.2 QoE-Aware Wireless Mesh Networking

In a multi-operator, multi-network and multi-vendor networking environment, users are expected to experience a high-bandwidth and ubiquitous network access through wired or wireless technologies, as well as diversified service provisioning [228]. For this reason, service providers need to explore new ways to properly adjust several network control and management processes based on the user perceived QoE. These processes can span from access network selection, routing, resource allocation algorithms, QoS mapping, transport functions or application-level parameter configurations [229].

In the following, we describe several proposals, most of them originally designed to WMNs, that are aware of user QoE feedbacks in order to improve and control the performance and behavior of several network functions to ensure that the QoE expectations of users are met.

In [230], it is proposed a cross-layer optimization strategy across multiple application classes that uses a MOS-based objective function to jointly optimize the application layer, the data-link layer and the physical layer of a wireless protocol stack. The results of such strategy, in terms of network resource usage and user QoE, clearly outperform the ones derived from a conventional throughput-based optimization strategy. The authors of [231] carried out an experimental survey of the user subjective quality of Web browsing, and derived a mapping function from the service response time to the user MOS-based QoE. Such function is then incorporated in a radio resource allocation algorithm in order to maximize the aggregate utility over all the users. In [232], it is defined a cross-layer network management strategy that allows a service provider to dynamically adjust the network resource reservation with its network operator based on the users' perceived QoE of the offered services, in order to minimize the service degradation periods. [233] presents a QoE-aware autonomic management architecture to: *(i)* detect the network problems that can lead to QoE drops for a given service (e.g., a congested link or bit errors on a link); *(ii)* determine an appropriate action to increase the QoE (e.g., switching to a lower bit rate or adding an appropriate number of Forward Error Correction (FEC) packets). In [234], it is proposed a dynamic bandwidth control mechanism for real-time applications that measures the current WMN situation and adapts the bandwidth of network flows, in order to ensure a high overall QoE level.

Concerning the video streaming, [235] defines a scheme to perform the resource allocation of each path with a video stream routed over multiple paths over wireless networks. This scheme works in such a way that the total weighted QoE of some competing video streams is optimized using a cross-layer design technique. Similarly, [236] and [237] present QoE-driven schemes that dynamically optimize the transmission rate and the network resource utilization for video services in wireless environments. The dynamic rate-adaptation of multicast video applications according to the users' QoE feedbacks is analyzed in [238], in which the rate is reduced or increased, when users respectively have bad or good QoE.

In [239], it is designed a packet scheduler that operates at each WMN node and maximizes the user perceived QoE of multiple flows, including audio, video and data. To this purpose, it is developed a mathematical model to determine the impact of packet dropping on different media types, which is then applied to an optimization algorithm that maximizes the overall flow utility under given resource constraints. Moreover, the authors in [240] present a QoE-oriented approach for handover management. This approach tries to improve the user perceived QoE, when roaming across a heterogeneous wireless network environment, by exploiting all the communication resources available in the user terminal. In [241], it is proposed a scheme that uses QoE indicators of ongoing users in candidate access networks in order to select the best network to assign to a new user.

Finally, there are several approaches that incorporate QoE feedbacks in learning-based strategies in order to improve the performance of wireless networking. In [242], it is introduced a cross-layer multi-path routing protocol for video streaming in MANETs, which includes a game theoretic approach to perform a dynamic selection of the forwarding paths. By taking into account the importance of the video frames in the decoding process, such approach has proved to enhance the performance of the overall user QoE, even in presence of network interference or user mobility. Moreover, [243] studies the user-centric service selection for IPTV services in heterogeneous wireless networks by using a game theoretic approach that considers functions for the user satisfaction and for the cost to switch to an alternative IPTV service provider. In [244], it is proposed a scheme that allows users to gather the information of the different network technologies and network providers available in a heterogeneous wireless system. Then, and using a learning strategy based on evolutionary game dynamics, users can update their vision about the current environment, and can select the best wireless technology to be connected according to their satisfaction measured in terms of a QoE metric.

In contrast to the approaches available in the literature, this Thesis has a two-fold objective with regard to QoE-aware network management. First, we aim to define a strategy that makes use of the overall quality status of each VN derived from the mean perceived QoE of the ongoing VN users, in order to increase the accuracy of the selection and adaptation of the best fitting VN to assign to a new user (from the candidate set of fitting VNs). Second, we aim to propose a hop-by-hop reinforcement learning strategy to dynamically compute the best VN routes on-demand based on the QoE feedbacks of ongoing VN users. Such solution is intended to leave behind the well-known problems of traditional proactive and reactive routing approaches for WMNs, and of the QoS-aware routing strategies that are not aware of the user QoE expectations. Moreover, this reinforcement learning routing solution can be dynamically adapted to VN topology or resource availability dynamics.

2.5 Distributed Network Control & Management

In the architecture proposed in this Thesis, VNs are built to serve the users in a personalized way. This way, VNs need to be frequently adapted in order to be incorporated in new WMN nodes due to user mobility, or new VNs have to be created in order to deal with new sets of user context requirements. Due to the dynamics of this architecture, it is important to define a flexible, scalable and adaptable VN control and management scheme that leaves behind the limitations of centralized approaches, such as processing bottlenecks, required high capabilities of a central server and limited scalability.

In this context, this Section first presents several architectures and proposals available in the literature that distribute the control and management responsibilities along the distinct entities in a specific networking environment, giving a special emphasis to WMNs.

Finally, this Section describes several approaches that make use of context information to optimize the cooperation and coordination among the network entities, in order to enhance the performance of distributed network control and management processes.

2.5.1 Distribution of Network Control & Management Functionalities

The Internet is envisioned to be a platform that provides support for multiple real-time services and complex multimedia applications for hundreds of millions of users, as well as the communication backbone of the emerging digital society that has significantly been evolved in both wireless and wired environments [245]. Due to this growing complexity of communication networks, the future Internet paradigm poses new challenges to network management. It is then required to extend the current Internet model to include QoS guarantees and differentiation mechanisms, reliability features and security assurance, or service-level agreements among infrastructure and service providers.

However, the traditional centralized approaches to control and manage the current Internet are not suited to deal with the novel requirements in the future, since they are not scalable and robust to cope with the high number of network nodes, services and operations available in the future. In addition, nodes might constantly join and leave the network and might not be reachable from a central station, which does not allow for an exact real-time network view when the dynamics is too high [246]. This way, several approaches have been proposed in the literature to leave behind the recognized drawbacks of centralized schemes to control and manage nodes and networks. These approaches aim to balance the control knowledge along distinct network entities, in order to perform several network control and management processes in a dynamic and distributed way. In the following, we first present several of these proposals in the context of a general networking environment, and then we describe the approaches focused on WMNs, highlighting how we aim to go beyond them.

In [247], it is proposed a decentralized access control model to enable network customers to have permissions and rights for the configuration of their devices. Such model allows the customizable granularity of access control policies and it defines a standard-based approach to access control in network configurations. The authors in [248] present a framework to perform two key network management functions, namely, the real-time monitoring and the event handling, in the scope of the In-Network Management (INM). INM is a systematic approach to embed and decentralize several management algorithms within the elements of a communication network. In the context of INM, [249] is a decentralized approach to turn on or off certain network management functions based on a probability, in order to reduce the efforts and resources dedicated to the management of large scale and dynamic networks. The aforementioned probability can be influenced by the operator, the node capabilities, the function itself or any other external information. In [250], it is proposed a distributed framework to control the topology of dynamic multi-agent networks, where the agents are equipped with local sensing and wireless communication capabilities. Due to power constraints, these agents are required to switch between the active and sleep operation modes. Therefore, [250] proposes a distributed coordination protocol to regulate the switching between the operation modes of every agent, while guaranteing an overall subset of multi-hop connections among several of these agents. [251] introduces a strategy for the scalable decentralization of control operations, called SOMEN, which is built based on the distribution of multiple network control decision points. The effectiveness of SOMEN was demonstrated by using the information of communication paths' correlation patterns and links' resource conditions on the over-reservation resource control in decentralized networks. Moreover, [252] and [253] are solutions focused on the distributed resource discovery by using the attractive properties of P2P overlay networks [254], which were initially designed to provide a good substrate for creating large scale and distributed data sharing, content distribution and application-level multicast applications. In [252], the authors use a P2P overlay to manage different wireless access networks, in which each peer stores a description and a pointer to each wireless access network indexed according to its geographic position and the range it covers. In [253], it is used a P2P overlay to provide a distributed directory service that allows the participants to discover resources and maintain resources' states. Finally, there are several approaches, such as OCALA [255] and SpoVNet [256], that enable the existence of multiple application-aware overlays over the same network infrastructure, each one providing support for specific distributed services.

Focusing on WMNs, [257] proposed a distributed QoS-aware admission control scheme to support real-time services in WMNs. Here, the WMN is partitioned into cliques, and only clique heads are involved in the admission control algorithm that aims to achieve high WMN resource utilization, while preventing the packet loss and delay metrics from exceeding predefined thresholds. The authors in [258] present a decentralized approach for the autonomic management of collaborating base stations. The individual base stations first aggregate and share network information from their neighboring base stations. They then implement a distributed algorithm that uses such information to automatically bootstrap, configure and manage themselves. MeshChord [259] is a distributed approach for the P2P-based resource sharing in WMNs based on the Chord protocol [260]. Such approach implements location-awareness and MAC layer cross-layering by resorting to the properties of WMNs, namely, the availability of a wireless infrastructure and the 1-hop broadcast nature of wireless communications. Following this strategy, MeshChord aims to improve the performance and reduce the overhead of the Chord protocol in WMNs. In [261], it is designed a fully distributed protocol for the broadcast time slot assignment in large WMNs. By employing contention-based reservation mechanisms in a topology-dependent way, every WMN node acquires a broadcast time slot that supports a minimum average SINR to all of its neighbors. The work presented in [262] investigated the feasibility and drawbacks of the existing distributed MAC protocols when they are implemented in WMNs, in the avenues of best-effort service support, priority guarantee, resource reservation, fairness enhancement, multi-channel communication support and cross-layer design. A distributed max-min fair rate allocation mechanism for WMNs is proposed in [263]. The max-min fair rate of a multi-hop flow is first derived by computing the max-min fair rate at each hop along the flow path, and finally, it is enforced the rate offered to the flow at the most constrained hop in the path. In [264], a distributed packet-based cross-layer algorithm is designed for the transmission of multiple video streams over a WMN. This algorithm considers the packet-based distortion impact and delay constraints, based on which, the most important video packet is selected and transmitted at each WMN node over the most reliable link, until it is successfully transmitted or its deadline is expired. Focusing on energy management in the context of WMNs, [265] describes a localized distributed topology control algorithm to derive the optimal transmission power at each WMN node, so that the WMN connectivity is maintained, the transmission power at each node is reduced to cover only the nearest neighbors, and the WMN lifetime is extended. Further, [266] presents the mathematical and simulation studies of a decentralized algorithm to dynamically optimize the WMN routing by minimizing power consumption objectives. The algorithm robustness is demonstrated with respect to time-varying transmission channels and node failures. Finally, [159] presents a WMN architecture, called OverMesh, that aims to build several overlays on top of WMNs in order to deploy distributed services and applications.

In the scope of the architecture defined in this Thesis, the presented solutions have potential to be integrated in the nodes of the physical WMN infrastructure or of the VNs, as a mean to perform several network control and management processes in a distributed way. However, in contrast to the proposals available in the literature, our main goal is to define an overall distributed framework to autonomously control and manage a virtualized WMN environment, in which each VN can be specialized to meet a specific context purpose and can be instantiated and controlled by a distinct VN provider or network operator. More specifically, we first aim to distribute specific control knowledge and intelligence along the WMN nodes, VN nodes and VNs that compose the architecture. Then, we aim to define a flexible scheme to enable the dynamic cooperation of these entities in order to locally or globally perform the discovery, selection, creation, adaptation or remotion of VNs on-demand to react to any type of user context change and mobility.

2.5.2 Context-Driven Approaches for Distributed Control & Management

When dealing with distributed network control and management processes, the knowledge and intelligence distribution along distinct network entities is not a straightforward task, since it can lead to complex structures to enable an accurate coordination among these entities, while ensuring a good performance of the distributed network processes.

This way, intelligent policies are required to ease or personalize the distributed coordination among the network entities with control and management responsibilities, e.g., to select what entities should be connected or to define what information should be exchanged among them. These policies can be thought by inspecting the context information characterizing such entities or the context purpose of the distributed network processes that need to be performed, such as content distribution, topology construction or adaptation, node/network discovery or resource management. Through these context-driven policies, the complexity required to coordinate the distinct network entities can be decreased, while improving the delay and overhead performance of the distributed network processes.

In this sense, we present below several strategies for the context-driven distributed creation and adaptation of services, routing mechanisms and other network operations. These strategies are also aligned with the main goals of this Thesis.

Focusing on the use of context information to dynamically build service-aware overlay networks, [267] presents a self-management approach to create, configure, adapt and teardown such overlays in the scope of the Ambient Networks project [268]. Such approach enables the distributed composition and delivery of services to the end-users, and also the dynamic self-adaptation of such services according to the changes of user/network context. A similar approach is proposed in [269] to form a policy layer that consists of a set of network agents. These agents dynamically cooperate to manage the behavior of a service-aware overlay network according to the context of the user, the network and the service providers. In the context of the European project AUTOI [270], the authors of [271] proposed a protocol to select and continuously maintain a set of network nodes to act as distributed service directories. By using P2P principles, these service directories keep track of the different types of services that are currently available in a network, being able to access the services that are needed to implement more tailor-made service(s) that consumers want. In [272], it is proposed a management scheme that accomplishes context-aware and dynamic adaptive features on the construction of the chains for service composition in service-aware overlay networks.

Concerning the context-driven global distributed discovery of network content, which can be associated to the information stored by specific nodes, peers or servers, there are several approaches that make use of the context information characterizing such content, in order to improve the searching and lookup mechanisms. In [273], it is proposed a semantic P2P context lookup system, where peers are grouped based on the semantics of their local data and self-organized as a semantic overlay network. This way, search requests are only routed to the appropriate nodes that have relevant data, thus reducing unnecessary query traffic and delays. The authors of [274] present a distributed information sharing system that supports semantic-based searches of similar documents in a P2P network. Here, the documents are summarized according to their content and maintained within the network with different granularity. Moreover, [275] describes a semantic overlay network of service repositories, which can be published by distinct service providers. Such approach exploits semantics and P2P technology for query routing and for grouping repositories that have similar information. In the following, these repositories form a group of coordinating nodes that allows the execution of complex queries with minimal query response times. DESENT [276] and INGA [277] are solutions that enhance the unstructured P2P-based Web searching through the building of semantic overlay networks based on content and topology knowledge [276] or based on similar social and community interests [277]. Finally, in [278], it is presented a framework that uses multiple context parameters of mobile nodes, such as available bandwidth, disconnection rate or remaining battery capacity. This framework then integrates these parameters into a single metric that is able to allocate each node to a P2P topology, in order to minimize negative effects of link instability and to ensure high service quality.

In this Thesis, we make use of the context information that characterize users and VNs in order to enhance the global distributed discovery of points of attachment of VNs available in the WMN. More specifically, since each VN is specialized to meet a specific set of user context features and can be instantiated by a specific VN provider or network operator with its own control authority or administration, we aim to define a distributed control structure that: *(i)* dynamically acquires the context information of the VNs available in the WMN; *(ii)* automatically creates and adapts several connections among similar context-aware VNs. From an user-centric perspective, similar context-aware VNs are the ones that are most probable to deal with the particular user context requirements and their variations. Therefore, this context-aware inter-VN cooperation can allow the fast user re-association to another VN that better meets the user context needs. Moreover, it can allow the fast context-driven redirection of the global VN discovery process in the WMN through an intelligent balancing of the control responsibilities along the distinct VNs.

2.6 Conclusions

Due to the strong focus on WMNs in this Thesis, we started this Chapter by providing the background of the WMN technology, which encompassed the description of its key characteristics, real world deployments and open issues of several WMN control and management processes. At a first glance, one can easily detect from the presented WMN overview, that the infrastructure, protocols and mechanisms of WMNs are characterized by a high level of flexibility, self-configuration, self-organization and self-adaptation, which can be used in favor of many network scenarios or inspiring numerous applications. In the context of the recent proliferation of multimedia content consumption, common usage of personalized services and large advances of mobile devices, which are triggering a whole body of research towards the future Internet, these attractive WMN features can also be used to deploy and test novel communication and architectural paradigms that are aware of the growing list of requirements of mobile users, devices, services and other network elements. Therefore, this Thesis aims to define an architecture to personalize WMN communications and control functionalities according to the preferences and expectations of mesh clients, and the needs of their devices and services, which we denominate by user context. More specifically, we want to quantify and model the user context requirements, and then use such information to dynamically build a set of logical networks that share the same physical WMN infrastructure. Each logical network will be specialized to meet a specific set of user context requirements through the proper configuration of its assigned resources, routes, paths and other control and management operations. In our approach, users can be grouped according to similarity on their context needs and assigned to VNs matching these needs.

Network virtualization is currently one of the most promising technologies to enable the support, in the actual Internet, of new architectures, protocols and other innovations from the research community in a flexible and modular manner, with enough isolation to separate experimental from real traffic. This way, this Chapter also provided an overview of the interesting concepts, features and challenges related to network virtualization. Since we aim to build an user-centric architecture that splits a WMN in a set of context-aware logical networks, it is then important to make use of the attractive properties of network virtualization to: *(i)* split a physical WMN infrastructure in a set of adaptable VNs in a flexible, dynamic and programable manner; *(ii)* configure the resources and protocols of each VN to properly meet a specific set of user context requirements, with the level of isolation and transparency required. The definition of a context-aware WMN architecture, built through network virtualization, is also an important step to stimulate the extension of the virtual platforms of network operators up to wireless environments.

This Chapter also dived into the description of the proposals available in the literature that make use of several context information available in WMN environments as an opportunity to enrich or personalize several WMN processes and control functionalities, instead of only a challenge that raises the WMN complexity. By performing a deep literature inspection, we come to the conclusion that most of these context-aware proposals only consider a single or a very small set of context requirements. Moreover, they are narrowly used to improve specific WMN control and management mechanisms, and do not incorporate an user-centric view of context information. Therefore, in contrast to the approaches available in the literature, we aim to quantify and model a multi-variate set of user context requirements in specific levels, and then map such levels in proper VN features that are able to characterize and personalize such VNs to meet the user needs. Moreover, we aim to embed information related to user context requirements and their ranges of variation in the definition of the metrics to select the best fitting VNs to assign to users, or the best WMN paths to perform the adaptations of VNs in order to react to any type of context change.

Since the user QoE expectations are important parameters that belong to the user context, this Chapter also provided an insight on the solutions available in the literature that adapt the behavior of several WMN operations according to the QoE feedbacks of mesh clients, in order to better meet the user QoE expectations. Despite being very interesting, the QoE-aware wireless mesh networking solutions available in the literature do not have the required tools to reach the goals of this Thesis, since they do not consider a multi-VN environment to serve multiple sets of user context needs in a personalized way. In this Thesis, we aim to: *(i)* make use of the QoE feedbacks of ongoing VN users to derive an overall quality status of each VN, which can be used to select VNs with high quality indicators; *(ii)* define a hop-by-hop reinforcement learning strategy to dynamically compute the best VN routes on-demand based on the QoE feedbacks of ongoing VN users.

The centralized network control approaches have serial scalability limitations, since they do not allow an exact real-time network view and an effective coordinative functionality when the network dynamics is too high. Therefore, this Chapter also summarized the most important solutions to distribute the control intelligence and functionalities along distinct network entities, and to then enable the cooperation of these entities to perform several network control and management processes in a distributed way. As a general remark regarding the presented distributed solutions, we can conclude that they do not enable the autonomic control and management of a virtualized WMN environment by fostering the distributed cooperation among the distinct WMN nodes, VN nodes and VNs, which is one of the aims of this Thesis. Further, none of works available in the literature are suited to improve the cooperation among similar context-aware VNs, in order to allow the context-driven VN discovery in the WMN.

After providing the background and related work that lead us to think in a new user-centric architecture to be applied to WMNs through network virtualization, and to be controlled and managed in a context-aware and distributed way, the following Chapters will provide an in-depth discussion of the main features, requirements, entities, functionalities and challenges raised by such architecture. They will also propose, model and evaluate the framework, mechanisms and algorithms to meet such challenges, comparing them with other approaches available in the literature.

Chapter 3

Context-Based Wireless Mesh Networks

After setting the background of the presented Thesis, this Chapter presents an user-oriented context-aware architecture to personalize WMN communications through network virtualization. Beyond the description of the goals and characteristics of such architecture, along with the main important functionalities of each architectural entity, this Chapter highlights the challenging issues related to the context management and VN control, which will be deeply discussed throughout this document. Later in the Chapter, we propose a generic probabilistic-based analytical model that tackles the most important aspects of the architecture. Using this model, and validating its results against a simulation study, we evaluate the impact of network virtualization on our approach.

3.1 Introduction

The recent proliferation of user-friendly portable devices, high quality multimedia services and innovative technologies that are able to provide ubiquitous connectivity, are changing the original Internet nature, thus triggering many research efforts towards the future Internet. This whole body of evolution is converging in the same Internet-based network infrastructure, which is reaching a breaking point trying to provide enough flexibility for end-users and profits for network operators, while meeting the strict context requirements of end-users, devices, services and other network entities. Therefore, tightly coupled with the future Internet research, new network architectures and mechanisms are required to deal with the context requirements brought by the emergent digital, multimedia and technological evolution.

To achieve this, Section 3.2 proposes a context-aware communication and architectural paradigm to be applied to WMNs, which, due to their flexible and self-adaptable characteristics, have been viewed as an important component of the future Internet. Briefly, this paradigm is focused on the quantification and modeling of the context requirements characterizing the mesh clients. This information is then used to split the physical WMN infrastructure into a set of several logical networks, each one personalized to meet the specific requirements of a group of users. These logical networks are autonomously built and adapted on-demand through network virtualization.

Beyond the description of the main goals and features of a multi-VN paradigm to enrich the communications of mesh clients, Section 3.2 also presents the key control and management challenges raised by the need to autonomously adapt VNs to any type of context change. Moreover, it describes the main important architectural entities, along with their functionalities and interactions, to materialize the proposed paradigm and its intrinsic mechanisms.

The use of network virtualization as a flexible tool to enable non-interfering and personalized WMN communications requires the evaluation of its impact on the architecture. Therefore, in Section 3.3, we propose a generic analytical model that covers the key aspects of the proposed architecture. By using probabilistic-based functions for WMN and VN topology, user arrival, traffic patterns and wireless interference, the model is able to capture the impact of network virtualization on the mean E2E delay of WMN communications.

In the following, Section 3.4 details the key implementation issues of the simulation platform designed to test several architectural mechanisms during the Thesis period.

Using the analytical model defined in Section 3.3, Section 3.5 provides the conceptual boundaries of the architecture, in terms of the number of VNs, attached users and virtual nodes per VN, in order to accomplish feasible delays for different data communication requirements. Section 3.5 also compares the theoretical analysis described above against a simulation study, and evaluates the influence of using a real WMN topology to materialize the multi-VN architecture. We close the Chapter by summarizing its achievements in Section 3.6.

3.2 Context-Aware Paradigm for WMNs

3.2.1 Concept Overview: Goals & Features

The self-* and flexible properties of the WMN infrastructure and protocols, along with the potential of WMNs to enable broadband Internet access in wireless environments, can be a great advantage to be engrained in the design of novel communication and architectural paradigms. One of the key requirements of the future Internet is to be aware of the demanding context requirements boosted by the recent (and foreseen) digital, multimedia and technological advances, which have collided with the small flexibility of the original Internet to meet them. In this scope, WMNs can be considered as one of the key technologies to provide support for the envisaged context-aware paradigms that will be part of future network architectures.

Typical WMNs are accessed by mobile users characterized by distinct mobility patterns, QoE expectations, price preferences, and requiring Internet accessibility everywhere and anytime. As a consequence, such users need to be linked to wireless access networks with specific cell ranges and robustness features, and probably belonging to ISPs with specific trust and security policies. Moreover, user devices are characterized by sophisticated capabilities and energy requirements, and such devices are able to run different applications and services with strict QoS requirements.

This huge WMN context heterogeneity has been seen as an opportunity to enrich wireless mesh networking by resorting to the highly attractive properties of WMNs, instead of being only seen as a challenge derived from the growing complexity of context relationships. First, the multi-hop and multi-path availability in WMNs, along with their high flexibility in changing and switching routes and transport connections for different context-aware purposes, can be used in favor of the traffic management or multi-homing support. Second, the provision of WMN communications can be improved by an appropriate WMN dimensioning and deployment based on the available technologies and in accordance with the expected QoS/QoE user demands (this can even lead to a better WMN resource usage). The literature review done so far has led to the conclusion that there is still room for innovation in the use of WMN context information as a mean to deploy a more flexible user-centric context-aware paradigm to personalize WMNs.

The features of network virtualization can be considered an added value for the personalization and enrichment of the WMN connectivity up to a high degree. Network virtualization has recently been advocated as one of the key enabling tools of the future Internet, since it enables the aggregation or sharing of physical network resources among different virtual instances in a flexible, dynamic and programable manner. Therefore, network virtualization can be used to deploy heterogeneous network architectures that cohabit on the same physical network infrastructure, or to implement and test new communication paradigms with the level of isolation and transparency required.

According to the above discussion, this Thesis proposes an architecture to be applied to WMNs through network virtualization, being such architecture driven by a general model of user context (published in [10]). By user context, we consider the security, price and mobility preferences of users, their QoE expectations, the capabilities of their devices or the QoS requirements of their services. As depicted in Fig. 3.1, we split a physical WMN into a set of VNs, where each VN is built, configured and adapted on-demand to properly meet a particular set of user context requirements.

More specifically, by means of network virtualization techniques, each physical mesh router can support multiple virtual nodes, each one configured in terms of, e.g., processing power and memory, to meet the context requirements of several users. Virtual networks can then be built by connecting the virtual nodes in which users with similar context needs are connected to. Each VN can be instantiated and controlled by a specific VN provider or network operator, and can use of an isolated and dedicated wireless channel. A high level of WMN personalization can be achieved by running the same protocols and mechanisms (e.g., security or mobility protocols) in all the nodes of a VN, which can then cooperate to better fulfil the

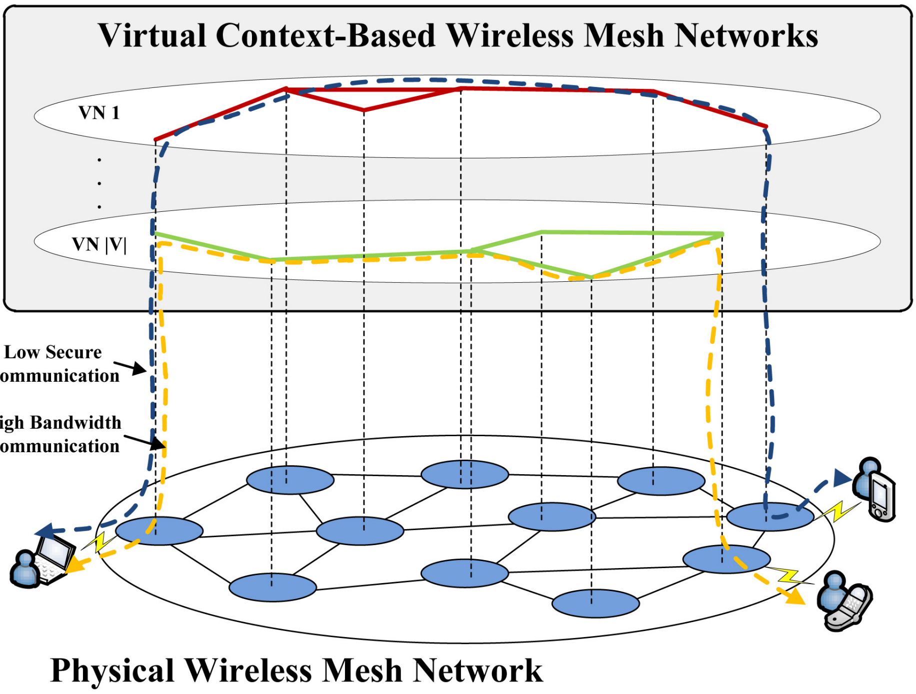


Figure 3.1: Multi-Virtual Architecture for Context-Based WMNs.

VN context purpose. Following this strategy: *(i)* one VN can be endowed with fast mobility-aware mechanisms in order to serve users that require fast handovers; *(ii)* a highly secure VN can be offered to mesh clients if needed, without resulting in delays for the remaining clients, which can be assigned to VNs characterized by low security mechanisms; *(iii)* a VN can be configured to enable a better fulfilment of strict QoS requirements than other VNs; etc.

According to the above discussion, the use of network virtualization to materialize the WMN architecture into several and independent logical networks appears to be promising. First, it enables a programable and flexible sharing of the WMN infrastructure to serve the VNs, which can have configured their own context-aware services and protocols. Second, it allows the grouping and forwarding of data flows with the same context patterns through the same WMN routes or channels, which can decrease the wireless interference among communications of different types. Third, from a network operator point of view, it is envisioned that well-managed VNs can stimulate the extension of the virtual platforms of network operators up to the WMN environments, which can thus maximize their infrastructure revenues. Finally, by means of network virtualization, an user can simultaneously access to different context-aware VNs, and multiple paths of distinct VN providers can be aggregated to increase reliability.

Users are grouped according to similarity on their context needs, and assigned to VNs matching these needs. Due to the constant change of the locations, required services and preferences of users, the VNs need to be dynamically discovered, selected and adapted to quickly react to user context change or mobility. If none of the VNs already available in the WMN fits the requirements of a particular user, a new VN can be created in the WMN. This way, most of the VN control processes raised by our approach remain in the sphere of control of the mobile users which define the context requirements. An intelligent understanding and modeling of the user context can then ease the user-centric and autonomous functionality and control of the architecture. In parallel to this user-centric perspective, the resource conditions of the WMN nodes have to be considered in the network-centric selection of the paths to create or adapt VNs in the WMN, as a mean to improve the WMN capacity.

In summary, the presented multi-VN architecture for WMNs, which can be supported by a multi-operator environment in the same or different WMN infrastructures, is capable to provide personalized support for a multiplicity of users, devices, applications and network protocols or mechanisms. It makes use of a generic user-centric context model to autonomously create, configure, adapt and remove VNs on-demand. Each VN is able to enrich the connectivity and increase the QoE of a group of users that share similar context requirements, and it is dynamically adapted in a resource-effective way to the change of these context requirements (these two aspects suit both user and network needs). These functionalities and mechanisms will be developed along this Thesis.

There are specific environments that may benefit from this type of context-aware networks in order to optimize group-based communications, community networks or content distribution. Those environments can span from residential areas, schools, universities or college campus, to business or industrial areas and tourist attractive areas.

3.2.2 Control & Management Challenges

The advantages that can arise from the splitting of a physical WMN infrastructure into a set of user-centric context-aware VNs raises several challenges with regard to the design of an autonomous context-aware management scheme to make the VNs usable, scalable and adaptive. In the following, we introduce the main issues that are under the vision of the proposed architecture, which are summarized in Fig. 3.2. These issues will be addressed in the following three Chapters, in which we present and evaluate the proposed solutions to tackle them.

Since one of the key requirements of the architecture is to be context-aware, it is important to acquire and model the context information that is relevant to self-configure and self-adapt on-demand the user communications and VN topologies according to any type of context change (Fig. 3.2→1). Such context information can span from the requirements, preferences and location of users, characteristics and quality status of VNs, and resource constraints of WMN nodes. The multi-variate context information has to be modeled in a way that simplifies the processes by which such information is stored, transported and used by the different architectural elements and mechanisms. Hereby, the user context needs and preferences can be modeled using ontologies (or meta-information) based on simple context levels capable to describe complex situations or relationships; the quality status of VNs can be derived from the overall QoE perceived by their associated users; the resource availability at each WMN node can be defined as a percentage of the total resource capacity of such WMN node. Chapter 5 will give special emphasis to these context modeling mechanisms; however, and when necessary, these mechanisms will be used or referred throughout this document.

The VNs are built and configured to personalize the communications of a group of users that share the same (or similar) context needs, and therefore, rules are required to automatically map these context needs into suitable VN features (Fig. 3.2→2). These context mapping rules can lead to the definition of algorithms for VN resource assignment driven by the specific context-aware characteristics of each VN (Fig. 3.2→3). Fully aligned with the

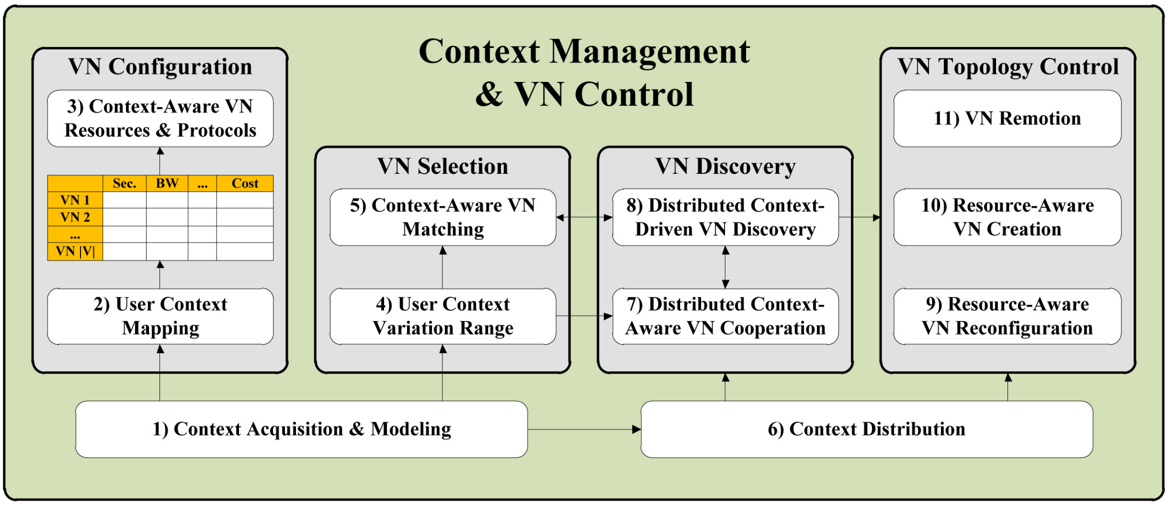


Figure 3.2: Control & Management Challenges.

proper context-aware VN configuration, the same protocols and mechanisms (e.g., security protocols or mobility-aware mechanisms) shall run in all VN nodes, which can then cooperate among them to better fulfill the VN context purpose (Fig. 3.2→3). The envisioned cooperation among the nodes of each VN, which can periodically exchange the relevant information of their current status or attached users, can also be used in the design of an intelligent VN routing protocol that dynamically selects the non-congested VN paths among distinct VN nodes. Apart from their frequent reappearance in several Chapters of this Thesis, the mechanisms for the user context mapping and the context-aware VN configuration will be mainly addressed in Chapter 5.

Due to the constant change of the user context requirements, we consider certain levels of flexibility of these requirements, in order to allow users' associations to VNs that do not exactly fit all their requirements, but may incorporate these users if their context flexibility allows to. This user context flexibility is important to avoid the constant creation of new VNs to serve single users, and it can be modeled by imposing a certain range of variation for the demand of each requirement during the user connectivity time (Fig. 3.2→4). Thus, the requirements that are more probable to change are the ones admitting high variation ranges.

Since each VN is configured to meet a specific set of context features, the allowed variation in the user context requirements can enable the user association to more than one VN available in the WMN. Therefore, the metric to select the best fitting VN to assign to a specific user has to consider the need to select VNs that are more probable to deal with high variations in the user context requirements, which can reduce the number of VN re-associations during the user connectivity time (Fig. 3.2→5). Further, such metric has to be designed to cope with the overall quality level of the candidate VNs for the user, which can be defined as the overall perceived QoE of their ongoing users. The work presented in Chapter 4 will be based on the selection of exactly fitting VNs for users; however, a more intelligent context-aware VN selection metric will be defined and evaluated in Chapter 5.

Each VN is dynamic in the sense that it may be dynamically built and adapted at runtime in order to cope with any type of user or network context change. Since the WMN infrastructure is the physical substrate for several VNs, we need to deploy a platform that dynamically gathers the ongoing context characterizing users, VNs and WMN nodes. Such information can then allow the configuration and adaptation of single VNs in the WMN without disregarding the presence of other VNs. This platform can be designed in a centralized way, where one element is responsible to collect all the context knowledge used to control and manage VNs. However, centralized approaches have known limitations, such as limited scalability, leading to bottlenecks and imposing single points of failure. This way, it is important to distribute the relevant context knowledge in the WMN (e.g., information of VN features/quality and indicators of WMN resource availability), so that each WMN node is able to be aware of both itself and its neighbor environment (Fig. 3.2→6).

The VNs can be instantiated by specific VN providers with their own administration. The proposed approach also contains a distributed scheme to enable the periodic exchange of specific knowledge among VNs characterized by similar context features (Fig. 3.2→7). From an user-centric perspective, similar context-aware VNs are the ones that are most probable to deal with user context flexibility during the user connectivity time. Therefore, the context-aware inter-VN cooperation can allow the fast and context-driven global VN discovery in the WMN. Despite the description of the main guidelines of a scheme to enable the dynamic cooperation of similar context-aware VNs in Chapter 4, the detailed definition and the overall evaluation of the mechanisms involved in such scheme will be performed in Chapter 6.

After the distribution of the context knowledge in the WMN in order to enable the cooperation among neighboring WMN nodes or similar context-aware VNs, these entities will then cooperate in the distributed VN control and management. These control and management mechanisms can include the local or global discovery and selection of a fitting VN to assign to an user, or to adapt the user connection to a better fitting VN in case of user context change (Fig. 3.2→8). It may also be required to quickly reconfigure the discovered fitting VN to be incorporated in the WMN node where the user is connected to (Fig. 3.2→9). This VN reconfiguration process has to be aware of the information about the WMN resource availability in order to select a non-congested WMN path to perform the VN reconfiguration. If the WMN does not provide support for any fitting VN for the user, the exactly fitting VN for the user can be created if there is a sufficiently large number of users that would be better served if their connections would be changed to the new VN (Fig. 3.2→10). Moreover, due to the dynamics of the network, several nodes of each VN (or even the whole VN) can be removed, when there are no attached users, to free wireless resources (Fig. 3.2→11). In Chapter 4, we will present a basic mechanism to distribute context knowledge along the WMN and to use such knowledge to dynamically discover and adapt VNs according to the arrival or mobility of users. Later, Chapter 6 will define an overall distributed and context-aware framework to autonomously discover, select, create, adapt and remove VNs on-demand.

Finally, the use of network virtualization as a flexible and programable tool to build and configure the VNs imposes the need of a scheme to dynamically map, schedule, and switch wireless interfaces and channels in order to serve the multitude of VNs. Despite the importance of this thematic, these lower layer issues will not be addressed is this Thesis.

3.2.3 Architectural Entities & Control Functionalities

We describe below the main important architectural entities, along with their key functionalities and interactions, to provide support for the proposed paradigm. The description of the role of each architectural entity, which is summarized in Fig. 3.3, will be driven by

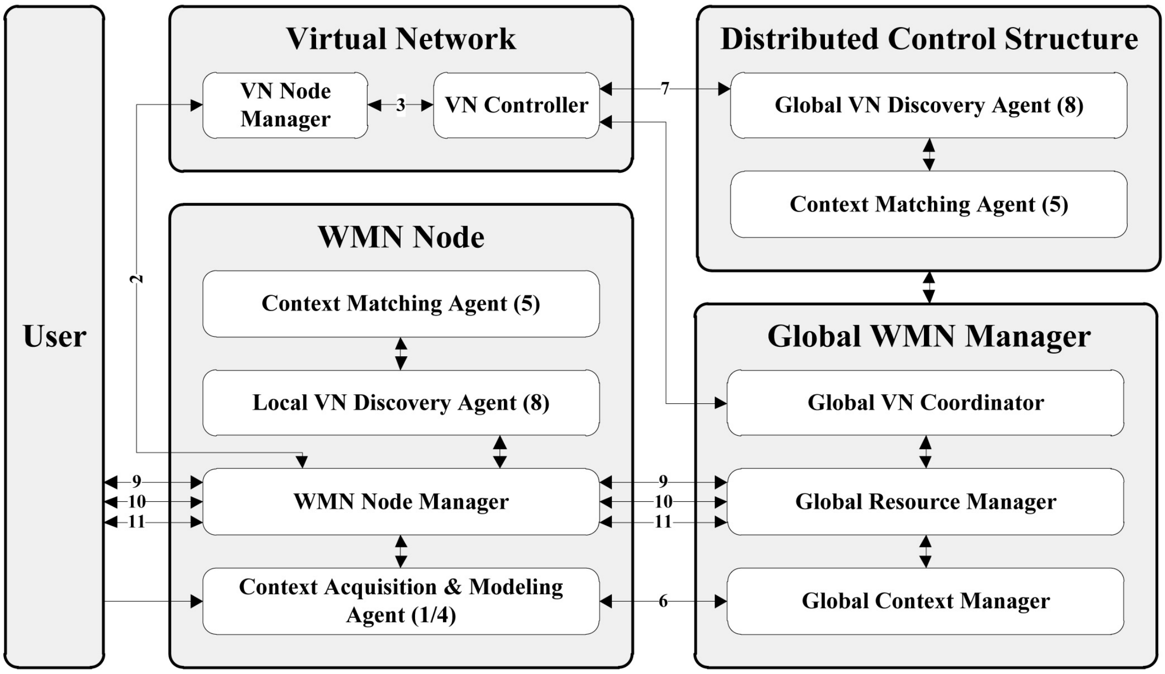


Figure 3.3: Architectural Entities & Control Functionalities.

the requirements and goals of the control and management mechanisms presented in Section 3.2.2. For a better understanding of the reader, Fig. 3.3 recalls the numbers used in Fig. 3.2.

3.2.3.1 User

Despite not being directly involved in any control process, the user, which defines the context requirements, is the core component to enable the context-aware operations and functionality of the overall architecture. The particular user context requirements that can change at a significant rate, along with the user mobility, are the most important factors to trigger the discovery, selection, creation, adaptation and remotion of VNs in the WMN.

3.2.3.2 WMN Node

The WMN node is the physical substrate for a set of VN nodes, and therefore, it is responsible for the context-aware configuration and scheduling of its resources among the VNs that it supports, according to specific context mapping rules. To perform these local resource management functionalities, the WMN node stores, and dynamically updates, the context requirements that characterize each VN node, as well as the number of users and flows associated to each VN node.

The WMN node is the first contact point for the user. Thus, it is responsible to enable the association of the user to a fitting VN. To achieve this, the WMN node first stores the context requirements that characterize the user. In the following, it translates and models such requirements in levels to be used by the several architectural control processes.

The WMN node then performs the context matching functionalities to select the best fitting VN to connect the user among the ones that it supports.

If none of these VNs fits the user context, the WMN node inspects the VN knowledge gathered from its physical neighborhood to select the best fitting VN to be extended to this node. In this process, the WMN node also inspects the resource conditions of its neighbor environment in order to select the best WMN path to perform the local VN extension.

If these local VN control processes cannot be performed due to the unavailability of fitting VNs for the user in the neighborhood, the WMN node triggers the global discovery and reconfiguration of a fitting VN to be assigned to the user, or even the possible creation of a new VN to connect a set of users.

Finally, when a VN node is not being used anymore, the WMN node is able to remove it and to free the wireless resources that are assigned.

3.2.3.3 VN Node & VN Controller

As previously referred, each WMN node is split into several VN nodes. The VN node belongs to a specific context-aware VN, and it has configured the specific protocols and mechanisms associated to such VN.

The VN node has the knowledge of its 1-hop virtual neighbors in order to perform data routing functionalities inside the VN. Further, it stores the relevant information of its attached users (e.g., context requirements and perceived QoE). This QoE information is then periodically conveyed in the headers of the control or data VN packets, in order to dynamically select and update the VN routes that provide better QoE support for VN users.

In addition, VN nodes are envisaged to cooperate in order to dynamically select a specific VN node, which we designate as the VN controller, to assist in the global coordination among all VN nodes. The VN controller can be dynamically updated according to VN topology changes and it is responsible to, e.g., configure the same mechanisms and protocols in all VN nodes according to the context purpose of the VN.

3.2.3.4 Global WMN Manager

In order to handle the VN creation, reconfiguration and remotion processes in a distributed way in the WMN, each WMN node has the knowledge of its neighbor WMN nodes, and periodically disseminates the information about its available resources and quality status of its supported VNs. Such context information is then forwarded along a pre-defined maximum number of WMN links, being used to: *(i)* select the best quality fitting VNs to assign to users; *(ii)* perform the resource-aware creation and reconfiguration of VNs in the WMN.

The WMN is also the physical substrate for a multitude of non-interfering VNs. It is then required an entity in the WMN, the global WMN manager, that dynamically allocates and schedules the WMN physical resources (e.g., wireless interfaces and channels) to provide support for the multitude of VNs. The dynamic channel allocation approach will be subject to future work beyond this Thesis.

The global WMN manager stores minimum control knowledge of the VNs available in the WMN in order to assist the distributed control and management of the architecture.

3.2.3.5 Distributed Control Structure

This architecture provides a distributed control structure based on the inter-operability among the VNs characterized by similar context features. This structure aims to enable the fast and global discovery of points of attachment of fitting VNs for users in the WMN.

Based on the periodic exchange of specific knowledge or context among the controllers of these VNs, we build a structured-based control overlay to accelerate the global discovery or adaptation of fitting VNs for users by enabling the context-driven redirection of these processes in the WMN.

3.3 Analytical Model: Impact of Network Virtualization

3.3.1 Goals & Methodology

After describing the main features, challenges and entities of the context-aware architectural paradigm to personalize WMNs, we dive into the assessment of the impact of applying network virtualization to materialize such paradigm.

It is foreseen that network virtualization can enable a programable and flexible sharing of the physical WMN infrastructure by the VNs, and can ease the configuration of the resources and protocols assigned to each VN according to its context purpose. But what are the real gains of the multi-VN architecture for WMNs, which groups user data communications according to the similarity on their context into different context-aware VNs? How is it compared to a traditional WMN environment, where the same communications compete indifferently and under given resource constraints?

In order to seek answers for these questions, and due to the absence of systematized tools and models in the research community to evaluate the impact of network virtualization, this Section proposes an analytical model that covers several aspects of the proposed architecture for WMNs (published in [11] and [16]).

By using probabilistic-based functions to define the mesh router density, mesh client density, user traffic patterns, locality of data traffic and wireless interference, the model derives a closed-form expression for the mean E2E delay of a generic WMN data communication. Then, we adapt such closed-form expression to capture the mean E2E delay of a data communication within a specific VN. Such adaptation needs to take into account the differentiated context-aware requirements of the users and communications supported by each VN, and the proper division of the available WMN resources among the context-aware VNs.

This analytical model aims to demonstrate the limits of the multi-VN architecture to accomplish feasible delays for distinct data communication requirements, under different environmental and load conditions. We will compare these delays against the ones obtained in a traditional WMN environment that provides support for the same set of users and communications. The proposed analytical model will also serve as reference to the evaluation performed in Chapter 4.

3.3.2 Model Definition

In the following, we adapt and extend the analytical work proposed in [279] (which was originally designed to derive the mean E2E delay of a data communication over a generic WMN), to model the mean E2E delay of a context-aware data communication within a specific VN of the proposed architecture.

In [279], a WMN is represented as an open queuing network through the adaptation of the diffusion approximation method [280]. Such method is commonly used to solve an open $G/G/1$ queuing network where all nodes in the network are single servers with infinite buffers and First-Come First-Serve (FCFS) service strategy (please see more details in [280]).

3.3.2.1 Model Preliminaries

Considering a WMN as a square unit area divided into non-overlapping square zones of area a each, we have $N = 1/a$ zones in the WMN, each one covered by a mesh router (please see Fig. 3.4). Each router has an utilization factor of ρ_i and it is equipped with an interface with capacity W_T.

The set of neighbors of router i is defined by $K(i)$. For simplicity, this model neglects the fact that the routers placed near the boundary of the considered WMN area will have a lower number of neighbors than the routers placed at the central WMN zone, which is known as boundary or border effect. Thus, $K(i)$ is considered equal to $K = 8$ in this study, which is the number of neighbors of a central router (please see Fig. 3.4).

There are c mesh clients accessing the WMN, which are uniformly and independently distributed among all mesh routers.

The WMN data packet arrival rate is a renewal process with a mean packet inter-arrival time of $1/\lambda_e$ and Squared Coefficient of Variance (SCV) of C_A^2. Each client may be source or destination of data packets with size S, and generated according to a renewal process with rate λ, which leads to $\lambda_e = c \times \lambda$.

We denote $p_{0i} = a$ as the probability that a data packet enters the WMN through the zone covered by router i. As soon as a packet is generated by a client, it is forwarded to the router within its zone, which will relay it over the backbone to the zone of the target client. Denoting p as the probability of a packet received by a router to be destined to a client within its zone, the probability to forward a packet to router i after completing its service at router j, p_{ij}, is defined by:

$$p_{ij} = \begin{cases} \frac{1-p}{K} & j \in K(i) \\ 0 & otherwise \end{cases}. \tag{3.1}$$

The visit radio of router i, e_i, is defined by the average number of times that a data packet visits the router. Thus, and considering from symmetry $e_i = e_j$, we have:

$$e_i = p_{0i} + \sum_{j=1}^{N} (p_{ji} \times e_j) = a + \sum_{j \in K(i)} \left(\frac{1-p}{K} \times e_j \right) = \frac{a}{p}. \tag{3.2}$$

The data packet arrival rate at router i is now defined by:

$$\lambda_i = \lambda_e \times e_i = c \times \lambda \times \frac{a}{p}. \tag{3.3}$$

Denoting s as the number of routers that forward a data packet before it reaches the destination, we know that $P[s = k] = (1-p)^{[k-1]} \times p$, for $k \geq 1$. This last definition is explained by the fact that, as previously defined, the probability of a packet received by a router to be destined to a client within its zone is p, and so, the probability of a packet received by a router to be forwarded to a neighboring router is $1-p$. Since, by symmetry, the traffic distribution at all routers is the same, if a packet is not absorbed by the clients located under the zone of a router, then all the neighboring routers are equally likely to be the next hop of the packet. Thus, the mean number of hops traversed by a data packet is defined by:

$$\overline{s} = E[s] = \sum_{k=1}^{\infty} \left(k \times (1-p)^{[k-1]} \times p \right) = \frac{1}{p}. \tag{3.4}$$

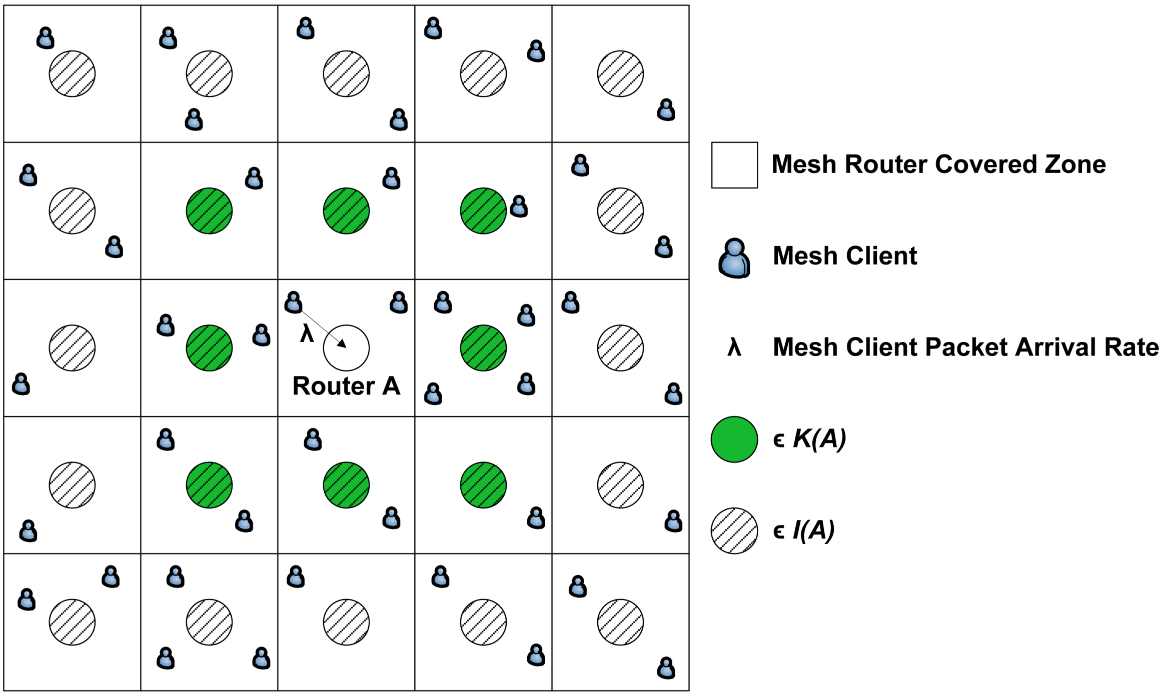


Figure 3.4: Network Model with Squared Zones of Area a.

A transmission made by a router i is successfully performed, if none of its one or two hop neighbors transmits concurrently on the same channel. The set of possible interfering neighbors of router i, $I(i)$, is considered equal to $I = 24$ in this study (again, we neglect the effects of the routers placed near the boundary of the considered WMN area, which are characterized by a lower number of possible interfering neighbors than the routers placed at the central WMN zone).

In order to have an analytical feasible random access MAC model, similar to the IEEE 802.11 DCF MAC model, [279] defines expressions for the: *(i)* number of active interfering neighbors of router i, H_i, which is a sub-set of $I(i)$; *(ii)* duration of the back-off timer of router i, T_i (exponentially distributed with mean $1/\xi$); *(iii)* number of times that the back-off timer of router i freezes during a transmission epoch, M_i, assuming that it is frozen each time that an active interfering neighbor starts transmitting.

We can now define the service time of router i for the transmission of a data packet, X_i. It is the sum of the duration of its back-off timer, the time for which the back-off timer remains frozen, and the data packet transmission time:

$$X_i = T_i + M_i \times \frac{S}{W_T} + \frac{S}{W_T}, \tag{3.5}$$

where the SCV of X_i, C^2_{Bi}, is defined by:

$$C^2_{Bi} = \frac{E[X_i^2] - E[X_i]^2}{E[X_i]^2}. \tag{3.6}$$

3.3.2.2 E2E Delay of a WMN Data Communication

After defining the characteristics of the system, which behaves as an open $G/G/1$ queuing network with N service stations, we will now make use of the diffusion approximation method [280] to derive a closed-form expression for the mean E2E delay of a WMN communication.

Considering the previously defined variables, and according to the diffusion approximation method, the SCV of the inter-arrival time of data packets at router i, C_{Ai}^2, is defined by:

$$C_{Ai}^2 = 1 + \sum_{j=0}^{N} \left((C_{Bj}^2 - 1) \times p_{ji}^2 \times e_j \times e_i^{-1} \right), \tag{3.7}$$

where $C_{B0}^2 = C_A^2$ in order to consider the data packets generated by the clients located within the zone of router i. Again using the diffusion approximation method, the probability that the number of data packets at router i equals k, $\pi_i(k)$, is defined by:

$$\pi_i(k) = \begin{cases} 1 - \rho_i & k = 0 \\ \rho_i \times (1 - \hat{\rho}_i) \times \hat{\rho}_i^{[k-1]} & k > 0 \end{cases}, \tag{3.8}$$

where,

$$\hat{\rho}_i = \exp\left(- \frac{2 \times (1 - \rho_i)}{C_{Ai}^2 \times \rho_i + C_{Bi}^2} \right). \tag{3.9}$$

With (3.8) and (3.9), the mean number of data packets at router i, Z_i, can be defined by:

$$Z_i = \sum_{k=1}^{\infty} (k \times \pi_i(k)) = \frac{\rho_i}{1 - \hat{\rho}_i}. \tag{3.10}$$

At this stage, we have expressions for the mean number of data packets at router i, Z_i (3.10), and for the data packet arrival rate at router i, λ_i (3.3). Hence, and according to Little's law, the mean data packet delay at router i, $\overline{D_i}$, is defined by:

$$\overline{D_i} = \frac{Z_i}{\lambda_i}. \tag{3.11}$$

By symmetry, the mean data packet delay at all routers is the same. Therefore, and using the derived expression for the mean number of hops traversed by a data packet, $\overline{s}$ (3.4), the mean E2E delay of a WMN data communication, D_T, is defined by:

$$D_T = \overline{s} \times \overline{D_i} = \frac{\rho_i}{c \times a \times \lambda \times (1 - \hat{\rho}_i)}. \tag{3.12}$$

3.3.2.3 Context-Aware Characterization of User Communications and VNs

Since our main goal is the provision of personalized VNs for users with specific context requirements, it is required to adapt the expression of D_T (3.12) in order to achieve a closed-form expression for the mean E2E delay of a context-aware data communication within a specific VN. Please note that, in the multi-VN architecture, each VN is only contained in a small set of the N mesh routers and accessed by a small amount of the c mesh clients.

In order to derive the mean data packet delay at each VN node, we need to describe the mean data packet delay at router i, $\overline{D_i}$ (3.11), according to the: *(i)* features of the user communications that are in place in such VN (S, λ, p); *(ii)* bandwidth assigned to the VN links (W) and utilization factor of the VN nodes (ρ) that provide support for such communications. To achieve this, we present below a set of possible rules to map user context features into proper VN features and running services that can be directly applied to the proposed analytical model. These rules will be deeply addressed and extended in Chapter 5.

The typical values for the size, S, and arrival rate, λ, of the data packets of common user services will be used to differentiate the services running at each VN. For instance, VoD or IPTV are characterized by a high size and arrival rate of their packets, whereas VoIP is characterized by small size packets.

The context-aware characterization of the probability of locality of data traffic within a specific VN, p, allows us to differentiate the VNs based on the delay requirements of their supported services. We know that, for instance, delay-sensitive communications are characterized by a high value of p, since it is required to have fast communications with a small number of hops ($\bar{s} = 1/p$). Considering three types of delay constraints, we will have:

$$p = \begin{cases} p_0 & \text{for small delay-sensitive VNs} \\ p_1 & \text{for medium delay-sensitive VNs} \\ p_2 & \text{for high delay-sensitive VNs} \end{cases}, \quad 0 < p_2 < p_1 < p_0 < 1. \tag{3.13}$$

The total bandwidth of mesh links, W_T, needs to be distributed among the available networks of the architecture according to the throughput requirements of the communications that occur over them. These networks not only include the different context-aware VNs, each one supporting distinct services with specific throughput requirements, but also the physical WMN infrastructure, in which the mechanisms and protocols to control the architecture take place. For instance, assuming that the control communications that occur in the physical WMN infrastructure require a small amount of the available WMN resources, and considering three levels for the throughput requirements of user services supported by the available VNs, the bandwidth assigned to the links of the available networks, W, will be defined by:

$$W = \begin{cases} \alpha_c W_T & \text{for physical WMN infrastructure} \\ \alpha_0 W_T & \text{for small throughput-aware VNs} \\ \alpha_1 W_T & \text{for medium throughput-aware VNs} \\ \alpha_2 W_T & \text{for high throughput-aware VNs} \end{cases}, \quad 0 < \alpha_c < \alpha_0 < \alpha_1 < \alpha_2 < 1. \tag{3.14}$$

Finally, the differentiation of the utilization factor of the nodes of each VN, ρ, enables the characterization of each VN according to its security context. In this case, it is obvious that a high secure VN requires higher values of ρ due to the high resource consuming security procedures that are in place in such VN. Thus, we can apply a very similar formulation than the one presented in expression (3.14) to split the capacity of utilization of WMN nodes among the nodes of the available networks of the architecture, but now based on security constraints instead of throughput requirements.

For simplicity, the presented analytical model assumes a single-radio WMN, that is, each WMN node has only a single wireless interface with a single server ($G/G/1$ model). However, this analytical model can be extended to deal with a multi-radio WMN, where WMN nodes may have multiple wireless interfaces (say M) and each one may have its own server ($G/G/M$ model). In such case, each WMN node can work in parallel in more than one wireless interface, and each interface can have assigned a different wireless channel. This requires to analytically assign a pre-defined wireless channel to a specific context-aware VN. It also requires to integrate a scheme to dynamically control the assigned wireless interface (and the respective working channel) to a specific VN at a specific WMN node. Since this thematic will not be addressed in this Thesis, we will leave this for future work.

3.4 Architecture Evaluation

In order to assess the performance, robustness and scalability of several aspects of the multi-VN architecture during the Thesis period, we implemented the proposed architecture and its intrinsic mechanisms in a network simulator, the NS-2.34 [281]. Despite making use of several network functionalities and protocols already implemented in NS-2.34, we also perform several extensions to the native NS-2.34 code, because of its limitations.

In the following, we describe the basic NS-2 functionality, the main important characteristics and limitations of the NS-2 implementation, and the extensions performed to provide the desired architectural functionality (published in [14]).

3.4.1 NS-2 Overview

NS-2 uses an open-source code written in C++, and it has implemented transport layer protocols such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), traffic sources such as File Transfer Protocol (FTP) and Constant Bit Rate (CBR), router queue management mechanisms such as Drop Tail and Random Early Detection (RED), PHY and MAC schemes for the IEEE 802.11 protocol, a variety of routing protocols, etc.

In NS-2, the communications among network components are triggered by an event scheduler. These components communicate among them by transferring packets, and the scheduler emulates the simulation delay required by the components to process and transfer the packets.

The event scheduler and the basic network component objects in the data path are accessible to an object-oriented TcL (OTcl) script interpreter through an OTcl linkage, which creates a matching OTcl object for each C++ object. This linkage allows that the control functions and configurable variables, specified by the C++ objects, to act as member functions and variables of the corresponding OTcl objects.

The NS-2 simulation results are written in trace files that can be analyzed through the use of script-based languages. Each line of these files is produced for an event of each packet, covering all directions from the sender node to the listener node. Additionally, NS-2 can generate files to be processed by a visual interface, the Network Animator (NAM). Finally, there are pre-processing NS-2 tools that allow traffic and topology generation.

3.4.2 NS-2 Implementation: Fundamentals

In NS-2, a basic network implementation requires the use of objects from distinct types of classes, such as application, agent, node or link. The application class allows to model any entity that is capable to receive, request or process data. The agent class enables the implementation of packet generators, timeout functions or packet processing methods that can be used by the control protocols in different layers. With the node class, we can define addressable network entities built from classifiers, which are responsible for distributing incoming data to the corresponding agent and distribute outgoing data to the appropriate link. The link component is a simple connection between two node objects with specified bandwidth and delay characteristics, being also characterized by a queue, a time to live checker and an object that processes link drops.

In the implementation of a NS-2 wireless scenario, which is the focus of this Thesis, the simple node structure also contains a link layer object associated to ARP, a MAC layer with the IEEE 802.11 protocol, a queue between MAC and ARP objects, and finally, a physical object that represents the network interface with the defined antenna and propagation model.

The NS-2 packets are the fundamental unit of exchange among objects of the simulation. These packets derive from the NS-2 packet class, which is a subclass of the NS-2 event class, and they are characterized by a set of headers plus the payload. It is possible to send a new packet or resend an existing one by making use of the NS-2 message class, which allows to configure the fields in the common and IP headers of the packet.

The implementation of a new NS-2 protocol requires the adaptation of the NS-2 agent class. This class was originally proposed to carry a string between two nodes, and so, it has the base methods and code needed to build the desired NS-2 protocol. In the new protocol definition, a variable number of new packet headers can be created by just declaring a new C++ packet structure with the corresponding fields. Depending on the type of the protocol message, some of these headers may not be used because the information required is not the same every time a protocol message is sent; however, the common packet header is always used independently of the protocol.

3.4.3 NS-2 Implementation: Details, Limitations & Extensions

In the following, we detail the key configuration and implementation aspects performed in NS-2, in order to define the desired architectural functionality. Please note that the key architectural entities and their functionalities, that we aim to implement in NS-2, were already presented in Section 3.2.3 and Fig. 3.2.

3.4.3.1 WMN Topology Generation

First, we resort to the OTcl interface in order to declare the number, type of connections (interfaces and channels), and transmission radius of the nodes that form the IEEE 802.11-based WMN topology, which provides support for the proposed architecture.

In most of the simulations performed in this Thesis, we will use a grid-based WMN topology. However, the area and topology of the WMN infrastructure can be configured by resorting to the Boston University Representative Internet Topology generator (BRITE) [282], which is a random topology generator developed for NS-2, or using the position of the nodes of real WMN topologies (e.g., the Funkfeuer Vienna WMN topology [17]).

3.4.3.2 User & Data Traffic Distribution

The OTcl interface then allows us to configure the number and levels of context parameters that drive the multi-VN architecture.

This interface also enables the definition of the type of distributions that characterize the inter-arrival rate and service time of users (or mobile nodes) in the WMN, the position, levels of context requirements and context flexibility of these users, as well as their mobility type and frequency of context change.

Moreover, the OTcl interface is used to define and attach the applications that generate data traffic in the desired mobile nodes.

3.4.3.3 Network Virtualization Emulation

Since NS-2 does not provide any mechanism to support network virtualization, it is emulated using the NS-2 Multi-Interface module [283].

First, each WMN node has assigned a distinct number of IEEE 802.11 wireless interfaces (one for control purposes and the remaining for the support of data traffic), and each of these interfaces has configured its own PHY and MAC layers, and works in a non-interfering and dedicated wireless channel. Second, the physical resources (bandwidth and queue length) of a typical wireless physical interface are distributed among these ”virtual” interfaces, according to proper context-aware rules for VN resource configuration. This way, we emulate the assignment of a specific bandwidth and queue length to each VN node.

Following this strategy, we can dynamically create a different number of VNs in the WMN, by connecting the WMN nodes that contain ”virtual” interfaces working on the same wireless channel. This is achieved through the emulation of two major concerns of network virtualization: resource isolation and resource sharing.

In this emulation process of network virtualization, it is important to note that the documentation to extend the NS-2 architecture to add multiple interfaces in a flexible way, presented in [283], provides instructions to modify the implementation of the methods of the original NS-2 AODV routing protocol. For instance, the AODV routing agent needs now to be endowed with intelligence to dynamically select the interface and channel to forward the traffic associated to each VN. Due to this fact, most of the simulations performed in this Thesis will make use of the AODV routing protocol [36].

3.4.3.4 User Context Storage

When a mobile node arrives at a specific WMN node, it performs the traditional link-layer association already implemented in NS-2. Then, each WMN node has associated a module derived from the native NS-2 agent class. With such module, each WMN node is able to store the context requirements (e.g., delay, security or energy) and the levels of these requirements (e.g., small, medium or high) of its connected users.

3.4.3.5 Context-Aware VN Configuration

The proper configuration of each VN, according to the context requirements of its assigned users, is another key functionality of the special agent attached to each WMN node.

First, each WMN node has associated different methods to enable the assignment and configuration of the resources of each VN node according to the specific VN traffic requirements. This context-aware resource configuration is performed by using and extending the NS-2 IEEE 802.11e module [284] to work with multiple interfaces, which can also be used to assign a distinct priority level to the channel assigned to a specific VN.

Second, each WMN node has to configure the VN control protocols (e.g., security or mobility protocols) according to the context requirements of VN users. However, due to the limitations of NS-2 to allow multiple protocols (of the same network layer) to run in the same NS-2 simulation environment, the special WMN node agent has also the ability to emulate the behavior of a particular network protocol in a specific VN node. For instance, a WMN node can emulate the processing delay and packet header length imposed by the use of a security authentication and/or encryption mechanism associated to a specific VN security context.

3.4.3.6 Control Knowledge Distribution

Beyond the proper context-aware VN configuration, we implement the control mechanisms and messages required to acquire and exchange the necessary information among the distinct architectural entities.

First, each WMN node periodically updates the information about its available resources and quality status of its supported VNs. Then, each WMN node sends this control information to its 1-hop physical neighbors, in order to allow better distributed decisions.

Second, we implement the mechanism to allow VN nodes to periodically collect and disseminate, to its 1-hop virtual neighbors, the control information required to dynamically select the VN routes that provide better QoE support for VN users, or to select and update the location of the VN controller.

Third, the NS-2 Bamboo-DHT [285] is adapted to implement a preliminary version of the distributed and context-aware cooperation among VN controllers, in order to enable the fast and global distributed discovery of points of attachment belonging to fitting VNs for users. However, this scheme will be evolved to enable a more flexible construction and adaptation of the distributed control structure, in order to allow the dynamic insertion, update and remotion of VN controllers according to VN dynamics.

The implementation of the mechanisms to exchange control information among WMN nodes, VN nodes and VNs, makes use of the NS-2 agent class to built the methods required to process the information. Moreover, it resorts to the NS-2 message, packet and timer classes to enable the periodic information exchange among objects of the simulation.

3.4.3.7 User Association & VN Control

Another essential part of the NS-2 implementation is the definition of an overall framework that uses the local or global control information exchanged among the distinct architectural entities, in order to dynamically discover, select, create, adapt and remove VNs on-demand to react to user mobility and context changes.

This framework is an extension of the NS-2 modules built to enable the control knowledge distribution in the network. It includes the implementation of distinct types of messages (e.g., request, response, acknowledge or indication messages) used to coordinate the users, WMN nodes, VN nodes and VN controllers in the overall VN control and management process.

Moreover, this framework contains several context-aware metrics and algorithms that allow each WMN node to: *(i)* select the VN that should be assigned to an user that arrives at a specific WMN node or suddenly changes the context requirements; *(ii)* select the path to perform a VN adaptation in the WMN; *(iii)* trigger a global VN discovery by using the distributed control structure of VN controllers; *(iv)* detect when to start the creation of a new VN in the WMN, or the remotion of unused VN nodes (or VNs).

After the selection and adaptation (or creation) of the VN to be assigned to an user, and the reconfiguration of the WMN nodes that will support such VN, the OTcl interface allows us to define the destination of the user flow within the VN.

In the following Chapters of this Thesis, we will propose and detail the context-aware rules to model and map user context requirements in proper VN features. In addition, we will define the control protocols implemented in NS-2 (including their messages and timers) to gather and distribute the control knowledge along the different architectural entities, and to use such knowledge in favor of a dynamic VN control and management. Finally, we will run a set of NS-2 simulations to validate the results of the analytical models proposed along this Thesis, and to evaluate the effectiveness of the implemented context-aware mechanisms for VN control and management under different environmental and load conditions. The metrics to evaluate these mechanisms will span from: *(i)* signaling control delay and overhead; *(ii)* number of VN reconfigurations to deal with user context change; *(iii)* quality of user data traffic by inspecting QoS metrics; *(iv)* user perceived QoE; *(v)* probability of triggering each control mechanism; etc.

3.5 Evaluation & Discussion

In the following, we evaluate and validate the analytical model, presented in Section 3.3, to study the impact of applying network virtualization on the proposed architecture.

First, Section 3.5.1 uses the proposed analytical model to provide the conceptual boundaries of the architecture, in terms of the number of VNs, attached users and virtual nodes per VN, in order to accomplish feasible delays for different data communication requirements.

Section 3.5.2 then validates the analytical model by comparing its communication delays against the ones obtained when using the simulation platform, described in Section 3.4, for a specific scenario. Such scenario will be driven by the main characteristics of the analytical model, in order to provide a set of results that are directly comparable to the analytical ones.

Finally, in Section 3.5.3, we make use of the simulation framework to assess the influence of using the node positions of a real WMN topology to materialize the multi-VN architecture.

3.5.1 Analytical Boundaries of the Multi-VN Architecture: Data Plane

In this Section, we use the proposed analytical model, defined in Section 3.3, to assess the suitability of network virtualization to personalize mesh client communications. This study is performed by determining the limits of the multi-VN architecture in order to accomplish feasible E2E delays for different data communication requirements.

3.5.1.1 Evaluation Scenario: Details & Variables

We start by defining all the variables and details used in this analytical study, which concerns, not only the specific features of the physical WMN infrastructure and VNs, but also the requirements of the users and communications that are supported by these VNs. Table 3.1 summarizes these variables and details.

The proposed multi-VN architecture is established over a grid-based WMN composed by $N = 10 \times 10$ mesh routers, each one with an interface with maximum capacity of $W_T = 54Mb/s$. The WMN is the physical substrate for $V = \{6, 18\}$ context-aware VNs. Each VN is composed by a mean number of $n = \{6, 6.5, 7, 7.5, 8, 8.5\}$ virtual nodes, which are uniformly distributed over the physical mesh routers. Different values for V and n could be used; however, we consider the aforementioned demonstrative set of values for V and n.

Mesh clients arrive at the WMN at a variable rate, $\lambda_T = \frac{1}{c}$, leading to a mean number of $c = \{80, 90, 100, 110, 120\}$ mesh clients in the WMN. They are equally distributed among the available mesh routers and VNs and each client generates one traffic flow characterized by the features of the VN in which the client is connected to.

Table 3.1: Analytical Model Details: Impact of Network Virtualization.

	Physical WMN Infrastructure	**Context-Aware VNs**		
		Delay-Sensitive	**Secure**	**Throughput-Aware**
N	100 (10 × 10)			
V	6 or 18 VNs in the WMN ($V/3$ per VN Type)			
n	6, 6.5, 7, 7.5, 8 or 8.5 Virtual Nodes per VN			
c (flows)	80, 90, 100, 110 or 120 Flows in the WMN (c/V per VN)			
K / I	8 / 24	$n-1$		
$W(W_T{=}54Mb/s)$	$0.1W_T$	$0.1W_T$	$0.3W_T$	$0.5W_T$
S	–	$64Bytes$	$96Bytes$	$128Bytes$
λ		$17.6Kb/s$	$96Kb/s$	$512Kb/s$
p		0.4	0.3	0.3
ρ		0.4	0.5	0.4

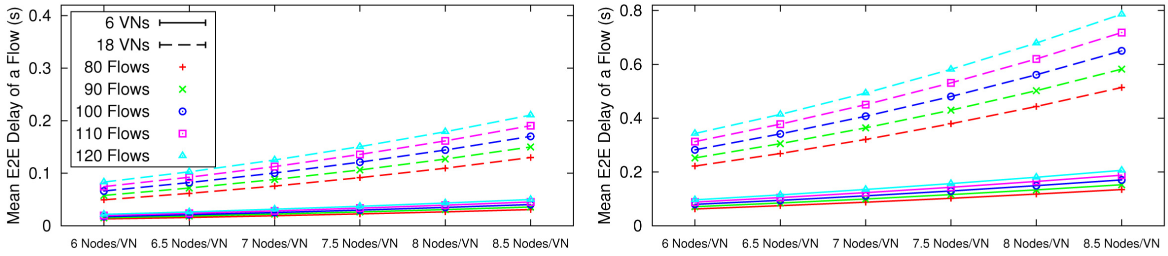
As referred in Section 3.3, it is considered that each mesh router has $K = 8$ neighbors and $I = 24$ possible interfering neighbors. In the VNs, K and I are considered equal to the total number of VN nodes minus the proper node, $n - 1$.

The specific requirements of the data traffic flows running at each VN, as well as the utilization factor and bandwidth of the VN nodes, are defined according to the context purpose of such VN (please refer to the discussion of Section 3.3.2.3). For simplicity reasons, we consider that each VN is mostly specific for one of the three types of communications: delay-sensitive, secure or throughput-aware communications; however, these context-aware VNs may be characterized by other features that have less importance for the VN behavior (e.g., a secure VN may offer low bandwidth for communications, whereas other secure VN may offer a higher level). Following this approach, we can describe the specific features of each VN, which are detailed in the last 5 lines of Table 3.1: *(i)* in a delay-sensitive VN, the data traffic is more located (high probability of locality of data traffic, p) in order to have quick communications with a small number of hops; *(ii)* in secure VNs, the higher utilization factor of their virtual routers (ρ) is the parameter that differentiates their traffic when compared to the other VN types; *(iii)* the VNs for high throughput communications require high capacity (W) of their virtual links, and the communications are characterized by packets with larger size (S) and higher arrival rate (λ) than the delay-sensitive or secure communications; *(iv)* the control processes that occur in the physical WMN infrastructure require a small amount of the available WMN bandwidth.

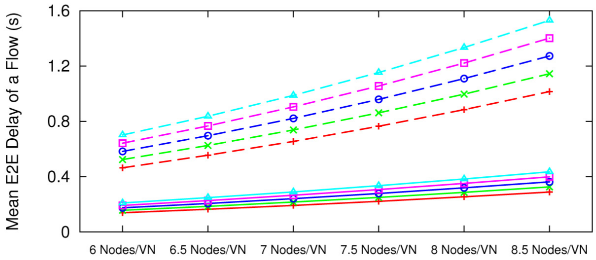
Note that the bandwidth assigned to each VN type refers to the total bandwidth assigned to all VNs from this type, and it is equally divided among the available number of VNs from this type. For simplicity reasons, the total number of VNs, V, is equally divided among the three considered types ($\frac{V}{3}$).

3.5.1.2 Results & Discussion

In this Section, we evaluate the maximum number of: *(i)* VNs that may be established over the WMN, V; *(ii)* virtual nodes that are part of each VN, n; *(iii)* attached mesh clients, c. To perform this study, we run the analytical model in order to know the break-down of the E2E delay of a data communication within each type of context-aware VN according to the setup defined above. Fig. 3.5 shows the mean E2E delay of a data communication, D_T (3.12), within a delay-sensitive VN (a), secure VN (b) and throughput-aware VN (c).



(a) Delay-Sensitive VN (b) Secure VN



(c) Throughput-Aware VN

Figure 3.5: Analytical Model: Mean E2E Delay of a Data Communication within a Delay-Sensitive VN (a), Secure VN (b) and Throughput-Aware VN (c), with a Total Number of 6 or 18 VNs established in a Grid-Based 10×10 WMN, and varying the Number of Nodes per VN and the Number of Users in the WMN.

From the obtained results, we observe that, when there are established 18 VNs in a 10×10 WMN, a VoIP communication, which maximum delay has to be lower than $100 - 200ms$, is only possible when these VNs are composed by a maximum number of 8 virtual nodes. For a higher number of virtual nodes per VN (8.5 in the example of Fig. 3.5), the delays of the communications performed in delay-sensitive VNs start to be higher than $200ms$, mainly when there are a large number of users in the WMN (120 in the example of Fig. 3.5).

On the other hand, we can see that high throughput communications present delays higher than $\approx 1s$, when the WMN provides support for 18 VNs, each one composed by more than 7 virtual nodes, and more than 100 mesh clients access the WMN. A communication delay of $\approx 1s$ can be acceptable for traditional VoD applications. However, for real-time video applications, the communication delays need to be lower than $1s$, which is only possible when there are established 6 VNs in the WMN.

Still from Fig. 3.5, since each VN meets the user requirements through the adaptation of its features to the context-aware purpose of the user communication, the differences between their communication delays are expected: *(i)* delay-sensitive communications present the lowest delays, since their packets flow over a small number of hops; *(ii)* high throughput communications present the highest delays, due to the higher size and arrival rate of their packets; *(iii)* secure communications present intermediate values for the data communication delays, since there is the need for a high processing time of the virtual routers.

The E2E delay of a context-aware communication is higher when there are more VNs, which implies the reduction of assigned physical resources to each VN, and increases the processing time at each virtual node. Moreover, the increasing number of virtual nodes per VN and users in the WMN, increases the E2E data communication delay, because there are

Table 3.2: Modeling & Simulation Details: Impact of Network Virtualization.

	Physical WMN Infrastructure	**Context-Aware VNs**		
		Virtual Slice 0	**Virtual Slice** 1	**Virtual Slice** 2
N	49 ($7 \times 7 \rightarrow 700m \times 700m$ of Simulated Area)			
V	9, 18 or 27 VNs in the WMN ($V/3$ per Virtual Slice)			
n	5 Virtual Nodes per VN			
c (flows)	1, 2, 3 or 4 Flows per VN			
K / I	8 / 24	$n-1$		
Context Features	2 (Throughput and Delay Services' Requirements)			
Context Levels	3 (Small, Medium, High)			
$W(W_T{=}54Mb/s)$	$0.05W_T$	$0.1W_T$	$0.35W_T$	$0.5W_T$
S	–	$64Bytes$	$256Bytes$	$512Bytes$
λ		$11Kb/s$	$45Kb/s$	$180Kb/s$
$\overline{s} = 1/p$		3	4	4
ρ		0.5		

more traffic and users per VN, which increases the intra-VN interference. Finally, we can say that optimized resource management schemes are needed to improve these results, since, for now, it is assigned a fixed bandwidth to each VN.

3.5.2 Model Validation

In order to assess the accuracy of the results provided by the analytical model, this Section resorts to the simulation framework, presented in Section 3.4, to define a simulation setup that aims to compare the mean E2E delays of WMN data communications supported by distinct VNs against the delays derived from the analytical model. This simulation setup also allows to evaluate the delays obtained in a traditional WMN environment without the presence of network virtualization.

3.5.2.1 Evaluation Scenario: Details & Variables

In Table 3.2, we summarize the details of the simulation setup, which is driven by the specific characteristics of the analytical model.

The WMN is characterized by a grid-based topology with $N = 7 \times 7$ mesh routers. We will also perform several simulations with 10×10 mesh routers (which will be presented at the end of this Section), but the NS-2 scalability limitations do not allow us to greatly increase the number of physical nodes and links, and to assess the gains of network virtualization in a larger WMN.

Each mesh router has a transmission radius of $100m$ ($700m \times 700m$ of simulated area). In each simulation, we simultaneously establish 3, 6, or 9 VNs per considered VN type (three types of VNs are considered, as will be described), giving a total number of $V = \{9, 18, 27\}$ VNs in the WMN. Each VN has $n = 5$ virtual nodes uniformly distributed over the physical mesh routers.

The number of users connected to each VN ranges from 1 to 4, and each of these VN users generates one data flow characterized according to the arrival rate (λ), packet size (S) and probability of locality (p) of the context-aware communications supported by such VN. The location of the target of each user flow is totally related to the probability of locality of the traffic supported by the VN in which the user is connected to, as will be described.

As referred in Section 3.3, the analytical model considers that each mesh router has $K = 8$ neighbors and $I = 24$ possible interfering neighbors. In the VNs, K and I are considered equal to $n - 1$.

According to the discussion of Section 3.4.3.3, each mesh router has 4 "virtual" interfaces (one for control purposes, and the remaining for data), each one working on a specific wireless channel. The bandwidth of a typical wireless interface, W_T=54Mb/s, is distributed among these 4 "virtual" interfaces according to proper context-aware rules described below. Therefore, we create 4 virtual slices in the WMN: S_c for control; S_0, S_1 and S_2 for data.

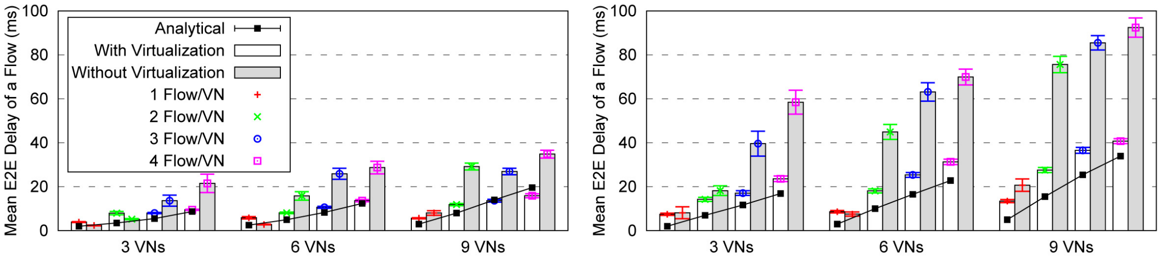
In order to differentiate the characteristics of the context-aware VNs, we consider 2 user context parameters: delay and throughput requirements of user required services. Each parameter can have 3 distinct levels: high, medium or small. Throughout this study, we assume three types of user communications with distinct packet sizes, S, arrival rates, λ, and probability of locality, p, each one supported by a specific VN: *(i)* high delay-sensitive and small throughput-aware; *(ii)* small delay-sensitive and medium throughput-aware; *(iii)* small delay-sensitive and high throughput-aware.

Throughput requirements are differentiated by using three types of CBR traffic, with distinct packet sizes and arrival rates: high) 512$Bytes$ and 180Kb/s; medium) 256$Bytes$ and 45Kb/s; small) 64$Bytes$ and 11Kb/s. Please note that we use lower values for the arrival rate of data traffic, than in the study of Section 3.5.1, in order to not increase too much the traffic generated in NS-2, and thus, decrease the duration of simulations and computational resources required. Since there are three data virtual slices in the WMN, VNs are distributed among them according to distinct throughput requirements. In this sense, VNs supporting communications with small/medium/high throughput requirements are built in $S_0/S_1/S_2$, respectively. In order to properly configure the data virtual slices, we introduce tailored factors to distribute the total WMN bandwidth among them (please refer to expression (3.14)): $S_0/S_1/S_2$ have respectively assigned W=0.1/0.35/0.5×W_T; the remaining bandwidth is assigned to the virtual slice S_c for control purposes.

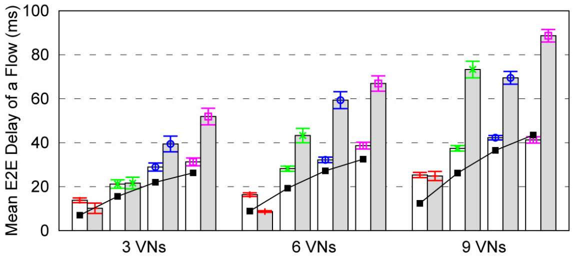
Concerning the characterization of the communications supported by each VN according to the two levels of delay constraints, we randomly select the VN node in which a VN user is connected to. Then, we randomly select the target of the user communication flow, ensuring that the hop distance to the source is lower in case of high delay-sensitive communications. Since our simulated WMN has 7×7 nodes, VNs supporting services with high/small delay constraints are characterized by flows with mean number of hops of $\overline{s} = 3/4$, respectively.

In the simulation platform, VNs are uniformly distributed over the physical mesh routers, and are grouped in three different virtual slices. Each node of a particular virtual slice has assigned a fixed bandwidth derived from expression (3.14), and this bandwidth is then distributed among the VNs running in such node. In our simulations, the bandwidth assigned to each node of a virtual slice is equally distributed among the VNs that such node supports.

To follow the simulation environment, we need now to adapt the analytical model to consider the overall bandwidth assigned to each VN running in a particular slice, instead of the overall bandwidth assigned to each virtual slice obtained through expression (3.14). In our simulation environment, since the VNs are uniformly distributed in the WMN, we can easily infer that, when the number of VNs per virtual slice starts to increase, the probability of these VNs to make use of the same links also increases. Consequently, from the discussion of the last paragraph, the bandwidth assigned to each VN node decreases, being dependent on: *(i)* VNs per virtual slice ($V/3$); *(ii)* virtual links per VN ($l = n - 1$); *(iii)* links of the grid-



(a) Virtual Slice 0 (b) Virtual Slice 1



(c) Virtual Slice 2

Figure 3.6: Analytical & Simulation Models: Mean E2E Delay of a Data Communication within a High Delay-Sensitive and Small Throughput-Aware VN (a), Small Delay-Sensitive and Medium Throughput-Aware VN (b) and Small Delay-Sensitive and High Throughput-Aware VN (c), with or without slicing a Grid-Based 7×7 WMN into VNs with 5 Nodes, and varying the Number of VNs and the Number of Users in the WMN.

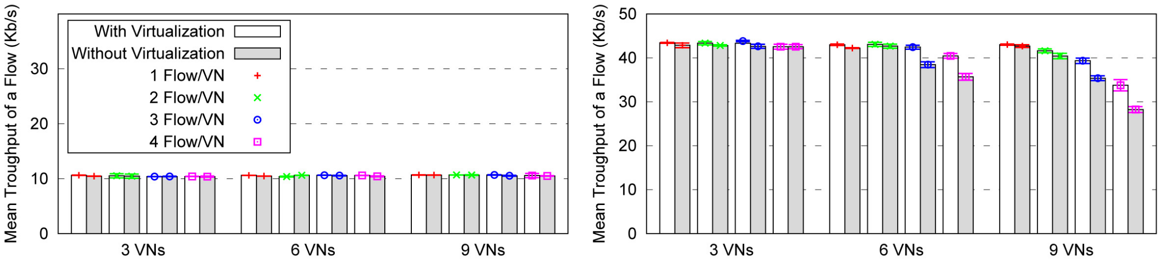
based WMN topology ($L{=}\sqrt{N}{\times}(\sqrt{N}-1){\times}2$). Due to this fact, and based on a probabilistic approach, the bandwidth assigned to each VN of a virtual slice can be approximated by:

$$W_{VN} = W \times \left(1 - \frac{V}{3} \times \frac{l}{L}\right). \tag{3.15}$$

3.5.2.2 Results & Discussion

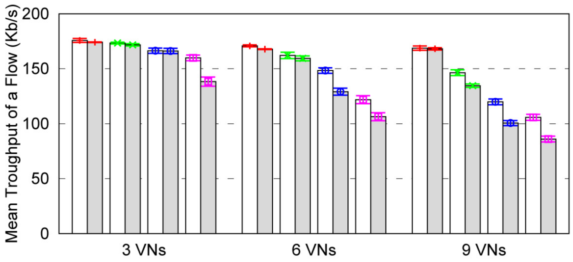
In this Section, we use both the analytical and simulation models (20 simulations per value, with a confidence degree of 90%) to evaluate and compare the mean E2E delay of a data communication, D_T (3.12), supported by the considered context-aware VNs. These results are presented in Fig. 3.6, and they are obtained by varying the number VNs per virtual slice and users per VN. Fig. 3.6 also presents the simulation results with the exactly same setup (same WMN topology and routers, VNs and virtual nodes, users, and types, sources and targets of user flows) but now running in a normal AODV-based WMN, instead of applying the proposed scheme to distribute data flows through VNs that run in isolated virtual slices. Following this strategy, we aim to evaluate the impact of network virtualization to materialize the context-aware WMN paradigm. Finally, and by resorting to the potential of the simulation framework, we show in Fig. 3.7 the mean throughput of a data flow running in a small throughput-aware VN (a), medium throughput-aware VN (b) and high throughput-aware VN (c), and the same results without applying the virtualization scheme.

From the obtained results, we show the benefits of having distinct VNs, each one appropriate to specific levels of delay and throughput requirements of user services. In the



(a) Virtual Slice 1

(b) Virtual Slice 1



(c) Virtual Slice 2

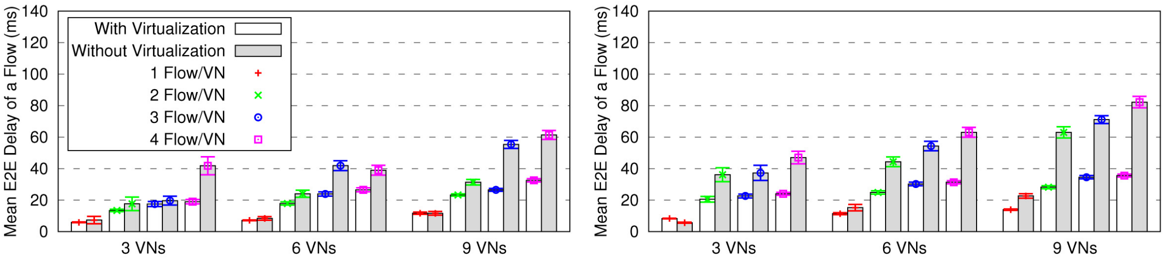
Figure 3.7: Simulation Model: Mean Throughput of a Data Communication within a High Delay-Sensitive and Small Throughput-Aware VN (a), Small Delay-Sensitive and Medium Throughput-Aware VN (b) and Small Delay-Sensitive and High Throughput-Aware VN (c), with or without slicing a Grid-Based 7 × 7 WMN into VNs with 5 Nodes, and varying the Number of VNs and the Number of Users in the WMN.

one hand, high delay-sensitive communications, which are supported by VNs running in S_0, present the lowest delays. On the other hand, S_1/S_2 present higher delays. The delays of the high throughput-aware communications supported by S_2 are slightly higher, due to the higher size and arrival rate of their packets. Fig. 3.7 also shows the effectiveness on the distribution of the available WMN bandwidth among the virtual slices ($S_0/S_1/S_2$), since the high throughput-aware communications achieve the highest throughput.

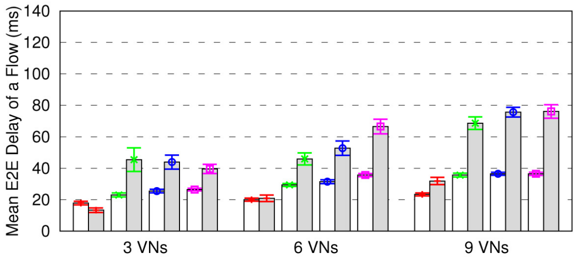
The increasing number of VNs (and flows per VN) introduces more traffic in the WMN, increasing the interference among different flows, which results in higher delays and lower throughput. The results show that network virtualization has a stronger impact when the WMN starts to be overloaded. Since network virtualization enables a higher isolation among distinct communications, its benefits are more notorious when the WMN has a large amount of data traffic (more interference). On the other hand, if there is only one flow per VN (lower interference), the results are usually better when no network virtualization scheme is applied. In this case, the static bandwidth sharing among VNs has no meaningfully advantages, since most of the bandwidth assigned to a particular VN will be wasted, and could be used in a more effective way.

Comparing the analytical and simulation results, we can clearly state that the results of the analytical approach and the NS-2 simulation environment are very similar. However, the delays of the analytical model are slightly lower, since it does not consider the limitations on buffer size and processing power of network nodes. On the other hand, the simulation framework has to deal with more protocols of the NS-2 stack, which increases the delays.

As previously referred, we also evaluated the mean E2E delay of a data communication



(a) Virtual Slice 0 (b) Virtual Slice 1



(c) Virtual Slice 2

Figure 3.8: Simulation Model: Mean E2E Delay of a Data Communication within a High Delay-Sensitive and Small Throughput-Aware VN (a), Small Delay-Sensitive and Medium Throughput-Aware VN (b) and Small Delay-Sensitive and High Throughput-Aware VN (c), with or without slicing a Grid-Based 10×10 WMN into VNs with 5 Nodes, and varying the Number of VNs and the Number of Users in the WMN.

supported by the considered context-aware VNs, or by a traditional AODV-based WMN without the presence of network virtualization, when using a grid-based WMN with 10×10 mesh routers. The results of these simulations are presented in Fig. 3.8, being performed 20 simulations per value, with a confidence degree of 90%. In these simulations, we increase the number of WMN routers and the mean number of hops of data communications (in one unit), thus reducing the probability of locality of data traffic, p. Apart from these two changes, the results of Fig. 3.8 are obtained by resorting to the simulation setup defined in Table 3.2.

The delays of Fig. 3.8 are slightly lower than the ones of Fig. 3.6. For instance, it is observed a mean delay of $38ms$ for the data communications performed in S_2, when there were 9 VNs per virtual slice and 4 flows per VN; for this specific case, Fig. 3.6 presents a delay higher than $40ms$. Such was expected, since in these simulations, the number of WMN nodes is higher, but the number of VNs, nodes per VN and users accessing the WMN is the same. Therefore, due to the random distribution of VNs in the WMN, there is a higher probability of these VNs to make use of non-congested WMN links, which reduces the wireless interference, and consequently leads to the decrease of the data communication delays. However, both the results of Fig. 3.8 and Fig. 3.6 present similar relative tendencies of delay increasing when varying the number of VNs per virtual slice and flows per VN.

3.5.3 Influence of WMN Topology

In this Section, as a proof of concept of our work, we perform simulations with the setup defined in Table 3.2, but using the node locations of the real WMN of Funkfeuer Vienna [17].

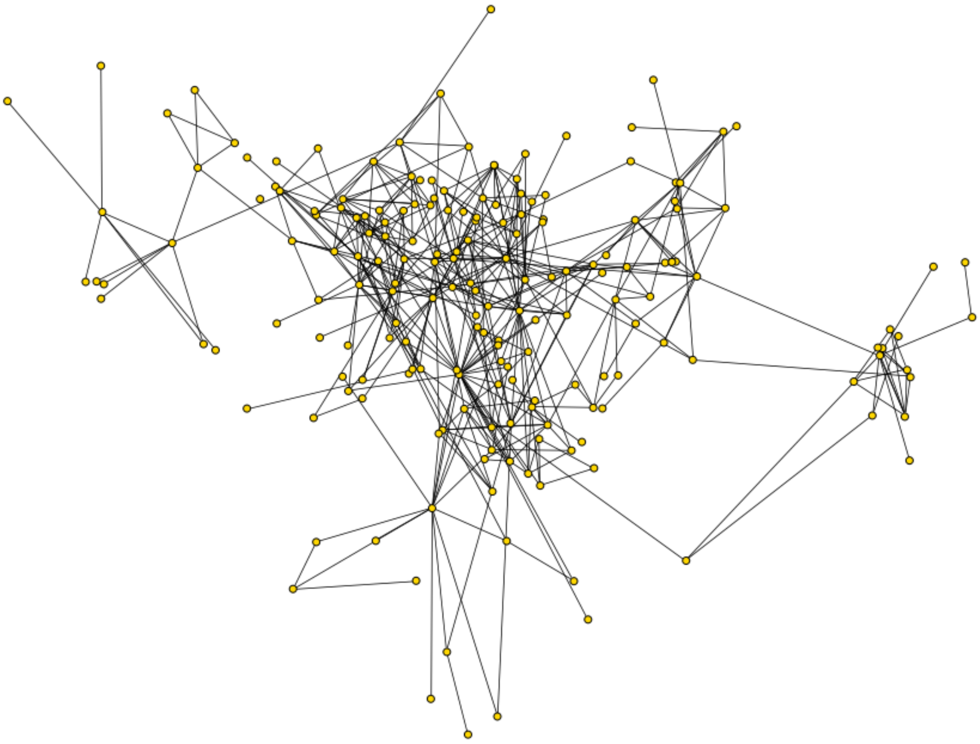


Figure 3.9: Funkfeuer Vienna WMN Topology (Network Partitions are removed).

3.5.3.1 Funkfeuer Vienna

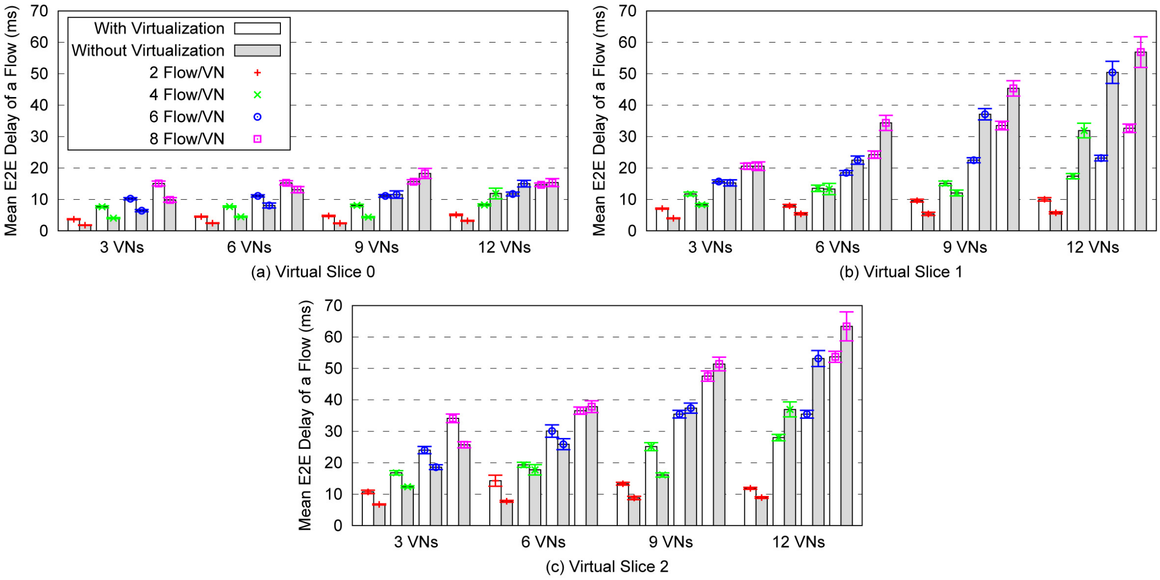
Funkfeuer is a non-commercial, free-of-charge experimental WMN operated by (private) volunteering peers. It is deployed independently (not interconnected) in several Austrian areas like Vienna, Graz, Bad Ischl and Weinviertel, and was started in 2005 in Vienna. The only requirement to join the WMN as a router is to obtain a static IP address from the Funkfeuer organizers. There are no restrictions regarding location and intended purpose (except special contents) of the router.

The Funkfeuer Vienna WMN consists of several wireless routers run by individuals which are interconnected with omni- and directional antennas using IEEE 802.11g and IEEE 802.11n standards in ad hoc mode. The employed router hardware is very diverse and ranges from stand-alone routers to Linux servers. Funkfeuer Vienna is mainly used for Internet access, which is provided through one centralized gateway. Besides Internet access, Secure SHell (SSH) tunneling, VoIP and TV channel streaming services are also offered.

Fig. 3.9 depicts the topology of the Funkfeuer Vienna WMN, which includes a total number of 211 routers and 701 links deployed within a city area of $414.89 km^2$. Routers without connection to the core of the network have been removed as they are partitioned/isolated networks.

3.5.3.2 Results & Discussion

We have accessed to the coordinates of the node locations of the Funkfeuer Vienna WMN and translated them to a NS-2 file. Then, we adapt the NS-2 simulation environment to be



(a) Virtual Slice 0

(b) Virtual Slice 1

(c) Virtual Slice 2

Figure 3.10: Simulation Model: Mean E2E Delay of a Data Communication within a High Delay-Sensitive and Small Throughput-Aware VN (a), Small Delay-Sensitive and Medium Throughput-Aware VN (b) and Small Delay-Sensitive and High Throughput-Aware VN (c), with or without slicing a WMN into VNs with 5 Nodes, varying the Number of VNs and the Number of Users in the WMN, and using the topology of the Funkfeuer Vienna WMN.

able to run with such topology, which includes the configuration of the number of links of each WMN node. In the following, we run distinct simulations (20 simulations per value, with a confidence degree of 90%), when there are 3, 6, 9 or 12 VNs per virtual slice (S_0, S_1, S_2), each one composed by $n = 5$ virtual nodes uniformly distributed over the physical mesh routers. Moreover, 2, 4, 6 or 8 users can simultaneously access each VN. The characteristics of the different virtual slices, the resources assigned to the context-aware VNs running at each virtual slice, and the details of the communications established over each VN are equal to the ones described in Section 3.5.2.1 and Table 3.2.

Fig. 3.10 presents the mean E2E delay of the distinct context-based communications supported by the available VNs in the WMN, when varying the number of VNs per virtual slice, and flows per VN. Such figure also demonstrates the delays of the same data communications, when running over a AODV-based WMN, without the presence of network virtualization.

The delays of Fig. 3.10 are lower than the ones of Fig. 3.6 and Fig. 3.8. For instance, when there were 9 VNs per virtual slice and 4 flows per VN, the mean delay of each data communication performed in S_2 is $\approx 20ms$, which is lower than the delay of $\approx 40ms$ obtained in the simulations of Section 3.5.2. Such was expected, since these simulations make use of the node locations of the real WMN of Funkfeuer Vienna, which is composed by a higher number of nodes than the WMN used in the simulations of Section 3.5.2, and they considered the same number of VNs established in the WMN, each one composed again by a mean number of 5 nodes. Due to the random distribution of VNs in the WMN, there is then a higher probability of VNs to be built over non-overloaded WMN links. This fact reduces the wireless interference and the data communication delays.

Finally, from the obtained delay performance results, we can conclude that our approach can be used to virtualize context-aware WMN topologies in real environments.

3.6 Conclusions

In this Chapter, we started by motivating the need to define new architectures and control mechanisms to cope with the demanding requirements of mobile users, as well as of their devices and required services, which we have denominated by user context. In this scope, we defined a context-aware architecture to be applied to the high flexible and self-adaptable WMNs by resorting to the programmability and isolation enabled by network virtualization. In this architecture, we quantify user context requirements in levels, and then use such levels to dynamically split a WMN into several adaptable, usable and scalable VNs. Each VN is specialized to meet a specific set of user context requirements and to cope with the change of these requirements. This way, users are grouped according to similarity on their context needs and assigned to VNs matching these needs, which can open the door to a new vision of user-centric personalized wireless mesh networking. In this architecture, network virtualization introduces the programmability needed to dynamically share the same physical WMN infrastructure among the multiple VNs, and the isolation required to instantiate different control operations and protocols at each VN according to its context purpose.

In order to ensure that diverse users, devices and services are served with a high quality level and to guarantee an effective WMN resource usage, this context-aware and virtual paradigm for WMNs yields a set of challenges that must be addressed to provide a consistent and flexible solution for the context management and VN control. First, rules need to be designed to quantify, model and map user context needs into proper VN features. Second, mechanisms are required to quickly discover and adapt suitable VNs to connect users, due to the constant context change and mobility of users. Third, the control of the architecture needs to take into account certain levels of user context flexibility to increase the usability of the VNs, which can reduce the probability and the expenditures of creating new VNs to serve single users. Fourth, the processes to select fitting VNs to be assigned to users, and to choose the WMN paths to extend such VNs up to the WMN nodes where the users are located, have to meet a trade-off among the user context needs, the conditions of the candidate VNs and the WMN resource availability. Finally, the control knowledge and functionalities need to be distributed along the several architectural entities (WMN nodes, VN nodes and VNs), which can then cooperate to autonomously perform the discovery, creation, configuration, adaptation and remotion of VNs on-demand to react to any type of context change. The aim of this last requirement is related to the increase of scalability in the architectural control. In this Chapter, we described the main guidelines of the VN control and management mechanisms that are under the vision of the context-aware paradigm for WMNs, and detailed the key architectural entities, along with their key functionalities and interactions, to provide support for the proposed paradigm.

We also discussed the potential of network virtualization to split a WMN into a set of logical networks, each one grouping users and communications according to similarity on their context. Since the effort to show the potential of network virtualization cannot be simply conceptual, we proposed a feasible analytical model that covers the key aspects of the multi-VN architecture. Such model enabled the evaluation of the mean E2E delay of a data communication performed within a specific context-aware VN, supported by a specific set of WMN nodes, and where a specific number of mesh clients are connected to. This model was a first step for the understanding of the limits of the proposed architecture, in terms of the maximum number of VNs, nodes per VN and users in the WMN, such that the data communication requirements are indeed met with reasonable delays.

The work presented throughout this Chapter also resorted to a simulation platform. The foremost objective of this platform is to properly implement and test the proposed architecture and its mechanisms during the Thesis period, so that it can provide a set of simulation results that goes beyond the mathematical proof. In this Chapter, we detailed the key configuration and implementation aspects performed in the used network simulator, the NS-2. This simulation platform also allowed us to validate the analytical model described in the last paragraph, since both analytical and simulation studies presented very similar results for a specific setup. We concluded that network virtualization has a major impact when the WMN is accessed by a large number of users and flows, since it can greatly reduce the interference among such flows, and consequently decrease their E2E delays. However, the findings of such simulations also led to conclude that a major requirement is to produce, in the future, a more sophisticated resource management scheme to: *(i)* select the best WMN paths to built the VNs, thus reducing the probability of congested WMN links; *(ii)* dynamically configure the physical resources assigned to each VN instead of having a fixed bandwidth assigned to each VN link. Through the simulations performed with a real WMN topology, we also concluded that the proposed multi-VN architecture can indeed be used in real running networks.

In summary, this Chapter presented an user-centric context-aware architecture to be applied to WMNs, and evaluated the impact of network virtualization to materialize such architecture through the analysis of its impact on the delay and throughout of several context-aware communications in specific WMN scenarios. Now, it is important to move into the definition and evaluation of concrete mechanisms to distribute the control knowledge and intelligence among the distinct architectural entities, enabling their context-aware and distributed cooperation to autonomously discover, select, create, configure, adapt and remove VNs on-demand, which will be addressed in the following three Chapters.

Chapter 4

Distributed VN Discovery & Extension

After presenting and modeling the context-aware WMN architecture built through network virtualization, in the previous Chapter, we aim to start assessing the potential of a distributed solution for VN control and management. Therefore, this Chapter presents a control mechanism to associate users to fitting VNs that can be available in the WMN nodes where users are connected to, or in their physical neighborhood, thus requiring the smooth reconfiguration of such VNs. Further, this control mechanism also encompasses a Chord-based distributed solution to globally discover fitting VNs to assign to users, and to extend such VNs up to the nodes where users are connected to. The proposed distributed VN control mechanism is analytically evaluated by resorting to the model defined in the previous Chapter, and its results are compared against the ones obtained through a simulation platform.

4.1 Introduction

So far, the splitting of a WMN in a set of context-aware logical networks, through network virtualization, has not been addressed in the literature. This architectural paradigm, proposed in this Thesis, introduces a virtual layer in the entire network stack. This layer is composed by a set of VNs that are built over the physical WMN infrastructure, and each VN is configured to meet a particular set of user context requirements. Therefore, when an user arrives at a specific WMN node, it is important to automatically assign the user to a VN that fits his/her context requirements. Such VN can be available in the WMN node where the user is connected to. However, if the VN is only available in any of the nodes of the user neighborhood, or in other part of the WMN, it is required to extend such VN up to the user location.

This way, each WMN node has to store and access the information about the location of the distinct VNs available in the WMN. Then, we need to define the mechanism to discover, select and adapt a VN to assign to an user that arrives at a specific WMN node (the user can also move inside the WMN, or can suddenly change his/her context requirements). This mechanism has to include: *(i)* a metric to select a fitting context-aware VN to assign to the user; *(ii)* the processes required to trigger the smooth reconfiguration of a fitting VN, available in the user neighborhood or in other part of the WMN, to be included in the WMN node where the user is connected to. Both these mechanisms are envisaged to improve the VN lookup, discovery and reconfiguration processes in order to enable the fast user association to a fitting VN, thus increasing the usability of VNs. Due to the flexible and self-* properties of the WMN infrastructure and protocols, and to break the centralized solutions for WMN and virtualization control, we define and evaluate these mechanisms in the light of the distributed and autonomous cooperation among WMN nodes and the other architectural entities.

Given the above discussion, Section 4.2 proposes a distributed control mechanism to associate users to fitting VNs available in the WMN. First, each WMN node has the knowledge of the VNs supported by itself or by its 1-hop neighbors. With this VN knowledge, each WMN node is able to locally select the best fitting VN to quickly assign to their users, or to quickly adapt the connections of these users to better fitting VNs in case of context change. These local processes can then lead to the smooth reconfiguration of VNs available in the physical neighborhood of the nodes where the users are connected to. In case of unavailability of fitting VNs for users in their physical neighborhood, it is started a distributed VN discovery in the WMN. Such global VN discovery makes use of the dynamic and self-organized features of the Chord-based mechanism [260], enhancing it to be aware of the context features that characterize the VNs. After finding a point of attachment of a fitting VN to assign to an user, such VN needs to be extended up to the user location, which is performed through the cooperation among the WMN nodes that provide support for such VN extension.

By using and extending the analytical model defined in Section 3.3 for data communications, Section 4.3 derives closed-form expressions for the mean E2E delays of the proposed distributed control mechanism to associate users to fitting VNs. Using such analytical model for control communications, Section 4.4 evaluates the delays for user association and VN control, when varying the number of VNs, nodes per VN and users accessing the WMN. Moreover, Section 4.4 extends the simulation framework presented in Section 3.4, ir order to implement the distributed VN control mechanism for user association. Then, this Section defines a specific simulation setup to validate the analytical model, and to evaluate the distributed VN control mechanism under the influence of mobility of mesh clients.

This distributed mechanism for VN control does not consider yet the resource availability of possible WMN paths to perform VN extensions, and it does not define the processes for VN creation or remotion; this will be subject of the following Chapters. However, through the definition and evaluation of such mechanism, we are able to detect the: *(i)* potentials of distributing the control functionalities and enabling the cooperation among WMN nodes; *(ii)* issues that need to be addressed in a more general distributed mechanism for VN control. Thus, Section 4.5 ends the Chapter by summarizing its main conclusions and providing some guidelines that will be addressed in the following two Chapters.

4.2 Distributed VN Control & Management

In traditional WMN environments, users are constantly arriving, accessing and leaving the WMN, and they are frequently changing their context requirements. In addition to this WMN dynamics, our context-aware multi-VN architecture further introduces more levels of dynamics. In such virtualized environment, each WMN node can support several VNs, each one representing a specific set of context features, which can change at a significant rate.

Due to the user and context dynamics, VNs need to be adapted accordingly. We need then to define intelligent and scalable mechanisms for VN control and management. These mechanisms can include the fast discovery and selection of the best fitting VN to assign to an user, or to extend such VN up to a different WMN node when the user moves, or even to move the user from a VN to a better fitting one in case of context change.

These VN control and management mechanisms can be supported by a centralized approach, where a single element controls the entire architecture. However, centralized approaches have well-known limitations, such as processing bottlenecks, required high capabilities of a central server and limited scalability. Therefore, we define a distributed approach to autonomously discover, control and manage VNs without constantly notifying a central WMN controller.

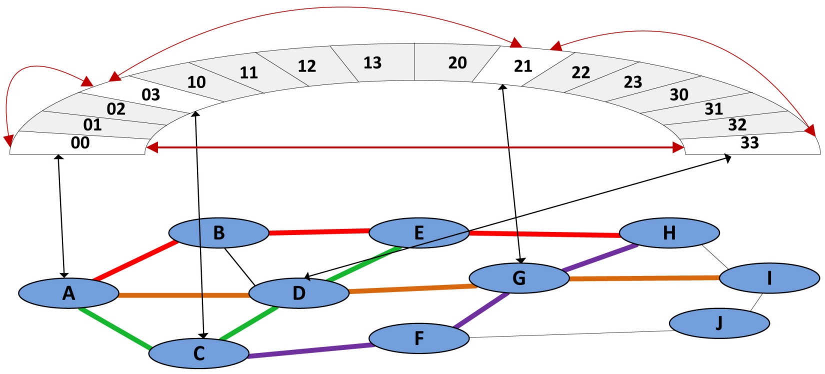
In such distributed approach, each WMN node is aware of the nodes that compose its 1-hop physical neighborhood. With such information, each WMN node can access to the VNs that it supports, or to the ones supported by its neighbor environment. Then, it can quickly select one of its supported VNs to assign to their users, or it can trigger the reconfiguration of VNs available in its physical neighborhood to be included on it.

Moreover, in our multi-VN approach, each VN is characterized by an isolated structure, and it can be instantiated by a specific VN provider or network operator with its own control authority or administration. Thus, the distributed cooperation among distinct VNs can be very useful to accelerate the global discovery of specific VNs in the WMN. In the following, we define a mechanism to enable the global context-aware coordination among the multitude of VNs (this work was published in [12] and [16]).

4.2.1 Distributed Context-Aware Ring

We define a distributed solution for the scalable search and discovery of specific VNs and virtual nodes in the WMN. This distributed solution can be thought and designed in a unstructured or structured manner.

An unstructured-based distributed solution [254] uses flooding as the mechanism to send VN queries across the WMN. This flooding mechanism can enable small discovery delays if the VNs are composed by a large amount of virtual nodes that are spread over the entire WMN, which can introduce a high level of replication of virtual nodes that belong to the



	Cost	Security
VN 1	0	0
VN 2	0	3
VN 3	2	1
VN 4	3	3

Figure 4.1: DHT-Based Context-Aware Ring.

same VN. Nevertheless, since the mesh clients can request a very specific VN in terms of context features, and the VNs can be located in a very specific WMN area to serve only a subset of mesh clients, an unstructured-based solution will certainly introduce a high level of overhead in the global VN discovery mechanism.

This way, we proposed a distributed structured-based ring (or overlay) based on context-aware Distributed Hash Tables (DHTs) [254], which is the most well-known structured-based solution available in the literature. The key-based routing features of DHTs are appellative to allow the fast discovery of specific VNs in the WMN, since we can embed context information in the characterization of the keys. An example of this distributed control structure, which concept was introduced in this Thesis in Section 3.2.3.5, is depicted in Fig. 4.1.

In such example, we consider that a WMN, with 10 nodes, provides support for 4 VNs: VN$_1$, VN$_2$, VN$_3$ and VN$_4$. Each VN is configured to meet two context parameters (cost and security), and each parameter can have four levels ($-- \Rightarrow 0$ or $'00'$, $- \Rightarrow 1$ or $'01'$, $+ \Rightarrow 2$ or $'10'$, $++ \Rightarrow 3$ or $'11'$, where 3 corresponds to a very high level of context). Moreover, each VN is identified according to its context purpose: the first two bits of the VN identifier describe the cost level of the VN, and the last two bits are related to the VN security level.

To build a distributed structured-based ring, driven by the context information that characterizes the VNs available in the WMN, we interconnect the VNs in a consecutive order of VN identifiers, i.e., in a consecutive order of the context parameters and levels that characterize the VNs. According to this ring organization, we can see in Fig. 4.1 that the VNs characterized by the same level of cost context, which is the most significant context parameter in the VN characterization, are located in the same zone of the ring, and so, they are semantically closer in the ring. Therefore, the main important requirements to build, characterize and organize the elements that compose the ring are:

- Within each VN, it is selected one virtual node (the VN controller, as it was introduced in this Thesis in Section 3.2.3.3) to assist in the global coordination among all VN nodes, and to perform the key control and management functionalities inside the VN. In Fig. 4.1, node A is the selected controller of VN$_1$, C of VN$_2$, G of VN$_3$ and D of VN$_4$;

- The set of controllers of the VNs available in the WMN also are the elements that compose the distributed structured-based ring;

- The key that identifies each VN controller in the ring, embeds the user context parameters, and the levels of these parameters, used to characterize such VN;

- The VN controllers are interconnected in a consecutive context-aware order, and there are also established several shortcuts among them. This ring connectivity is required to enable the fast and context-driven redirection of a global VN discovery to a zone of the ring that is semantically closer to possible fitting VNs to assign to users;

- Each VN controller needs to be constantly updated, since VNs are dynamically created, adapted or removed to adapt to context change.

According to the description presented above, we define the basis of a framework for the global distributed and autonomous discovery of points of attachment of fitting VNs to assign to users. To materialize such framework, the work of this Chapter assumes that: *(i)* it is randomly selected a virtual node from each VN to be its controller, which has to be updated in case of VN adaptation due to context change; *(ii)* Chord-DHT [260] is used to establish the links and shortcuts among the VN controllers, enabling their inter-operability to perform the global discovery of a specific VN in a DHT-alike way.

Later in Chapter 6, we will better detail and enhance the mechanisms of the structured-based distributed control ring in order to: *(i)* insert and remove new VN controllers in the ring; *(ii)* dynamically update the location of each VN controller to improve the performance of the control and management functionalities inside the VN; *(iii)* perform a better selection of the links and shortcuts that are established among consecutive VN controllers in the ring, as a mean to avoid topological mismatching problems between the physical WMN infrastructure and the proposed control overlay; *(iv)* try to reduce the complexity of maintaining a Chord-DHT, alleviating the level of control signaling required to maintain the links and shortcuts among VN controllers; *(v)* improve the distributed global VN discovery and selection by enabling the exchange of specific knowledge among the controllers of VNs that are most probable to deal with user context requirements and their variations.

4.2.2 Distributed VN Discovery & Extension: Scenarios & Signaling

This Section proposes three control scenarios to associate an user that arrives at a specific WMN node, to a fitting VN available in the WMN. These scenarios are summarized in Algorithm 1 and they respectively detailed in Fig. 4.2, Fig. 4.3 and Fig. 4.4.

Algorithm 1 may be performed in the WMN node where the user is connected to, or in its physical neighborhood, or it may even require the involvement of more WMN nodes in the global discovery and extension of a fitting VN to assign to the user. Following this way, we aim to have a first notion of the impact of distributing the control knowledge and intelligence among the WMN nodes; they will cooperate to locally or globally perform the discovery of fitting VNs to assign to users, and the extension of such VNs up to the nodes where users are connected to, without the need to constantly notify and update a central entity.

4.2.2.1 Assumptions

In this Chapter, we aim to assess the impact of different scopes of decision (local or global) for the discovery and reconfiguration of VNs to assign to mesh clients. Therefore, for simplicity, the work of this Chapter assumes that mesh clients are connected to the VNs that

Algorithm 1: Distributed VN Discovery & Extension

```
   $u$ : User accessing the WMN;
   $n_1$ : Mesh router where the user $u$ is connected to;
   $v_u$ : Fitting VN to assign to the user $u$;
   $PoA_{v_u}$ : Point of attachment of the VN $v_u$;
   $P_{n_1}$ : Physical neighborhood of the node $n_1$;
   $n_2$ : Physical neighbor of the node $n_1$ that supports the VN $v_u$;

   /* Initialization Phase                                            */
1  $u$ sends context requirements to $n_1$;
   /* VN Discovery & Extension Phases                                 */
2  if ($PoA_{v_u} \subset n_1$) then                     /* Scenario 1 */
3      break;
4  else
5      $n_1$ asks $P_{n_1}$ to know their current VNs;
6      $P_{n_1}$ answer $n_1$;
7      if ($PoA_{v_u} \subset P_{n_1}$) then             /* Scenario 2 */
8          Create a virtual link between $n_1$ and $n_2$;
9      else                                              /* Scenario 3 */
10         Global discovery mechanism to find $PoA_{v_u}$;
11         Create a virtual connection between $n_1$ and a WMN node supporting $v_u$;

   /* Conclusion Phase                                                */
12 Notify $u$;
```

exactly fit their context requirements. In Chapter 5, we will propose and evaluate a novel context-aware VN selection metric that takes into account the user context flexibility.

We also consider that there is a distributed entity, which was introduced in this Thesis in Section 3.2.3.4, the global WMN manager. This entity is able to: *(i)* update the resources that need to be assigned to each VN node; *(ii)* select the best WMN paths to perform VN adaptations through the cooperation of the WMN nodes that can provide the physical substrate for the new VN connections. In this Chapter, we assume that a VN extension is performed through the available shortest path between the user location and the nearest point of attachment of the exactly fitting VN. In Chapter 6, we will evolve this mechanism to be aware of the WMN resource availability in the possible paths to perform the VN extension.

We define a set of variables that help to understand Algorithm 1 and the three control scenarios described below. In our description, we consider that the user u arrives at the WMN node n_1, and wants to be connected in the VN that exactly fits the context requirements of u, v_u. A point of attachment of the VN v_u, PoA_{v_u}, has then to be found in order to trigger the extension of v_u to be included in n_1. PoA_{v_u} can be accessible in: *(i)* the WMN node where the user is connected to, n_1; *(ii)* the WMN node n_2 that belongs to the 1-hop physical neighborhood of n_1, which is defined by P_{n_1}; *(iii)* any other node of the WMN, which has to be discovered by resorting to the distributed structured-based ring already presented.

4.2.2.2 Scenario 1: VN Update

When the user u arrives at the WMN node n_1, user u establishes the link-layer connection, and then, user u sends his/her context requirements to node n_1 (Alg.1-line1). This process

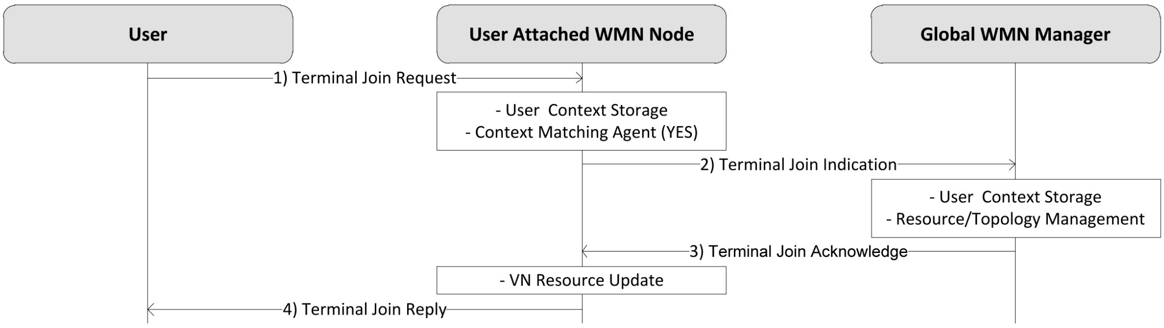


Figure 4.2: Scenario 1: VN Update.

is performed by sending a control message that includes the context levels required by user u (Fig. 4.2→1).

In the following, node n_1 stores the context requirements of user u, and matches them against the features of the VNs that node n_1 supports. This matching involves the direct comparison among the levels of the context features provided by the VNs supported by node n_1, and the levels of the context parameters required by user u.

If the VN v_u is already available in node n_1 (Alg.1-line2), node n_1 notifies the global WMN manager that a new user will be connected to its virtual node from the VN v_u. This notification is described in Fig. 4.2→2, and the used control message also includes the context of user u.

The global WMN manager stores the context of user u, and performs the resource management functionalities to update the resources that node n_1 has to assign to its virtual node from the VN v_u. This new resource information is then conveyed in the message sent by the global WMN manager to node n_1 (Fig. 4.2→3). Finally, user u is connected to the VN v_u (Alg.1-line12 and Fig. 4.2→4).

4.2.2.3 Scenario 2: Local VN Extension

If the VN v_u is not available in the WMN node n_1, this means that the user u is connected to a WMN node that does not contain the VN v_u. It is then required to find the closest WMN node that has a virtual node belonging to the VN v_u.

To this purpose, node n_1 triggers a local VN discovery mechanism in its 1-hop physical neighborhood (Alg.1-line5). This local mechanism can enable the fast detection of any virtual node from the VN v_u, and it is important to convey the context of user u in the VN discovery request (Fig. 4.3→2). In the following, the physical neighbors of node n_1 compare the features of the VNs that they support against the context of user u, and send a reply to node n_1 (Alg.1-line6 and Fig. 4.3→3).

After receiving the replies from its physical neighbors, node n_1 can detect that the VN v_u is available in one of these neighbors (Alg.1-line7). In this case, node n_1 notifies the global WMN manager that the VN v_u, available in one of its 1-hop physical neighbors (e.g., node n_2), has to incorporate node n_1. This notification is described in Fig. 4.3→4, and the used control message also includes the context of user u.

The global WMN manager stores the context of user u, and performs the topology and resource management functionalities to update the topology of the VN v_u, and the resources

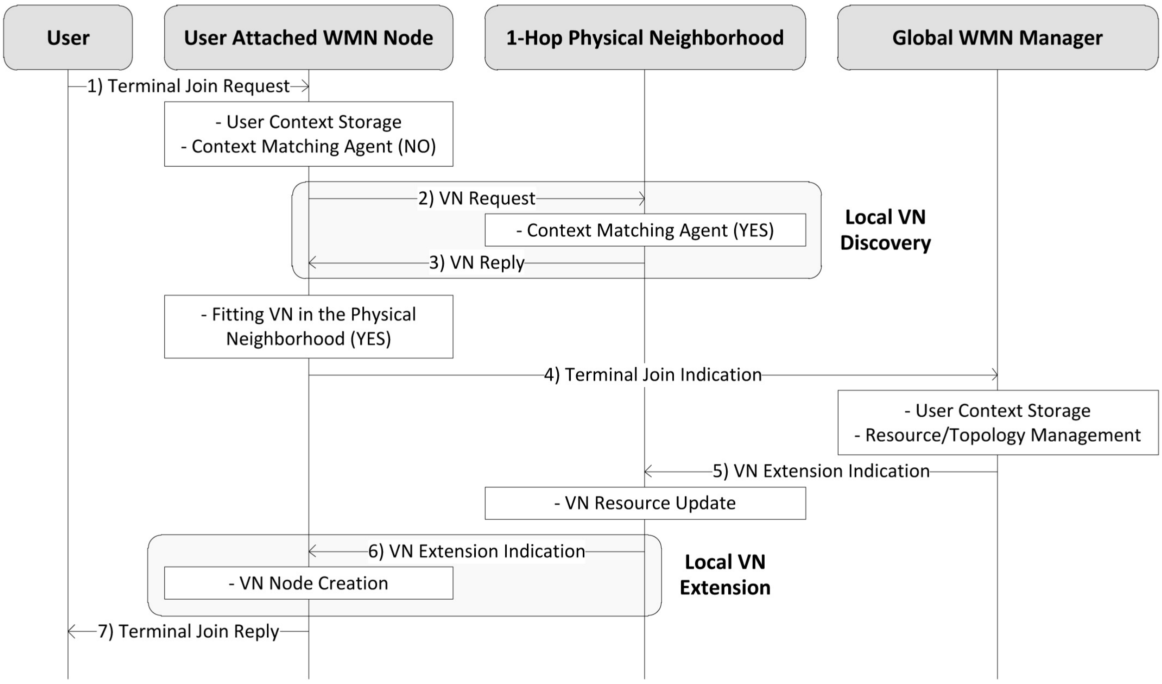


Figure 4.3: Scenario 2: Local VN Extension.

that need to be assigned to the new virtual link between nodes n_2 and n_1.

This topology and resource information is then conveyed in the message sent by the global WMN manager to node n_2 (Fig. 4.3→5), which can update the resources assigned to its virtual node from the VN v_u. In the following, node n_2 notifies node n_1 to create a new virtual node from the VN v_u, in order to extend the VN v_u (Alg.1-line8 and Fig. 4.3→6). Finally, user u is connected to the VN v_u (Alg.1-line12 and Fig. 4.3→7).

4.2.2.4 Scenario 3: Global VN Extension

After checking its 1-hop physical neighborhood, the WMN node n_1 can learn that the VN v_u is not available in the near WMN nodes. In this case, node n_1 triggers a global discovery mechanism in the WMN to find a point of attachment with a virtual node from the VN v_u (Alg.1-line10). This process is performed by using the distributed context-aware ring presented in Section 4.2.1. For instance, remembering Fig. 4.1, and assuming that node n_1 is node F and the VN v_u is VN$_1$, scenario 3 will take place since VN$_1$ is not supported by node F or by any of its 1-hop physical neighbors.

To start the global VN discovery in the WMN, node n_1 checks its stored information about the VN controllers that are located on it or on its 1-hop physical neighborhood, in order to redirect the discovery process to the zone of the ring that is semantically closer to the VN v_u. According to the example provided in Fig. 4.1, the VNs characterized by the same level of cost context are semantically closer in the ring (e.g., VN$_1$ and VN$_2$). Therefore, again assuming that node n_1 is node F and the VN v_u is VN$_1$, node n_1 will have information about the location of the controller of VN$_2$, since this controller is located in node C that is a 1-hop physical neighbor of node F. Node n_1 can then select the controller of VN$_2$ to initiate

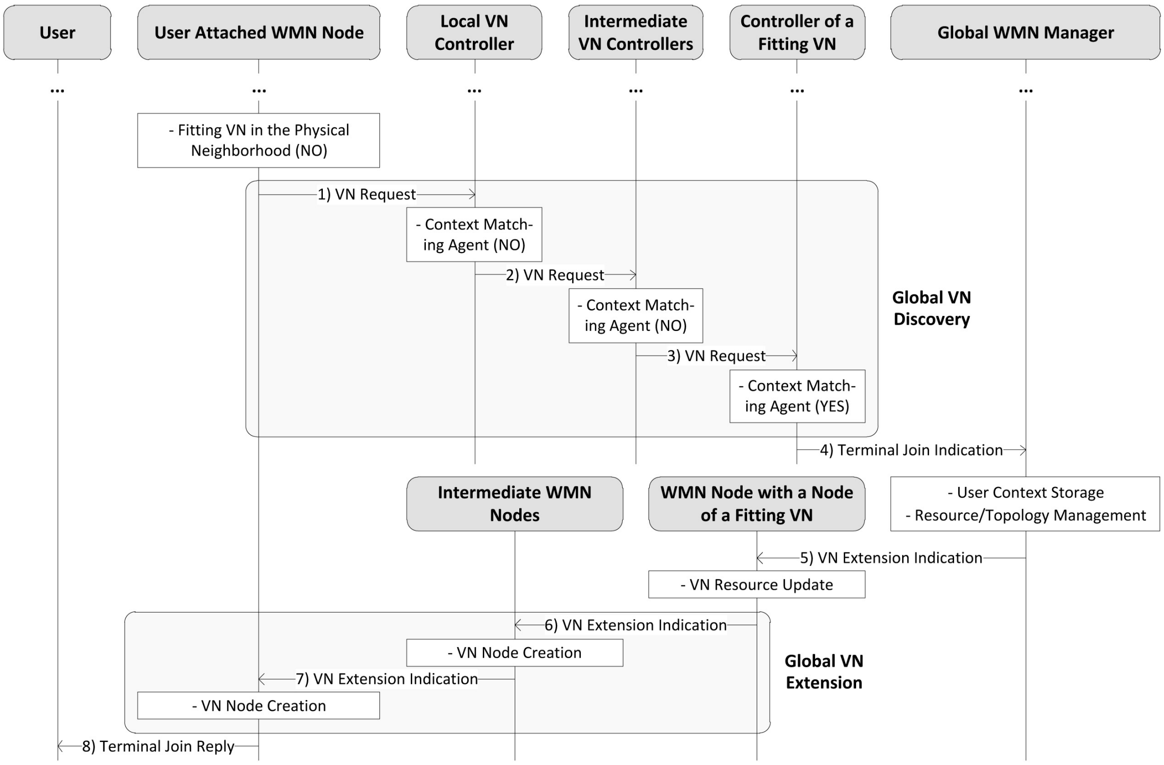


Figure 4.4: Scenario 3: Global VN Extension.

the global VN discovery (Fig. 4.4→1).

If node n_1 has no information about any VN controller to trigger the global VN discovery, it starts a hop-by-hop discovery process until finding any VN controller. Such hop-by-hop discovery process is unusual, and if performed, it will take a small number of hops.

Within the ring structure, the global VN discovery then continues through the links and shortcuts that are established according to the Chord-DHT [260], until finding the controller of the VN v_u (Fig. 4.4→2/3). At each VN controller, the levels of the VN context features are compared against the levels of the context parameters required by user u; it is then important to convey the context of user u in the global VN discovery request. Again using the example of Fig. 4.1, and assuming that it was selected the controller of VN_2 to initiate the global discovery of VN_1 (or the VN v_u), it will only be required one ring communication to find the controller of the VN v_u, since the controllers of VN_2 and VN_1 are 1-hop logical neighbors in the ring structure.

After reaching the controller of the VN v_u, the VN v_u needs to be extended up to node n_1. First, the controller of the VN v_u notifies the global WMN manager about the required extension. This notification is described in Fig. 4.4→4, and the used control message also includes the context of user u. Second, the global WMN manager stores the context of user u, and performs the topology and resource management functionalities to update the topology of the VN v_u, and the resources that need to be assigned to the new virtual connection from the VN v_u. As previously referred, the extension of the VN v_u is performed through the shortest WMN path between node n_1 and the nearest virtual node from the VN v_u (Alg.1-

line11). Third, this topology and resource information is conveyed in the message sent by the global WMN manager to the WMN nodes, through which the extension of the VN v_u will be performed. These WMN nodes can then update (Fig. 4.4→5) or create (Fig. 4.4→6/7) their virtual nodes from the VN v_u. Finally, user u is connected to the VN v_u (Alg.1-line12 and Fig. 4.4→8).

4.2.3 Real Implementation: Signaling Protocols

In the real implementation of the distributed mechanism for VN discovery and extension proposed in the last Section, we can resort to several signaling protocols available in the literature, instead of defining a new signaling protocol that cannot be compatible with the existing/standard ones. The use and/or adaptation of the mechanisms and messages of these signaling protocols can also enable the possibility to easily incorporate the proposed mechanism in many other scenarios and approaches. In the following, we present examples of signaling protocols available in the literature that can be incorporated in the proposed VN control mechanism, in order to implement the context transference, global VN discovery, resource reservation and authentication procedures.

First, the Context Transfer Protocol (CXTP) [286] can be used to perform the transference of the user context requirements between the user terminal and the WMN node where the user is connected to, and also between such WMN node and the global WMN manager.

Concerning the DHT-based context-aware overlay to perform the global VN discovery in the WMN, the code to build and maintain a distributed overlay based on the Chord-DHT is available in [287]. We need then to adapt this code to automatically assign a context-aware key to each VN controller based on the context parameters, and the levels of these parameters, that characterize such VN. Moreover, it is important to adapt this code to enable the dynamic update of the location of the key that characterizes a specific VN in the control overlay, when the location of the VN controller is updated to react to VN topology changes.

The Next Step in Signaling (NSIS) [288] protocol can be used to convey the E2E resource reservation requests along the WMN, when it is required to extend a VN to be included in a specific WMN node. Moreover, the Diameter [289] protocol can provide the means to authenticate mesh clients according to their credentials, and to authorize WMN resource reservations. In the following, we provide a short overview of these two signaling protocols, and detail how they can be integrated in the proposed distributed VN control mechanism.

4.2.3.1 NSIS

NSIS is an extensible and generic framework for enabling various signaling applications over IP-based networks. NSIS separates the functionalities for signaling message transport, such as reliability, fragmentation, congestion control and integrity, from signaling applications. Therefore, architecturally, NSIS consists of two protocol layers: the lower layer, the NSIS Transport Layer Protocol (NTLP), provides a generic transport service for different signaling applications that reside in the upper layer, the NSIS Signaling Layer Protocol (NSLP). These two protocol layers are shown in Fig. 4.5 [290].

NTLP is primarily composed by a specialized messaging layer, the General Internet Signaling Transport (GIST) [291], that runs over standard transport and security protocols to provide an universal transport service for signaling application messages. On the other hand, NSLP is designed for a particular signaling application, interacting with both the NTLP layer

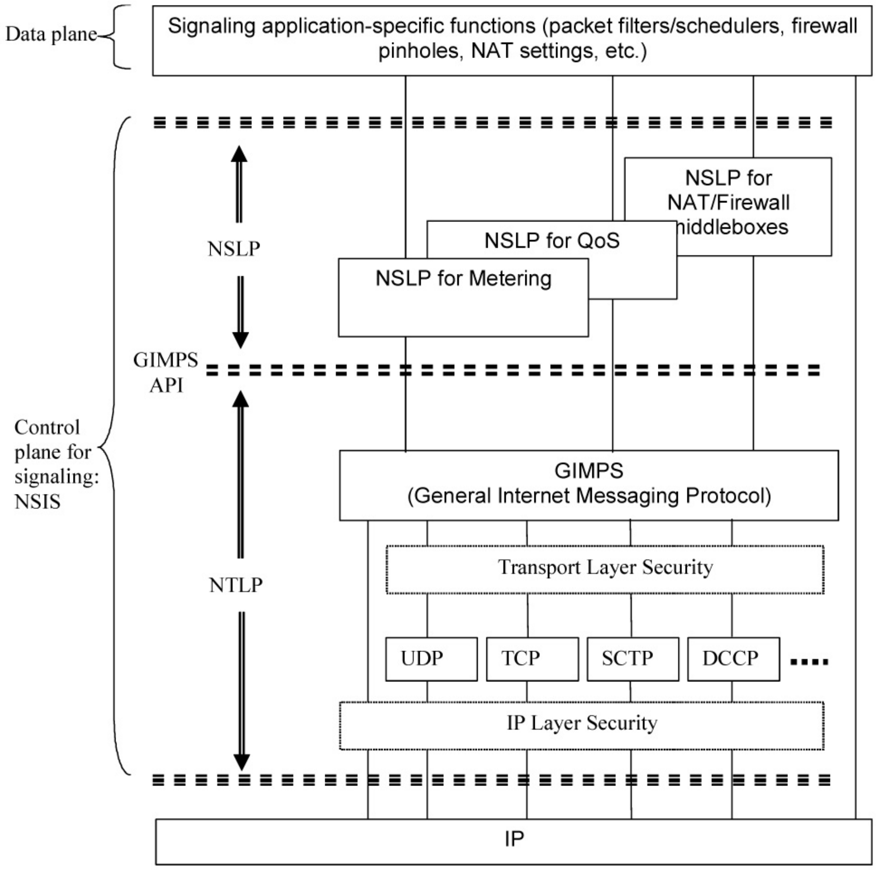


Figure 4.5: Logical Components in a NSIS-Aware Entity.

and the application-specific signaling. This upper layer includes formats and processing rules for the exchanged NSLP messages. NSIS also provides a mechanism to decouple the discovery of the next signaling node from the delivery of signaling messages between neighboring nodes, which introduces more flexibility to use standard transport and security protocols.

The relation among NSIS entities can be described as neighbors and/or adjacent peers; however, to be in an adjacent relationship, the NSIS entities have to support the same NSLP protocol. In the NSIS protocol operation, one node, the NSIS initiator, initiates signaling, while some nodes along the signaling path, called NSIS forwarders, intercept and then forward signaling messages, and the NSIS responder terminates the signaling. The basic NSIS message consists of a common header and a sequence of type-length-value objects. The common header indicates whether it is a datagram-mode (with UDP as the initial choice) or connection-mode (with TCP as the initial choice) message. It also indicates whether it is headed upstream or downstream, as well as the NSLP type and hop counter to avoid message loops.

In the scope of the proposed distributed VN control mechanism, the signaling application QoS NSLP [292], which was designed to provide E2E resource reservation signaling support, can be used to establish and manipulate resource reservations in the WMN, when required to create, reconfigure or remove VNs. The QoS NSLP is very well suited to guarantee a coherent QoS signaling along heterogeneous wireless networks and technologies, being able to interact with mobility protocols for seamless handovers. Therefore, we can resort to QoS NSLP to dynamically adapt the topologies and assigned resources of the context-aware VNs in dynamic WMN environments, in order to react to user context changes and mobility.

The QoS NSLP supports different message types that can be used in the proposed distributed VN control mechanism: *(i) RESERVE* to create, refresh, modify or remove the resources assigned to a VN at a specific WMN node; *(ii) QUERY* to collect information about the available resources on a possible path to create or extend a VN; *(iii) RESPONSE* to signal the success of a resource reservation, to respond to a *QUERY* message, or to send error-specific information; *(iv) NOTIFY* to indicate error conditions in an asynchronous way.

After installing the NSIS modules (GIST and QoS NSLP) in all WMN nodes, they are then able to perform the QoS negotiation and resource reservation process to dynamically create, extend and remove VNs in the WMN. The QoS parameters that characterize a resource reservation are included on a QoS Specification (QSPEC) object carried by the QoS NSLP messages. The QSPEC object then incorporates the information required to configure each node along the WMN path where a specific VN extension takes place.

For instance, when required to perform a QoS NSLP-based VN extension between the WMN nodes n_1 and n_2, n_1 first creates a *RESERVE* message containing the QoS information about the resources to be assigned to the VN in the QSPEC object (e.g., bandwidth). The message is then transported by GIST, in a hop-by-hop fashion, to n_2. At each WMN node of the path traversed between nodes n_1 and n_2, the QSPEC object is processed by the resource manager of such WMN node. The corresponding resources are then allocated to the VN that is being extended, according to the information included in the QSPEC object. Finally, n_2 sends a *RESPONSE* message to n_1, indicating the success of the VN extension.

4.2.3.2 Diameter

The Diameter protocol is intended to provide an Authentication, Authorization and Accounting (AAA) framework for applications such as network access or IP mobility. Diameter is implemented as a P2P architecture, meaning that every node where the Diameter protocol is deployed is able to act as a client or server.

In a general Diameter communication, a Diameter node sends a request message to another Diameter node acting as a server. The Diameter server then processes the received request, and decides the appropriate actions to be performed: if the request is successful, the Diameter server sends a response message to the Diameter client informing of the request success; if the Diameter server is unable to perform the requested actions, an error message is sent. The described architecture might seem comparable to a standard client-server architecture. However, in some situations, the Diameter server is also able to act as a Diameter client.

The network nodes where the Diameter protocol is deployed can be more than just clients or servers; these others are named Diameter agents, and they can span from relay, proxy, redirect and translation agents. The communication between Diameter entities is achieved by exchanging data packets through the use of the Diameter protocol, which has defined several types of Diameter messages identified by different command codes.

If we make use of the QoS NSLP signaling for VN resource reservation, we should have in mind that the current QoS NSLP implementation does not address AAA or micro-payment mechanisms for guaranteed access to resources. In such case, an individual NSIS entity will, in many cases, be unable to make an authorization decision by itself without consulting third parties; this is particularly true in an environment where hosts roam from one network to another. Therefore, the integration between the QoS NSLP and the Diameter protocol needs to be explored in the real implementation of the multi-VN architecture, so that NSIS entities can contact the AAA infrastructure to delegate the decision to authorize (or not) the resource reservations. Finally, the Diameter protocol can allow an user to be authenticated in the WMN, through the user credentials conveyed by the NSIS signaling.

4.3 Analytical Model: VN Discovery & Extension

4.3.1 Goals & Methodology

The previous Section proposed a set of control scenarios to associate users to VNs that exactly fit their context needs, which include: *(i)* the cooperation among WMN nodes to discover and adapt such VNs up to the locations of such users; and *(ii)* the use of a distributed structured-based context-aware platform to improve the global VN discovery in the WMN. This Section assesses the impact of applying such scenarios on the VN control and management.

It is foreseen that the balancing of the control responsibilities along the different WMN nodes and the fostering of the dynamic and autonomous cooperation of these nodes can: *(i)* avoid the complexity and limited scalability inherent to the existence of a single WMN controller; *(ii)* introduce flexibility to ease the support, control and management of the context-aware VNs in the WMN. But what are the real gains of different scopes of decision (local or global) for the distributed discovery and reconfiguration of VNs already available in the WMN? What are the open issues that need to be addressed in the evolvement of the proposed distributed control mechanism for VN discovery and extension?

These questions lead us to the definition of an analytical model (published in [12] and [16]) to evaluate the delays of the processes to locally or globally discover and extend a VN to assign to an user that arrives at a specific node of the WMN. In such WMN, there can be a distinct number of context-aware VNs, each one supported by a set of WMN nodes.

4.3.2 Model Definition

In the following, we adapt the analytical model proposed in Section 3.3.2 for data communications. We aim now to derive closed-form expressions for the mean delay of the local and global VN discovery and extension processes involved in the three control scenarios defined above. Based on a probabilistic approach for the occurrence of each control scenario, we also model the mean total delay for user association to the exactly fitting VN.

4.3.2.1 Model Preliminaries

In the modeling of the control plane, we will consider the same set of variables and assumptions already defined in Section 3.3.2 for the modeling of the architectural data plane. This way, the architecture is built over a grid-based WMN with N mesh routers, and c mesh clients are uniformly and independently distributed among them. Each VN is composed by a set of n virtual nodes that are uniformly distributed over the physical mesh routers. According to expression (3.14), the fraction of the total bandwidth of mesh links required to perform the control processes (and so, the communications involved in the three scenarios for user association) that take place in the physical WMN infrastructure is $W = \alpha_c \times W_T$.

In the definition of the analytical model for data communications, we derived expressions for the mean number of hops traversed by a data packet, $\overline{s}$ (please refer to expression (3.4)), and for the mean delay experienced by a data packet at router i, $\overline{D_i}$ (please refer to expression (3.11)). In the following two paragraphs, we re-define these two variables, but now in the context of control communications.

As mesh clients and VN nodes are uniformly distributed in the WMN, the grid-based WMN topology influences the mean number of hops traversed by a control packet, $\overline{s_c}$. Following the assumption of Kumar-Gupta model [293], $\overline{s_c}$ is defined by:

$$\overline{s_c} = \sqrt{\frac{N}{log(N)}}. \tag{4.1}$$

This expression can be directly applied to a global VN discovery, since it is performed among the VN controllers that compose the distributed structured-based ring. Each VN controller is randomly selected among the VN nodes, which are uniformly distributed over the physical mesh routers. However, in a global VN extension, N needs to be replaced by the mean number of virtual nodes per VN, n. From the relation achieved for $\overline{s}$ (please refer to expression (3.4)), the probability of locality of control traffic, p_c, is defined by:

$$p_c = \frac{1}{\overline{s_c}}. \tag{4.2}$$

The mean control packet delay at router i, $\overline{Dc_i}$, can be translated from $\overline{D_i}$ (please refer to expression (3.11)). In a VN discovery, $\overline{Dc_i}$ only depends on the control traffic that flows in the physical WMN infrastructure. However, in a VN extension, $\overline{Dc_i}$ is totally related to the characteristics of the data traffic that flows in such VN, since it is required to create new virtual nodes or to update the resources assigned to the existent ones.

4.3.2.2 VN Discovery

In a local VN discovery, the WMN node where the user is connected to, sends a message to its 1-hop physical neighbors, which then reply with an answer. The mean delay of a local VN discovery, D_{LD}, is then defined by:

$$D_{LD} = 2 \times \overline{Dc_i}. \tag{4.3}$$

A global VN discovery is performed along the distributed context-aware ring. Since mesh clients, VN nodes and VN controllers are uniformly distributed in the WMN, the starting point of this control process is uniformly distributed among all available WMN nodes. As previously referred, the work of this Chapter considers that the structure of the control ring is based on the Chord-DHT [260], and it is already known from the literature that the mean number of messages required by a Chord lookup procedure is $log(\#Peers) = log(M^C)$ (where C is the number of context parameters used to characterize the VNs, and M is the number of levels of these parameters). In this global VN discovery, the mean number of hops traversed by a control packet, $\overline{s_c}$, was already defined by expression (4.1). Moreover, by symmetry, the mean control packet delay at router i, $\overline{Dc_i}$, is the same at all routers. According to the discussion of this paragraph, the mean delay of a global VN discovery, D_{GD}, is then defined by:

$$D_{GD} = log(M^C) \times \overline{s_c} \times \overline{Dc_i}. \tag{4.4}$$

4.3.2.3 VN Extension

In a local VN extension, it is required to create one virtual link between the WMN node where the user is connected to, and one of its physical neighbors. The mean delay of a local VN extension, D_{LE}, is then defined by:

$$D_{LE} = \overline{Dc_i}. \tag{4.5}$$

The global extension of a VN up to the user location involves the communication between a point of attachment of such VN and the WMN node where the user is connected to, in order to create a new virtual connection. Again, the mean number of routers traversed by this communication can be obtained through expression (4.1), and, by symmetry, the mean communication delay at router i, $\overline{Dc_i}$, is the same at all routers. Therefore, the mean delay of a global VN extension, D_{GE}, is then defined by:

$$D_{GE} = \overline{s_c} \times \overline{Dc_i}. \tag{4.6}$$

4.3.2.4 Mean Delay for User Association

After setting the closed-from expressions for the mean delay of the local and global VN extension and discovery processes involved in the three proposed scenarios for user association to the exactly fitting VN, we now define the probability of each scenario to occur in order to derive the mean total delay for user association.

The probability of any virtual node (available in the WMN node where the user is connected to) to be part of the exactly fitting VN for the user is equal to the probability of scenario 1 to be triggered. As users and VN nodes are uniformly distributed in the WMN, this probability is defined by:

$$p_1 = \frac{1}{N}. \tag{4.7}$$

Scenario 2 is an extension of scenario 1. This way, the probability of scenario 2 to occur is equal to the probability of not triggering scenario 1 times the probability of at least one virtual node, available in the 1-hop physical neighborhood of the WMN node where the user is connected to, to be part of the exactly fitting VN for the user. As referred in Section 3.3.2, this analytical study considers that each WMN node has $K = 8$ physical neighbors, thus ignoring the fact that the nodes placed near the boundary of the considered WMN area will have a less number of neighbors. Since the users and VN nodes are uniformly distributed among the N WMN nodes, the probability of scenario 2 to occur is then defined by:

$$p_2 = \left(1 - \frac{1}{N}\right) \times \frac{K}{N}. \tag{4.8}$$

If the exactly fitting VN for the user is not available in the WMN node where the user is connected to, neither in its 1-hop physical neighborhood, the probability of scenario 3 to occur is defined by:

$$p_3 = \left(1 - \frac{1}{N}\right) \times \left(1 - \frac{K}{N}\right). \tag{4.9}$$

In our approach, scenario 1 does not include any VN extension or discovery process, scenario 2 involves the local variant of these two processes, and scenario 3 triggers the global VN discovery and extension after the inspection of the 1-hop user physical neighborhood.

Therefore, using a probabilistic approach, the mean total delay for user association to the exactly fitting VN, D_{Tc}, is then defined by:

$$D_{Tc} = p_2 \times (D_{LD} + D_{LE}) + p_3 \times (D_{LD} + D_{GD} + D_{GE}) \, . \tag{4.10}$$

4.4 Evaluation & Discussion

In the following, we evaluate the delay and overhead of the signaling messages required by the distributed control mechanism for user association to the exactly fitting VN.

First, Section 4.4.1 provides a study of the mean total delay for user association, in which the results are obtained from the analytical model previously defined. In order to validate the accuracy of the analytical model, Section 4.4.2 resorts to the simulation platform already presented in Section 3.4, in order to implement the proposed distributed control mechanism and to provide a set of results that are directly comparable to the analytical ones. Finally, Section 4.4.3 presents simulations to evaluate the feasibility of the distributed mechanism for user association and VN control under the influence of mobility of mesh clients.

4.4.1 Mean Delay of VN Discovery, Extension & User Association

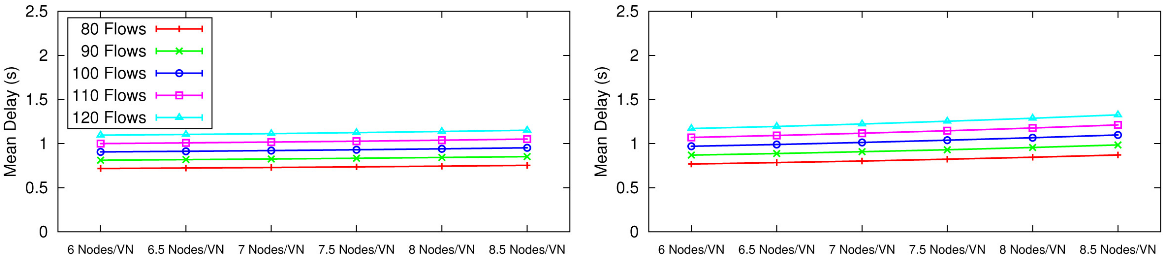
In this Section, we use the analytical model defined in Section 4.3 to determine the delays for user association to the exactly fitting VN, when varying the number nodes per VN and the number of users accessing a large size WMN.

In this analytical study, we resort to the setup already defined for the evaluation performed in Section 3.5.1, which is summarized in Table 3.1. Please note that the users accessing the WMN are equally distributed among the available mesh routers and VNs, and each user generates a traffic flow characterized by the features of the VN in which the user is connected to. In addition, we consider that each VN is characterized by $C = 5$ context parameters, each parameter can have M=4 levels, and the size of the packets required to perform the control processes involved in the three proposed scenarios for user association is S_c=48$Bytes$.

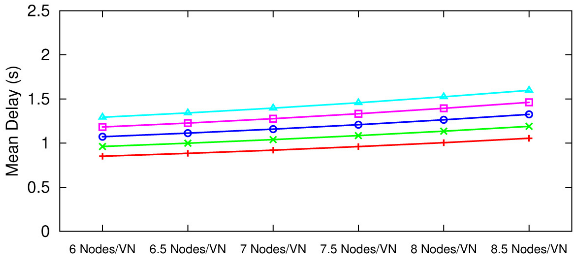
By running the analytical model when there are 12 VNs in the WMN (4 per each of the three considered VN types), and varying the number of nodes per VN and users in the WMN, Fig. 4.6 shows the mean total delay for user association, D_{Tc} (4.10), in a delay-sensitive VN (a), secure VN (b) and throughput-aware VN (c). Further, Fig. 4.7 presents the delay of each process involved in each scenario for user association, when there are 12 VNs established in the WMN, each one composed by 7 virtual nodes, and 100 users accessing the WMN.

As it was expected, the mean delay for user association increases from scenario 1 to 3, since each scenario is an extension of the last one. From the comparison between Fig. 4.6 and Fig. 4.7, we can see that the mean total delay for user association is clearly affected by scenario 3, since this is the most probable scenario to occur. Due to the large size of the WMN ($N = 10 \times 10$ mesh routers), there is often the need to start a distributed VN discovery in the WMN, which is the control process that lasts more time.

From the obtained results, we can conclude that when there is a high amount of users accessing a large size WMN, in the conditions above, the mean delays for user association can be higher than 1s. According to Fig. 4.7, this fact is explained by the high delays of the global VN discovery mechanism, since the local VN discovery and extension mechanisms present reasonable delays (usually lower than $150 - 200ms$). The high delays of the global VN discovery mechanism, modeled as a Chord lookup procedure, are due to: *(i)* first, this

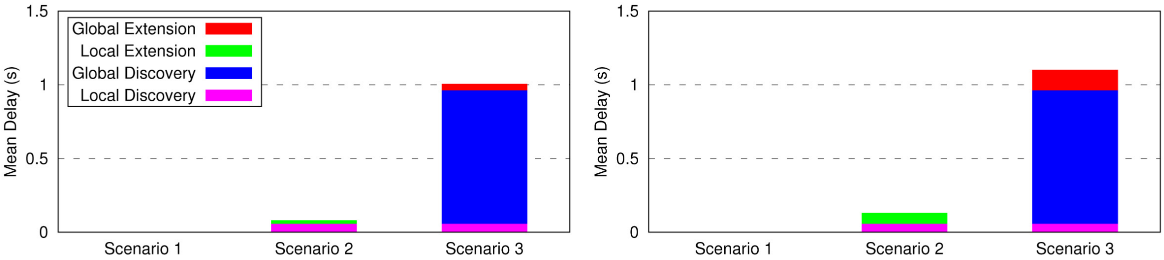


(a) Delay-Sensitive VN (b) Secure VN

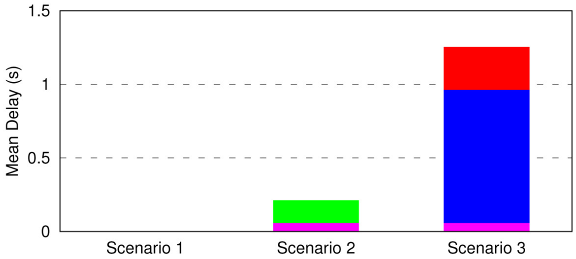


(c) Throughput-Aware VN

Figure 4.6: Analytical Model: Mean Delay for User Association in a Delay-Sensitive VN (a), Secure VN (b) and Throughput-Aware VN (c), with a Total Number of 12 VNs established in a Grid-Based 10 × 10 WMN, and varying the Number of Nodes per VN and the Number of Users in the WMN.



(a) Delay-Sensitive VN (b) Secure VN



(c) Throughput-Aware VN

Figure 4.7: Analytical Model: VN Discovery & Extension Delays for User Association in a Delay-Sensitive VN (a), Secure VN (b) and Throughput-Aware VN (c), with a Total Number of 12 VNs with 7 Nodes established in a Grid-Based 10 × 10 WMN, and 100 Users in the WMN.

process can involve a high number of communications among distinct VN controllers; *(ii)* second, these communications can travel a high number of WMN links, due to the high WMN size and the random distribution of VN controllers. This fact shows that there is the need to define a more intelligent distributed VN discovery mechanism than a Chord-based solution.

The local and global VN discovery processes are independent from the type of VN data traffic, since they take place in the physical WMN infrastructure (separation between data and control planes). Moreover, the VN discovery delays increase with the increasing number of users accessing the WMN, which imposes more control traffic in the WMN, and so, a higher level of interference among control packets.

The specific characteristics and volume of data traffic supported by each VN type affect the delay involved in the extension of such VN, thus affecting the mean total delay for user association: this value is lower for the user association to a secure VN than for a throughput-aware VN, and even lower for a delay-sensitive VN.

4.4.2 Model Validation

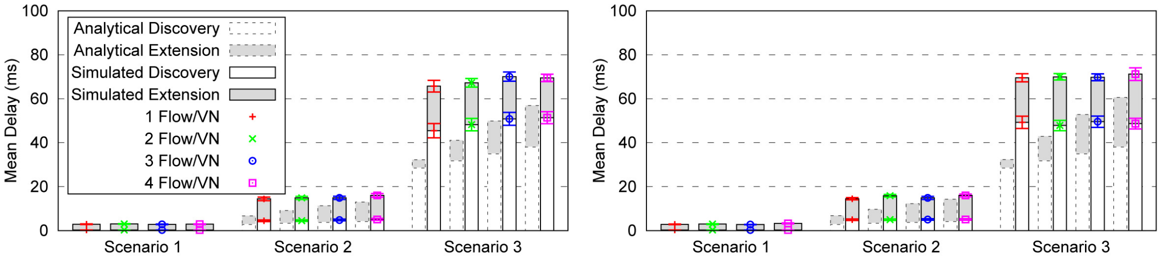
To assess the accuracy of the analytical results, this Section resorts to the simulation platform already presented in Section 3.4, to implement the proposed distributed control mechanism for user association. We then compare the delays of the VN extension and discovery processes, involved in the three proposed scenarios for user association, obtained through analytical and simulation tools. The simulations also allow to evaluate the overhead of the distributed mechanism for user association and VN control.

As referred in Section 3.4, we extend the simulator with the control mechanisms proposed for user association and VN control. In this context, the NS-2 Bamboo-DHT [285] is adapted to implement the distributed structured-based context-aware ring proposed in Section 4.2.1. We add new functionalities to the original Bamboo-DHT code in order to: *(i)* enable the automatic mapping of context information in specific Bamboo keys that identify the VN controllers (as described in Fig. 4.1); *(ii)* update the controller of each VN and the respective ring connections in case of adaptation of VN topologies due to context change, or mobility of VN users; *(iii)* adapt the number of shortcuts among Bamboo nodes according to the number of WMN nodes, VNs, context features and levels of such features.

In this study, we resort to the setup already defined for the evaluation performed in Section 3.5.2, which is summarized in Table 3.2, and we again consider $S_c = 48Bytes$, as in the previous Section. Then, by randomly placing users in the WMN to trigger the reconfiguration of the VNs that exactly fit their context needs, we analyze the number of times that each control scenario for user association occurs, and compute the mean delays of the VN extension and discovery processes involved in each of these scenarios.

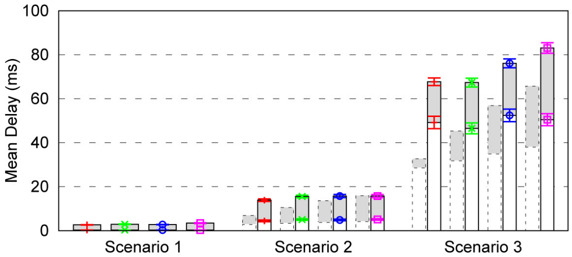
Fig. 4.8 presents the mean delays of the VN discovery and extension processes involved in each scenario according to the analytical and simulation results (20 simulations per value, with a confidence degree of 90%). Further, Fig. 4.9 shows the overhead of the local and global VN discovery processes of scenarios 2 and 3, respectively. Please note that this overhead reflects the percentage of traffic introduced by the two VN discovery processes in relation to the total amount of traffic that flows in the WMN.

From the obtained results, we conclude that scenario 1 only depends on the processing power of mesh routers, since the exactly fitting VN for an user is already available in the WMN node where the user is connected to. The VN extension delays are very small, only



(a) 9 VNs in the WMN

(b) 18 VNs in the WMN



(c) 27 VNs in the WMN

Figure 4.8: Analytical & Simulation Models: Mean VN Discovery & Extension Delays for User Association when varying the Number of VNs with 5 Nodes established in a Grid-Based 7 × 7 WMN (9 (a), 18 (b) and 27 (c)) and the Number of Users in the WMN.

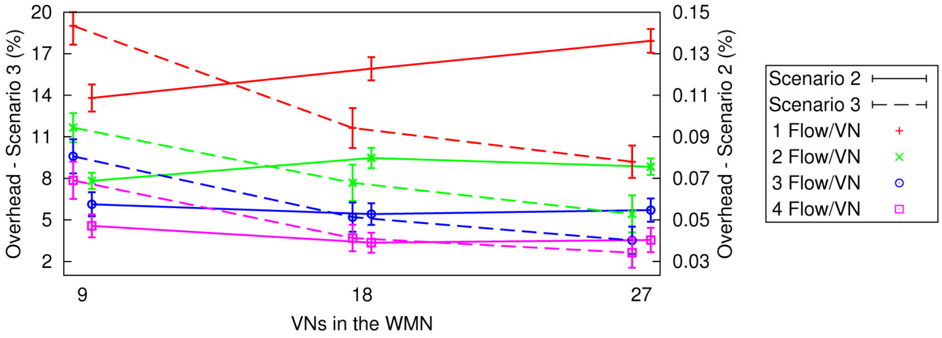
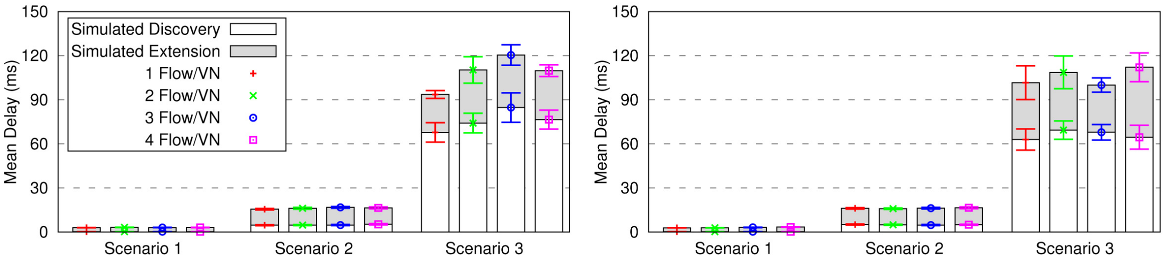


Figure 4.9: Simulation Model: Overhead introduced by the VN Discovery Mechanisms for User Association when varying the Number of VNs with 5 Nodes established in a Grid-Based 7 × 7 WMN and the Number of Users in the WMN.

comprising the creation of a virtual link between the user and the WMN node where he/she is connected to (this scenario was not addressed by the analytical model).

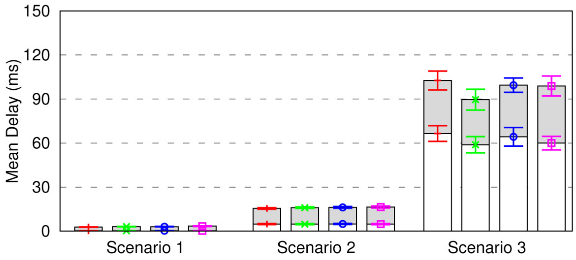
Scenario 2 involves a local VN discovery in the physical neighborhood of the WMN node where the user is connected to, and the creation of a new virtual link. Due to the small scope of these processes, their delays are also small, even if the number of VNs and flows increases. Moreover, the control traffic introduced in the WMN by the local VN discovery processes is clearly negligible when compared to the data traffic that flows in the WMN.

It is observed that, in scenario 3, the global VN discovery process has the largest delays (in the order of $60 - 80ms$). The controller of each VN is randomly placed in the WMN.



(a) 9 VNs in the WMN

(b) 18 VNs in the WMN



(c) 27 VNs in the WMN

Figure 4.10: Simulation Model: Mean VN Discovery & Extension Delays for User Association when varying the Number of VNs with 5 Nodes established in a Grid-Based 10 × 10 WMN (9 (a), 18 (b) and 27 (c)) and the Number of Users in the WMN.

Therefore, the number of control communications in the distributed control ring until finding the controller of the exactly fitting VN to assign to an user is variable, increasing the mean delay of this discovery process. However, these delays are not significant, and they are independent from the data traffic that flows in the WMN, since the communications performed within the control ring take place in the physical WMN infrastructure (separation between data and control planes). The overhead introduced by the global VN discovery processes decreases as the number of VNs and flows increase in the WMN; such overhead is higher when there is a lower amount of data traffic (1 flow per VN). Moreover, the overhead is higher than in the scenario 2, mostly because of the update and maintenance control traffic of the Bamboo-DHT. Concerning the global VN extension process of scenario 3, it is more time consuming than the one of scenario 2, since it is often required to create virtual connections with more than one virtual link. Finally, this VN extension process lasts more time when the WMN starts to be overloaded (4 flows per VN).

Comparing the two models, we can conclude that the analytical one is more conservative, presenting always lower delays than the ones obtained in NS-2. Both models present similar tendencies of delay increasing. However, the discrepancies between them are higher than in Fig. 3.6, since this model was originally deployed for data communications. In the one hand, concerning the global VN discovery of scenario 3, the analytical model does not take into account the maintenance control traffic of the Bamboo-DHT, which influences the NS-2 model. On the other hand, in NS-2, the VN extension process is also influenced by other variables that are not addressed by the analytical model, such as the buffer size of WMN nodes.

When performing simulations with a 10×10 physical WMN, the delay (see Fig. 4.10) and overhead (see Fig. 4.11) results of the global distributed VN discovery performed in scenario

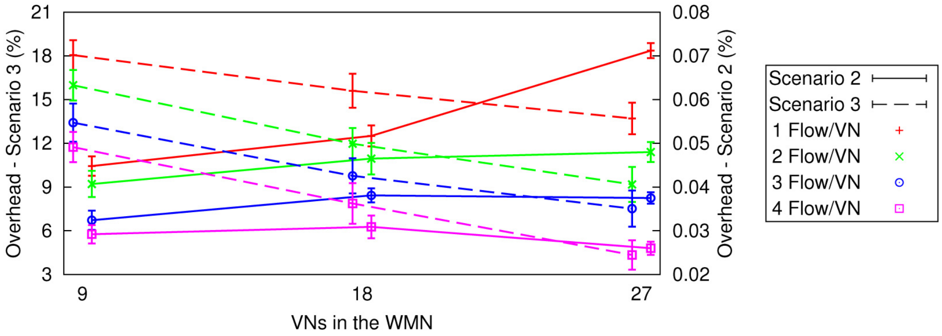


Figure 4.11: Simulation Model: Overhead introduced by the VN Discovery Mechanisms for User Association when varying the Number of VNs with 5 Nodes established in a Grid-Based 10×10 WMN and the Number of Users in the WMN.

3, are slightly higher than the ones of Fig. 4.8 and Fig. 4.9, respectively. Due to the larger size of the WMN, the control communications performed within the ring may have a large number of hops, which increases the control delay and overhead. However, the distributed VN discovery delays are always lower than $100ms$, even with the increasing number of VNs and flows in the WMN.

4.4.3 Influence of User Mobility

In this Section, we perform simulations with the setup defined in the last Section for a grid-based 7×7 WMN, but now introducing mobility of mesh clients to evaluate its influence in the distributed mechanism for user association and VN control.

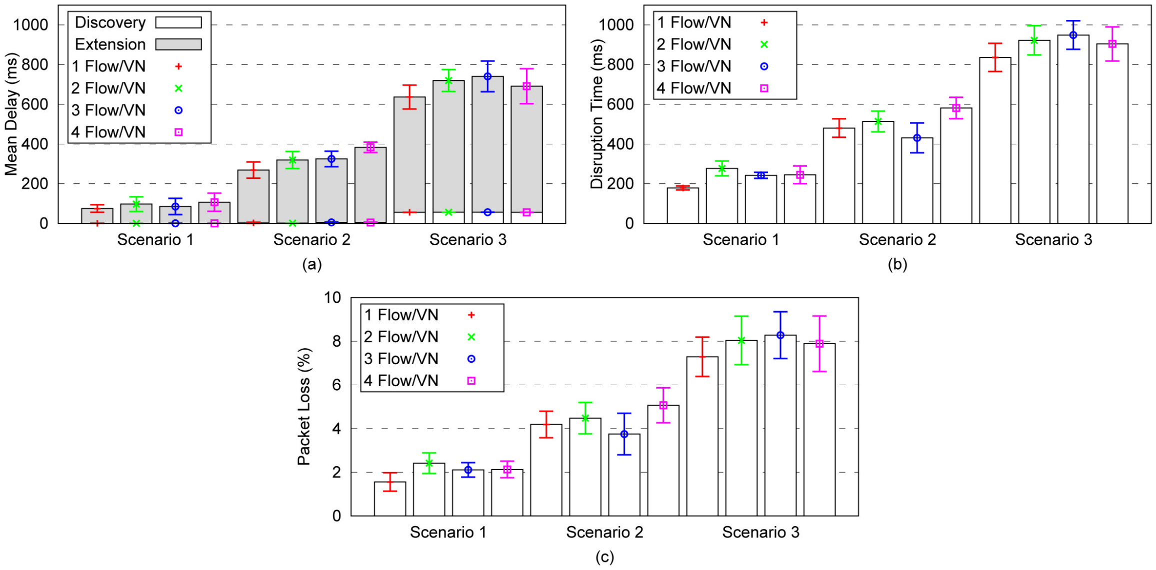
We consider 18 VNs in the WMN (6 per virtual slice) with a variable number of flows generated by mesh clients that are randomly moving among 1-hop neighboring mesh routers at a vehicular velocity of $15m/s$. Then, we evaluate the VN reconfiguration delays (see Fig. 4.12 (a)), the disruption time (see Fig. 4.12 (b)) and packet loss (see Fig. 4.12 (c)) of user communications during the mobility process, considering the three proposed control scenarios.

The results of Fig. 4.12 show that the VN discovery delays of scenarios 1 and 2 are not significantly affected by the dynamics of the environment. On the other hand, the distributed VN discovery of scenario 3 presents higher delays than in a static environment (see Fig. 4.8).

With respect to the VN extension processes, their delays slightly increase due to the mobility behavior of mesh clients. Several VN reconfigurations are simultaneously triggered in the WMN, probably occurring over the same physical nodes and links. Such fact increases the interference and delays involved in the creation of the new virtual connections.

The session disruption time is always lower than $1s$. Beyond the VN discovery and extension delays, this metric also includes the delay of the link-layer handover and of the communication path update. The session packet loss is always less that 10%, even if the WMN starts to be overloaded.

From the obtained results, we can conclude that user mobility has a significant impact on the communications, and this impact mainly depends on the required level of the VN reconfiguration.



(a) (b) (c)

Figure 4.12: Simulation Model: (a) VN Discovery & Extension Delays, (b) Disruption Time of each Session measured at the Target, (c) Packet Loss the each Session measured at the Target, considering Mobility of Users and a Total Number of 18 VNs with 5 Nodes established in a Grid-Based 7 × 7 WMN, and varying the Number of Users in the WMN.

4.5 Conclusions

This Chapter defined and evaluated a distributed control mechanism to associate users to the VNs that exactly fit their context needs. This mechanism has the following characteristics: *(i)* each WMN node is able to inspect the information of its supported VNs or of the VNs of its 1-hop physical neighborhood; *(ii)* it can then trigger the smooth reconfiguration of these VNs if they exactly meet the context needs of users that want to be connected to such WMN node; *(iii)* in addition, each WMN node may trigger the global discovery of points of attachment of specific VNs in the WMN, which can then be reconfigured in order to be included in such WMN node; *(iv)* it makes use of a distributed structured-based control ring, based on the cooperation among VNs characterized by similar context features, in order to quickly perform the global VN discovery in the WMN. With this distributed mechanism for VN control, we analyzed the impact and effectiveness of different scopes of decision for the fast discovery and reconfiguration of VNs already available in the WMN. Moreover, we discovered the open issues that need to be addressed in a more complete solution for the control and management of the multi-VN architecture.

To evaluate the distributed VN control mechanism, this Chapter adapted the analytical model defined for data communications in Chapter 3, in order to derive a closed-form expression for the mean total delay for user association to the exactly fitting VN. Further, we implemented the distributed approach in the simulator to compare the user association delays obtained with the simulation platform against the ones derived from the analytical model. The simulations also analyzed the overhead introduced by the VN discovery processes, and the impact of mobility of mesh clients on the VN reconfiguration delays.

From the obtained results, we concluded that the proposed distributed mechanism for VN discovery and reconfiguration presents acceptable values of signaling control delay and overhead. Therefore, it has potential to be integrated in a more flexible and scalable solution for the control and management of VNs in dynamic WMN environments. Moreover, the potential of DHT-based distributed solutions, along with their key-based routing functionalities, can allow the building of context-specific approaches to be applied to our concept. Finally, the closeness between the analytical and simulation results, for a specific network environment, allowed us to validate the proposed analytical model for control communications.

Despite the encouraging results of this Chapter, the several assumptions that were made, along with the high overhead introduced by the Chord-based global VN discovery, highlight the need to define and evaluate more appellative solutions to properly handle the context dynamics of WMN environments in the control and management of VNs. First, we need to introduce resource-aware metrics to select non-congested WMN paths to perform VN extensions, instead of using the shortest path between the user location and the nearest point of attachment of a fitting VN. Second, it is important to allow each WMN node to gather information of the VNs supported by its 2-hop or 3-hop physical neighborhood in order to increase the probability of local VN extensions to be triggered in the WMN, which are small time and resource consuming processes. Third, novel context-aware metrics for VN selection need be deployed to enable the user association to VNs that do not exactly fit all the context needs of users, but are characterized by context features that are very similar to such needs. Fourth, the penalties brought by a Chord-based mechanism for global VN discovery can be minimized with the proper implementation of several concepts that we aim to include in the distributed structured-based context-aware ring, such as: *(i)* the dynamic insertion, update and remotion of the elements (VN controllers) that compose the ring according to VN dynamics; *(ii)* the metric to select the best virtual node to be the controller of a VN, in order to try to improve the performance of the control and management functionalities performed inside the VN, and also to reduce the length of the communications performed among distinct VN controllers in a global VN discovery; *(iii)* the fast cooperation among the controllers of VNs, that are most probable to deal with user context requirements and their variations, can enable fast re-associations of users to other fitting VNs in case of user context change or mobility. Finally, the signaling mechanisms and the used criteria to decide when to trigger a VN creation or remotion have to be clearly defined. Solutions for these open issues will be proposed and evaluated in the following two Chapters.

Chapter 5

Context-Aware Control & Data Path

The use of context information to build personalized context-aware VNs and to drive several VN control and management functionalities is the scope of this Chapter. With this purpose in mind, we first propose rules or policies to map distinct levels of user context into suitable VN features, and also to derive the impact of distinct VN context-aware mechanisms on the performance of VN flows, and consequently on the perceived QoE of VN users.

We also address the context-aware selection mechanism of fitting VNs for users, through the proposal of a metric that takes into account the context flexibility of users. This metric allows users' associations to VNs that do not exactly fit all their requirements, but may incorporate these users if their context flexibility allows to.

Due to the strong emphasis on QoE in this Thesis, this Chapter ends with the definition and evaluation of a hop-by-hop reinforcement learning VN routing mechanism based on QoE feedbacks of ongoing VN users, which can be adaptable to the particular features of a context-aware VN.

5.1 Introduction

A flexible and autonomous control and management of VNs, that we aim for, has to be driven by the context requirements of users, without disregarding the context characteristics of the VNs available in the WMN. Therefore, we need to define intelligent mechanisms to properly handle the particularities and dynamics of the context information characterizing users and VNs. Such context information can span from the preferences, QoE expectations and location of users, which can change at a significant rate, or the context features and current quality status of VNs. Due to the context-driven architectural functionality, it is also important to model the user context requirements in levels, which have then to be properly mapped in context-aware VN features.

Given the distinct variants of user context requirements, Section 5.2 considers a demonstrative set of such requirements in order to define possible rules to model context in levels. In the following, we give examples of policies to: *(i)* automatically configure the structures, running flows, control protocols and assigned resources of each VN in order to meet a specific set of user context levels; *(ii)* derive the impact of these context levels on the perceived QoE of VN users. Finally, this Section provides a discussion on the importance of considering certain levels of flexibility provided by the users to be assigned to VNs.

When an user arrives at the WMN, the metric to select the VN in which the user will be connected to, has to be designed based on a combined view of the context information characterizing both the user and the VNs belonging to the candidate set of fitting VNs for the user. Moreover, this metric should enable the reduction of the number of WMN reconfigurations required to adapt the VN assigned to the user, in case of change of the user context requirements during the user connectivity time.

To this purpose, Section 5.3 presents a metric to select the best VN, from the candidate set of fitting VNs, to be assigned to the user. This metric considers the context flexibility of the user, which can be expressed as the allowed variation ranges of the context requirements of the user during his/her connectivity time, in order to reduce the number of VN re-associations. Such metric can also take into account the overall quality level of each VN (that belongs to the candidate set of fitting VNs for the user), which can be expressed by the overall QoE perceived by the users currently connected to such VN.

We end Section 5.3 by defining and evaluating an analytical model to derive the impact of incorporating different levels of context flexibility of users on the definition of the VN selection metric. This can allow users' associations to VNs already available in the WMN, while avoiding the constant creation of new VNs to serve single users. The results of this analytical model, in terms of the probability of triggering a VN update, extension or creation, are validated against the ones provided by a simulation study. By incorporating the VN selection metric in the distributed mechanism for user association and VN control defined in the previous Chapter, the simulation platform also allows to evaluate other performance gains of our approach when compared to other distributed VN control approaches.

Fully aligned with the context-aware VN configuration, Section 5.4 defines a double reinforcement learning strategy to dynamically select and update the routes to perform the communications between two distinct VN nodes. This routing strategy is adapted according to the specific QoE feedbacks of ongoing VN users. In this Section, we evaluate the gains of the QoE-aware routing strategy against the conventional AODV routing protocol, when applied to the VN routing in the proposed architecture for WMNs. We close the Chapter by summarizing its findings in Section 5.5.

5.2 User Context & VN Features

This Section focuses on the mechanisms for context modeling that need to be in place in the multi-VN architecture for WMNs.

First, the multi-context paradigm requires the correct modeling of the user context requirements to be properly used in the configuration and control of VNs. We then need to provide simple models for describing these requirements, which can simplify the processes by which they are stored, transported and used. In this Thesis, we use simple modeling based on levels to describe the user context requirements. Since VNs are built to personalize user communications, rules are then required to automatically map these levels in VN features.

Second, the QoE perceived by an user is commonly defined and measured in relation to E2E QoS metrics associated to the user flow, e.g., delay (D), data rate (R) and packet loss (PL), and represented as an unified MOS value. In the proposed multi-VN architecture, the quality of a VN flow is not only impaired by the amount of flows running in such VN, which will increase the traffic load and consequently the interference and collision probability, but it is also dependent on the VN context. It is then important to assess the impact of distinct VN context-aware mechanisms on the performance of VN flows, and consequently on the perceived QoE of VN users.

Third, users are characterized by multiple context parameters and the WMN can be the substrate for multiple VNs. Therefore, it is important to consider a certain level of flexibility to connect users to VNs that may not exactly fit all of their needs. Due to the user-centric perspective of the architecture, it is the user who is in the best position to further specialize and ease context-aware functionality and control of the architecture, and so, this context flexibility should be defined and controlled by the user.

Although there may be a large set of user context parameters to drive the multi-VN architecture, this Section considers the following demonstrative set of these parameters:

$$C_1 = \{\mathit{Service\ Type}, \mathit{Security}, \mathit{Energy}, \mathit{Price}\}; \tag{5.1}$$

however, mobility, communication type or preferred access technology are examples of other parameters. In the following, we present possible rules to model each $c \in C_1$ in levels, and then to map such levels in proper VN features (the basis of the mapping of user context in VN features is summarized in Table 5.1). We also discuss the impact of each $c \in C_1$ on the perceived QoE of VN users. At the end, we provide insight on the importance of user context flexibility in the architectural functionality.

5.2.1 User Context Modeling, Mapping & Impact on QoE

5.2.1.1 Service Type

Users can request distinct types of services (e.g., audio, video or data transfer) that can be grouped according to similarity on their particular requirements. Each group of services can be characterized by a level (e.g., 0...2, where 0 represents the group of audio services, 1 of video services, and 2 of data transfer services).

Each group of services, which can be supported by a specific type of VNs, imposes the assignment of distinct levels of resources to the nodes and links of the VNs that support such services. This way, the particular QoS requirements of each type of service have to considered in the definition of the functions to properly map and share the WMN resources among the

Table 5.1: Mapping between User Context and VN Features.

User Requirement	**VN Feature**
Service Type	• Resources Assigned to VN Nodes and Links
Security Constraints	• Authentication, Authorization and Accounting Protocols • Encryption and IPSec-Aware Mechanisms
Energy Consumption	• Interference- and Congestion-Aware Protocols • Operation Mode and Transmission Power of VN Nodes
Price Preferences	• Priority Assigned to the VN Channel

VNs. For instance, in expression (3.14), we already considered the throughput requirements of the services supported by a specific VN in order to configure the bandwidth assigned to the links of such VN. This bandwidth was defined as a tailored percentage of the total bandwidth available at the WMN links supporting such VNs.

As previously referred, the QoE perceived by an user is commonly derived from the QoS metrics (D, R and PL) related to the user service flow. However, the impact of resource limitations on such QoS metrics, and consequently in the user perceived QoE, is different for different service types. For instance, dropping a video I-frame has more severe consequences on QoE than dropping an individual audio packet.

In this Thesis, we make use of several models available in the literature to derive the perceived QoE of an user based on the QoS metrics of the user service flow supported by the VN in which the user is connected to. Considering audio, video or data transfer as the types of services supported by the available VNs, the following expressions, respectively defined in [294], [295] and [296], can be used to obtain QoE values:

$$QoE_{audio} = 1 + 0.035R_f + 7 \times 10^{-6} \times R_f \times (R_f - 60) \times (100 - R_f), \tag{5.2}$$

$$QoE_{video} = 4.5 - \frac{3.5}{1 + exp(0.5 \times (PSNR - 30))}, \tag{5.3}$$

$$QoE_{data} = 2.1 \times log_{10}[0.3 \times R \times (1 - PL)]. \tag{5.4}$$

The R_f factor in the QoE formula applied to audio flows (5.2) accounts for the impairments caused by delay and loss of audio packets. The $PSNR$ parameter used to derive the video QoE (5.3) takes into account the overall distortion caused by the loss of several video frames. The QoE formula for a data service (5.4) considers the data and packet loss rates associated to the requested data service. For more detail about these expressions please refer to [239].

5.2.1.2 Security

The incorporation of security mechanisms in the data exchange, such as authentication procedures (e.g., Secure Hash Algorithm - SHA), encryption protocols (e.g., Data Encryption Standard - DES, or 3DES) or IPv4/IPv6 tunneling, introduces extra resource consumption (CPU, memory and bandwidth) and increases the E2E delays of data communications, with a negative impact on the QoS metrics.

The employment of an authentication procedure to authorize or reject user access and to enable connectionless integrity increases the delays to establish a secure association, which includes the negotiation and exchange of the authentication protocol, algorithm and key. Moreover, if an encryption protocol is also applied to guarantee data confidentiality and integrity, the delays to encipher and decipher data at transmission are also added, which are dependent on the used encryption algorithm and key length. Further, both authentication and encryption mechanisms impose extra data packet headers that can lead to more collisions and higher packet loss rates due to the increase of the traffic load. Finally, according to the security protocol mode (transport or tunnel), the employment of a secure tunnel involves the addition of a new IPv4 (or IPv6) header, raising the required bandwidth to transmit the data.

In the multi-VN architecture, the security requirements of users can be mapped in ascendant levels according to the increase in the demand of such requirements; for instance, E-Commerce or online transfer requires a higher security level than online gaming. Then, the users' context levels have to be fulfilled by selecting and configuring proper authentication and encryption protocols to run in the VNs where these users are connected to. Therefore, each VN is characterized by a particular type (or level) of security that is related to the level of security requested by the users connected to such VN.

A specific VN security level will have a direct impact on the length of the extra security header of VN packets, on the user authentication delays (that can be averaged per transmitted packet in order to be reflected in the E2E session delay, since authentication is only performed at the beginning of a session), and on the time spent by the VN nodes to construct, encrypt, decrypt and process VN packets.

Therefore, in this Thesis, we characterize a VN security level by configuring the VN packets with an extra security header (EH) and the VN communications with an extra E2E delay (ED). In such approach, it is obvious that a high secure VN will have a higher data packets' delay than a small secure VN, due to the extra security header length and the security-related processing, such as complex encryption and decryption. According to this discussion, and considering three types of VN security levels, we will have:

$$EH = \begin{cases} EH_0 & \text{for small secure VNs} \\ EH_1 & \text{for medium secure VNs} \\ EH_2 & \text{for high secure VNs} \end{cases}, \; EH_0 < EH_1 < EH_2 \;\; [Bytes], \tag{5.5}$$

and

$$ED = \begin{cases} ED_0 & \text{for small secure VNs} \\ ED_1 & \text{for medium secure VNs} \\ ED_2 & \text{for high secure VNs} \end{cases}, \; ED_0 < ED_1 < ED_2 \;\; [s]. \tag{5.6}$$

Considering the same VN resources' availability, the increase of the VN security level will increase the length of the security header of VN packets and the security-related processing, which can lead to more collisions and higher packet loss rates in the VN, and it will impose a higher E2E delay of VN communications. Therefore, the increase of the VN security level will decrease the QoS of a VN flow, and consequently the perceived QoE of a VN user.

5.2.1.3 Energy

Energy is a critical resource of wireless environments, mainly because of the finite battery lifetime of wireless devices. In this sense, the reduction of the energy expenditures in the design of WMN infrastructures and mechanisms is a current key goal for the research community. Several proposals are available in the literature that aim to minimize the WMN energy consumption [297], such as: *(i)* coordinated switching among the operation modes of WMN nodes (transmit, receive, idle, sleep or off); *(ii)* dynamic adjustment of the transmission power of antennas based on the number of active users and volume of traffic load; *(iii)* intelligent selection of the level of network-layer broadcast redundancy and the congestion control scheme in order to reduce the overhead, error rate and number of collisions, which can result in packet retransmissions that lead to unnecessary power consumption.

The energy constraints of users' devices can be dynamically mapped in levels according to the ongoing battery capacity of such devices. Then, we need to differentiate the VNs according to the distinct levels of energy constraints of users' devices, and so, each VN has to be configured to meet the specific energy-efficient level requested by the users connected to such VN. A specific VN energy-efficiency level can be achieved through a proper coordination of one (or more) of the energy-aware mechanisms, presented above, in the VN nodes.

The increase of the overall VN energy savings will mainly decrease the performance of VN flows, and consequently decrease the perceived QoE of VN users, when the VN starts to be overloaded. However, due to the multi-dependent and complex relation between the increase of energy savings and the decrease of QoS/QoE, a concrete energy-QoS/QoE model is not available in the literature (as far as we know).

Therefore, in this Thesis, we adapt the utility-based function, proposed in [199], to derive the impact of a specific VN energy-efficiency level on the decrease of the QoE perceived by a VN user, with respect to the mean load of the VN. By resorting to such utility-based function, we can model the relation between "*the attenuation factor for the QoE perceived by an user connected to a VN characterized by a specific energy-efficiency level*" (defined by Y) and "*the mean resource occupancy of a VN node*" (defined by X). This relation is formalized by:

$$Y(X) = 1 - \frac{\left(\frac{X}{X_0}\right)^{\zeta}}{1 + \left(\frac{X}{X_0}\right)^{\zeta}} \;\; , \;\; 0 \leq X \leq 1, \tag{5.7}$$

where X is a normalized value that considers the mean load of each VN node with respect to its maximum resource capacity, ζ is the steepness parameter of the downward relation between X and Y, and X_0 is the middle point of the model (i.e., $Y(X_0) = 0.5$).

From expression (5.7), we observe that Y decreases in function of X, that is, when a VN starts to be overloaded (which is related to a higher mean load of each VN node - X), the QoE perceived by a VN user is lower. Nevertheless, it is still required to differentiate the model that is in place in a VN characterized by a high energy-efficiency level from the one in place in a VN characterized by a small energy-efficiency level. As easily inferred, the first one has a higher QoE decay than the second one, because the increase of the overall VN energy savings implies the reduction of the active or powered-on VN resources, which reduces the QoS of VN flows and QoE of VN users.

The differentiation of the model according to the VN energy-efficiency level is achieved by properly configuring the ζ and X_0 parameters. In our approach, in order to still guarantee minimum quality assurance when applying a certain scheme for energy savings, we consider that Y cannot be lower than a pre-defined value Y_{min}, that is, $Y(1) = Y_{min}$. Again, in a VN characterized by a high energy-efficiency level, Y_{min} will be lower than in a VN characterized by a small energy-efficiency level, since the smaller VN energy savings will cause a smaller reduction of the QoE of VN users. Following this strategy, after assigning a value for the

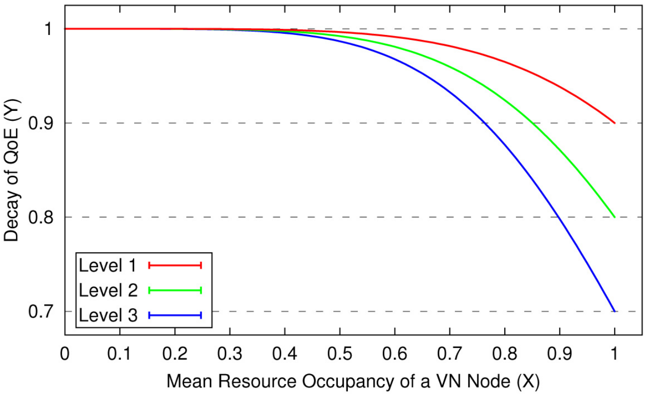


Figure 5.1: Impact of the VN Energy-Efficiency Level on the User QoE.

steepness parameter (ζ) and Y_{min}, we can derive the X_0 parameter to completely define the model that is in place in a VN with a specific energy-efficiency level.

In Fig. 5.1, we present an example on how the model, presented in expression (5.7), can be used to express the decrease of Y in function of X in three distinct types of VNs, where each VN is characterized by a small (1), medium (2) or high (3) energy-efficiency level. By assuming $\zeta = 5$, and considering that Y_{min} is equal to 0.9, 0.8 or 0.7 when we respectively apply the model in a small, medium or high energy-efficient VN, we can derive the X_0 parameter for each VN type. From the curves of Fig. 5.1, we can clearly conclude that a VN characterized by a high energy-efficiency level imposes a higher QoE decay than a VN characterized by a small or medium energy-efficiency level.

5.2.1.4 Price

The lower or higher price that users are willing to pay to get access to low or high quality communications, can be another feature to differentiate VNs of the architecture.

First, these price preferences can be mapped in distinct levels of users' price context. In the following, we can select the wireless channel assigned to a specific VN provider according to the levels of price context of the users connected to the VNs instantiated by such VN provider. Therefore, users that pay more (or less) to be served will be connected to VNs that are assigned to higher (or lower) quality channels.

To this purpose, it is known that the IEEE 802.11e technology [298] offers 7 levels of priority to differentiate the wireless traffic at the MAC level, and thus, it can be used to differentiate the quality of wireless transmission at each VN. Therefore, in the multi-VN architecture, we can resort to this IEEE 802.11e technology, in order to assign a specific MAC priority level (P) to the WMN channel assigned to a VN, based on the price preferences of the users connected to such VN. This way, a VN in which users that pay more (or less) to be served are connected to, will have assigned a high (or small) IEEE 802.11e-based MAC priority level to its working WMN channel. According to this discussion, and considering the existence of three VN types that differentially provide support for users with distinct price context, the MAC priority level assigned to each VN, P, will be defined by:

$$P = \begin{cases} 5 & \text{for VNs supporting users with small price requirements} \\ 6 & \text{for VNs supporting users with medium price requirements} \\ 7 & \text{for VNs supporting users with high price requirements} \end{cases}, \tag{5.8}$$

where 7 is the highest MAC priority level provided by the IEEE 802.11e technology.

Finally, this specialized VN channel prioritization scheme will induce distinct buffer delays and losses at the VN nodes, and bandwidth assigned to the VN links. Therefore, it directly influences the performance of VN flows, and consequently the perceived QoE of VN users.

5.2.1.5 Impact of Context on User QoE

Throughout Section 5.2.1, we have presented rules to model the service, security, energy and price requirements of users in context levels, and then, to use these levels to automatically select and configure the flows, resources, channels and control protocols that need to be assigned to a specific context-aware VN. Moreover, we have provided insight on the impact of the proposed mechanisms to map user context in VN features on the quality performance of VN flows, and therefore, on the QoE perceived by VN users.

Remembering the previous discussion, the type of service running in a specific context-aware VN can be used to derive the formula to measure the QoE perceived by the users connected to such VN. In the expressions (5.2), (5.3) and (5.4), we presented possible examples to measure the QoE of audio, video and data transfer services. These QoE expressions are derived from the QoS metrics related to the VN services, being the main important the delay (D), data rate (R) and packet loss rate (PL).

In a traditional networking environment, these QoS metrics are affected by the level of resources assigned to such VN, and by the amount of flows running in a specific VN, which will affect the traffic load and consequently the interference and collision probability. Nevertheless, in our multi-VN architecture, these QoS metrics (D, R and PL) are also dependent on the context-aware network mechanisms that are in place in the VN, which needs to be properly configured to meet the context needs of its users.

This way, along Section 5.2.1, we have also presented several functions to model the: *(i)* impact of the VN security level on the increase of the length of the security header of VN packets (EH, as defined in expression (5.5)) and of the extra E2E delay of VN communications (ED, as defined in expression (5.6)); *(ii)* influence of the VN energy-efficient level on the decay of the user perceived QoE (Y, as defined in expression (5.7)); *(iii)* effects of the VN price context on the selection of the priority assigned to the WMN channel associated to the VN (P, as defined in expression (5.8)). All these functions have a direct impact on the QoS metrics related to a VN service flow (D, R and PL), and therefore, on the QoE perceived by a VN user. Therefore, in order to summarize the previous discussion, the QoE perceived by the user u that is connected to the VN v, $QoE_{(u,v)}$, can be formally defined by:

$$QoE_{(u,v)} = Y \times QoE_{service}\left(D + D_{(EH,ED,P)}, R + R_{(EH,ED,P)}, PL + PL_{(EH,ED,P)}\right). \tag{5.9}$$

5.2.2 User Context Flexibility

Due to the dynamics of WMNs, where users are constantly changing their locations, devices or required services, this Thesis will consider certain levels of flexibility in the context needs requested by the users to assist several architectural control functionalities.

In this Thesis, this flexibility is modeled by allowing a certain range of variation for the demand of each user context requirement during his/her connectivity time; in such approach, the requirements that are more probable to change are the ones admitting high variation ranges. It is foreseen that the introduction of user context flexibility can increase the possibility of re-using VNs available in the WMN, since users can be connected to a more broad set of VNs, instead of the constant creation of new VNs to serve single users. Moreover, other mechanisms for architectural control can take advantage of this flexibility:

- In the one hand, since each VN is configured to meet a specific set of context features, the user context flexibility can allow the association of the user to more than one VN that does not exactly fit all of user context levels, but may incorporate this user if his/her context flexibility allows to. Thus, the distinct levels of user context flexibility need to be considered to define the metric to select the best VN to connect the user, from the candidate set of fitting VNs, in order to reduce the number of VN re-associations during the user connectivity time (this thematic will be addressed in Section 5.3);

- On the other hand, the distinct levels of user context flexibility can be used to optimize the connections that are established among the VN controllers that compose the distributed context-aware ring, proposed in the previous Chapter. This can be important to decrease the time spent in the distributed discovery of a point of attachment of a fitting VN to assign to the user. In Chapter 6, we will propose a scheme to enable the fast cooperation among controllers of VNs that are more probable to deal with user context requirements and their variations, thus enhancing the distributed ring structure.

Concerning the definition of the user context flexibility, we first consider that the demand of a specific user context requirement (e.g., c) can assume a distinct set of levels during the user connectivity time. Second, we consider that the mean time elapsed between two variations of the demand of the user context requirement c, is represented by a specific time interval Δ_c; therefore, the user context requirements characterized by a lower Δ_c can change its demand with a higher probability than the ones characterized by a higher Δ_c. Moreover, we consider that the user context requirements, characterized by a lower Δ_c, can admit more distinct levels during the user connectivity time than the ones characterized by a higher Δ_c, since they can more frequently change its demand.

According to this discussion, this Thesis models the flexibility associated to the demand of each user context requirement c, by assigning a maximum range of variation (or maximum set of levels) that such demand can admit during the user connectivity time, which is represented by T_c ($\propto \frac{1}{\Delta_c}$). In such approach, the requirements that are more probable to change are the ones admitting high variation ranges.

When considering the user context requirements represented by the set C_1, we conclude that they can admit distinct ranges of variation for their demands. First, since VN providers pay to instantiate VNs supported by wireless channels characterized by a specific quality (which can be derived from the MAC priority level assigned to such channels), they aim to increase their revenues. However, since users are usually reluctant to exceed their price preferences, we then consider that the accepted variation range of price preferences during users' connectivity times is extremely restrictive. Moreover, security constraints are related to services' purposes and trustability; variations of these demands are also rarely allowed by users. In addition, users may change their requested services and characteristics, or may easily have access to devices' battery chargers. This implies that the requirements of users'

Table 5.2: Architectural Definitions & Notations.

Definition	**Notation**
WMN	(N, L)
Set of WMN Nodes	$N=\{n_1, ..., n_{\vert N \vert}\}$
Set of WMN Links	$L=\{(n_a, n_b) \in N \mid n_a$ is in transmission range of $n_b\}$
Set of l-hop Neighbors of $n \in N$	$K_{(n,l)}$
Shortest Path between n_w,$n_z \in N$	$L_{n_w \rightarrow n_z}$
Set of User Context Requirements	$C=\{c_1, ..., c_{\vert C \vert}\}$
Set of Possible Levels of $c \in C$	$M_c=\{1, 2, 3, ..., \vert M_c \vert\}$
Set of Possible Users in the WMN	$U=\{u_1, ..., u_{\vert U \vert}\}$
Node where $u \in U$ is connected to	$n_u \in N$
Context Requirements of $u \in U$	$R_u=\{r_{u_c} \vert r_{u_c} \in M_c, c \in C\}$
Maximum Variation Range of the Context Requirements of $u \in U$	$T_u=\{t_{u_c} \vert t_{u_c} \in M_c, c \in C\}$
Set of Possible VNs in the WMN	$V=\{v_1, ..., v_{\vert V \vert}\}$
Context Features of $v \in V$	$R_v=\{r_{v_c} \vert r_{v_c} \in M_c, c \in C\}$
Set of Nodes supporting $v \in V$	$N_v \subset N$
Set of Users connected to $v \in V$	$U_v \subset U$
Set of VNs supported by $n \in N$	$V_n \subset V$
Exactly Fitting VN for $u \in U$	$v'_u \in V$
Set of Fitting VNs for $u \in U$	$V'_u=\{v \in V \vert \forall c \in C, \vert r_{u_c} - r_{v_c} \vert \leq t_{u_c} \quad \&\& \quad \exists c \in C, \vert r_{u_c} - r_{v_c} \vert \neq 0\}$
VN where $u \in U$ is connected to	$v_u \in V_u = v'_u \cup V'_u$.

services or the energy constraints of users' devices may admit some variation ranges during the user connectivity time. In summary, the maximum range of variation (or maximum set of levels) that the demand of each $c \in C_1$ can admit during the user connectivity time, T_c, has to be considered lower for price and security context, than for service and energy context.

5.3 Context-Aware VN Selection Mechanism

This Section presents and evaluates a metric to dynamically select (or update) the best VN, from the candidate set of VNs available in the WMN, to assign to a specific mesh client that arrives at the WMN (published in [18]).

Due to the user-centric perspective of the architecture proposed in this Thesis, this VN selection metric is designed to deal with the context requirements of users and to quickly adapt to variations in these requirements. Moreover, it considers the context features that characterize the VNs available in the WMN, and the overall quality level of these VNs.

5.3.1 Preliminary Definitions

Before the proposal of the VN selection metric, we present a set of notations related to the context-aware multi-VN architecture for WMNs, that will aid the formal description of such metric. These notations are summarized in Table 5.2, and they will also be used in Chapter 6 to assist the description of several control and management mechanisms.

The WMN is formally defined by (N, L), where the set $N = \{n_1, ..., n_{|N|}\}$ represents the set of WMN nodes (in the overall document, $|X|$ is representative of the number of elements of the set X). L is the set of WMN links between neighboring elements of N, being defined by $L=\{(n_a, n_b) \in N \mid n_a$ is in transmission range of $n_b\}$. The set of traversed wireless links of a communication performed through the shortest path between the nodes n_w,$n_z$$\in$$N$ is defined by $L_{n_w \to n_z} = \{(n_w, n_x), ..., (n_y, n_z)\}$. The set of nodes located at l hops of distance to the node $n \in N$, or the set of l-hop neighbors of the node $n \in N$, is defined by $K_{(n,l)}$.

The architecture is driven by a set $C=\{c_1, ..., c_{|C|}\}$ of user context parameters. Each context parameter $c \in C$ may be quantified into $M_c=\{1, 2, 3, ..., |M_c|\}$ normalized levels.

A set $U=\{u_1, ..., u_{|U|}\}$ of users may access the WMN, and the WMN node where the user $u \in U$ is connected to, is defined by $n_u \in N$. The context levels requested by the user $u \in U$ are represented by $R_u=\{r_{u_c} \mid r_{u_c} \in M_c,\ c \in C\}$. Still characterizing the user $u \in U$ based on the discussion of Section 5.2.2, the maximum variation ranges of the demands of his/her context requirements are represented by $T_u=\{t_{u_c} \mid t_{u_c} \in M_c,\ c \in C\}$. For instance, if $|C| = 4$, $|M_c| = 5$ $(\forall c \in C)$, $R_u = \{3, 3, 3, 3\}$ and $T_u = \{1, 0, 2, 2\}$, after inspecting the result of $R_u \pm T_u$, we can conclude that the user $u \in U$ can admit: *(i)* the levels $\{3 \pm 1\} = \{2, 3, 4\}$ on the first context parameter; *(ii)* only the level 3 on the second context parameter; *(iii)* any level on the third and fourth context parameters, which are the more flexible ones.

The WMN is the substrate for a set V of possible VNs. Each VN $v \in V$ is properly configured to meet the context levels represented by the set $R_v = \{r_{v_c} \mid r_{v_c} \in M_c,\ c \in C\}$. The identifier of the VN $v \in V$ is related to R_v. For instance, if $|C| = 4$ and $|M_c| = 5$ $(\forall c \in C)$, the $v_{ID} = 3333$ is representative of the VN $v \in V$ that is configured to meet the normalized level 3 on each context parameter $c \in C$, that is, all context parameters are set to the level 3.

Each VN $v \in V$ is supported by a set $N_v \subset N$ of nodes and it is accessed by a set $U_v \subset U$ of users. Each node $n \in N$ is the substrate for virtual nodes of a set $V_n \subset V$ of VNs.

The VN $v'_u \in V$ exactly fits the context requirements of the user $u \in U$, that is, the VN $v'_u \in V$ is characterized by the same context parameters, and also by the same levels of these context parameters, when compared to the ones requested by the user $u \in U$ in the vector R_u. All the other VNs do not exactly fit the context requirements of the user $u \in U$, that is, these VNs have at least one context feature that is set to another level than the one requested by the user $u \in U$. The set $V'_u=\{v \in V | \forall c \in C, |r_{u_c} - r_{v_c}| \leq t_{u_c}\ \&\&\ \exists c \in C, |r_{u_c} - r_{v_c}| \neq 0\}$ is composed by the partially fitting VNs, which are the ones that do not exactly fit all the requirements of the user $u \in U$, but each feature of the VN $v \in V'_u$ is characterized by a level that meets the threshold defined by the user $u \in U$ for such feature. The user $u \in U$ is connected to the VN v_u, being such VN selected from the candidate set $V_u=v'_u \cup V'_u$ of VNs.

5.3.2 VN Selection Metric

Due to the flexibility of the user $u \in U$ to be connected to more than one VN available in the WMN, this Section proposes a context-aware metric to select the best fitting VN v_u to assign to the user u. The VN v_u is selected from the set of fitting VNs for the user u (V_u).

More specifically, we define a metric to compare the **C**ontext-aware **D**ifference among the context needs of the user u (R_u, T_u) and the context features of the VN $v \in V_u$ (R_v), being such metric represented by $CD_{(u,v)}$. In the one hand, $CD_{(u,v)}$ is designed to more frequently select the VN $v \in V_u$ that is configured to exactly meet the context requirements of the user u (R_u). On the other hand, $CD_{(u,v)}$ also takes into account the maximum variation ranges of the demands of the context requirements of the user u (T_u). This can enable the possibility to select VNs that are more probable to deal with high variations in the context of the user u, which can reduce the number of VN re-associations during the connectivity time of the user u, as will be explained below.

Due to the multitude of user context parameters (C) and their levels ($M_c, \forall c \in C$), $CD_{(u,v)}$ is defined as a weighted sum of the absolute differences between the context requirement of the user $u \in U$, r_{u_c}, and the context feature of the VN $v \in V_u$, r_{v_c}. Therefore, $CD_{(u,v)}$ is represented by:

$$CD_{(u,v)} = \frac{1}{|C|} \times \sum_{c \in C} \frac{|r_{u_c} - r_{v_c}|}{t_{u_c}} \;\; , \;\; 0 \leq CD_{(u,v)} \leq 1, \tag{5.10}$$

where $\frac{1}{t_{u_c}}$ is the weight assigned to the absolute difference between r_{u_c} and r_{v_c} ($\forall c \in C$).

According to the discussion of Section 5.2.1, a specific VN context feature (r_{v_c}) can be automatically configured to meet a specific context requirement of a group of users (r_{u_c}). In this case, VN features can be set based on the service, security, energy or price preferences of users, and so, these VN features remain statically configured during the VN lifetime (the level assumed by r_{v_c} does not change).

Nevertheless, the QoE expectations of a group of users can be another parameter to take into account in the VN selection process. In this case, it is important to compare the QoE expectations of a specific user $u \in U$ against the overall quality level provided by the VN $v \in V_u$, which can dynamically change during the VN lifetime. Therefore, in order to allow the possibility to incorporate this dynamic VN feature in the VN selection process, we consider that the overall quality level of the VN $v \in V$ is measured in relation to the mean perceived QoE of ongoing users in such VN, U_v, and it is represented by QoE_v. In expression (5.9), we already defined the QoE perceived by the user $u \in U$ that is connected to the VN $v \in V$, by $QoE_{(u,v)}$. QoE_v can then be defined by:

$$QoE_v = \frac{1}{|U_v|} \times \sum_{u \in U_v} QoE_{(u,v)}. \tag{5.11}$$

Concerning the weight assigned to the absolute difference between r_{u_c} and r_{v_c} ($\forall c \in C$), it is derived based on the maximum variation range of the demand of r_{u_c} (t_{u_c}). As explained in Section 5.2.2, the user context requirements that are more probable to change during the user connectivity time, are the ones admitting high variation ranges (or the ones that can take a more broad set of levels). Therefore, in this VN selection process, we assign a lower weight to the absolute context-aware differences on these parameters, than on the parameters that can take a more restricted set of levels (or admit low variation ranges). Following this strategy, we can reduce the number of future required adaptations of the selected VN to assign to an user, due to variations on the user context requirements, as explained in the following example.

For instance, consider that $|C| = 4$, $|M_c| = 5$ ($\forall c \in C$), $R_u = \{3,3,3,3\}$, $T_u = \{1,2,2,2\}$, and there are two fitting VNs for the user $u \in U$: the VNs v_1,$v_2 \in V_u$, where $R_{v_1} = \{3,2,3,3\}$ and $R_{v_2} = \{2,3,3,3\}$. Although $\sum |R_u - R_{v_1}| = \sum |R_u - R_{v_2}|$, the impact of a future variation of $r_{u_{c_1}}$ is higher than if such change occurs on $r_{u_{c_2}}$, since $t_{u_{c_1}} < t_{u_{c_2}}$. Therefore, it is important to select the VN v_1 as v_u, since $r_{u_{c_1}} = r_{v_{1_{c_1}}}$, while the VN v_2 is already in the lower border to meet $r_{u_{c_1}}$. Both the VNs v_1 and v_2, if selected, still have a high probability to allow a future variation of $r_{u_{c_2}}$, and so, we need to assign a lower weight to the absolute context-aware difference on the parameter c_2 than on the parameter c_1. According to this discussion, the weight assigned to the absolute difference between the user context requirement r_{u_c} and the VN context feature r_{v_c} ($\forall c \in C$) is inversely proportional to the allowed variation range of the demand of r_{u_c} (t_{u_c}), that is, the relative importance of $|r_{u_c} - r_{v_c}|$ increases with a lower t_{u_c}. In expression (5.10), this weight is then defined by $\frac{1}{t_{u_c}}$ ($\forall c \in C$).

Remembering the distributed VN control mechanism presented in Chapter 4, this context-aware VN selection metric can be used by the WMN node $n_u \in N$ where the user $u \in U$ is connected to, in order to select the best fitting VN $v_u \in V_u$ to assign to the user u, from the ones supported by the node n_u. Moreover, the node n_u can use this metric to select the best fitting VN for the user u, that needs to be extended up to it. In this case, this metric is applied over the set of fitting VNs for the user u, that are discovered in the WMN through the local or global VN discovery mechanisms triggered by the node n_u. Therefore, the node n_u can inspect its gathered VN information to select the best fitting VN to assign to the user u, v_u, as the one that minimizes the $CD_{(u,v)}$ metric (defined in expression (5.10)). The VN v_u is then selected according to:

$$v_u \in V_u \;\; s.t. \;\; \min_{v \in V_u} \{CD_{(u,v)}\}. \tag{5.12}$$

5.3.3 Analytical Model: Impact of User Context on VN Selection

5.3.3.1 Goals & Methodology

In this Section, we assess the impact of the context-aware VN selection metric, defined in expression (5.10), on the control and management of the context-aware VNs.

To this purpose, we present an analytical model (published in [18]) to derive the impact of considering distinct levels of user context flexibility (associated to the maximum variation ranges of the demands of the user context requirements, T_u) on the increase of the possibility of re-using VNs available in the WMN, while decreasing the need to create new VNs. Such impact is captured by defining expressions for the probability to update or extend a VN, already available in the WMN, to connect a new user that arrives at the WMN, or to create a new VN. In such WMN, users are characterized by very heterogeneous context requirements, and they are constantly arriving and leaving the WMN, which leads to the constant adaptation of VNs in the WMN. Moreover, the model is able to capture the mean number of VNs available in the WMN according to the mean number of users accessing the WMN.

5.3.3.2 VN Update, Extension, Creation & Remotion

In the work of this Chapter, we recall the VN update and extension processes presented in Section 4.2. As an enhancement of these two control processes, users can now be connected to any VN selected according to the metric defined in expression (5.10), which considers certain ranges of variation for the requirements of users (or their flexibility), instead of being only connected to VNs that exactly fit these requirements.

Similarly to Section 4.2, when required to extend a VN to be incorporated in the node where a new user is connected to, such VN extension is performed through an available shortest path between the user location and the nearest point of attachment of the selected fitting VN for the user. In this VN extension process, the node where the user is connected to, first limits the candidate set of VNs discovered through the local or global VN discovery mechanisms, to the ones accessible in a minimum number of hops; this can enable the creation of a lower number of virtual nodes in a future VN extension. Then, such node selects the VN according to the metric defined in expression (5.10).

Since we aim to assess the impact of the VN selection metric on a dynamic WMN environment, it is important to introduce the control processes for VN creation and remotion. First, when an user arrives at the WMN, the exactly fitting VN for such user is created if none of the VNs available in the WMN fits the user requirements. This VN is created in the shortest path between the WMN gateway and the user location. Second, each VN end-point (a node with only one VN connection) has to periodically inspect its state; if there are no attached users in a specific timeout, such VN node is removed and its assigned resources are released. More details about these two VN control processes will be provided in Chapter 6.

5.3.3.3 Model Definition

In the following, we define an analytical model to derive the probability of each VN control process (VN update, VN extension or VN creation) to be triggered in a dynamic WMN environment. In such environment, a different number of users, characterized by distinct context requirements and flexibility of such requirements, are constantly arriving and leaving the WMN. We also model the mean number of VNs available in the WMN. In the model definition of this model, we will make use of the variables defined in Table 5.2.

- **Model Preliminaries**

Similarly to the analytical models proposed in Chapters 3 and 4, we consider a grid-based topology to place $|N|$ WMN nodes, which is represented in Fig. 5.2.

The arrival rate of users at the WMN is a Poisson process with mean inter-arrival time of $1/\lambda$. Each user $u \in U$ is connected to a random node $n_u \in N$, and his/her service time follows an exponential distribution with mean $1/\mu$. From the queueing theory, the mean number of users in the WMN is then defined by $\overline{U} = \lambda/\mu$.

The model will be left completely generic to be adapted to a distinct number of user context requirements (C) and their levels (M_c, $\forall c \in C$). For each user $u \in U$, the requested context level ($r_{u_c} \in R_u$) and the allowed variation range for such level ($t_{u_c} \in T_u$) are randomly chosen from the set of possible levels M_c ($\forall c \in C$).

For simplicity reasons, the central WMN node is considered to be the global WMN manager, which was introduced in this Thesis in Section 3.2.3.4. In this work, the global WMN manager: *(i)* contains information of every VN controller; *(ii)* is able to update the resources that need to be assigned to each VN node; *(iii)* is the node which starts, when necessary, a VN creation process up to the location of a new user that arrives at the WMN.

- **VN Creation & Number of VNs**

When the user $u \in U$ arrives at the node $n_u \in N$, the exactly fitting VN for the user u (v'_u) is created if none of the VNs available in the WMN, belongs to the set of fitting VNs for the user u (V_u). Or in other words, if there are no users already connected to VNs that belong to the set of fitting VNs for the user u.

Among the $\overline{U}$ users that are already accessing the WMN at the time of arrival of the user u, there may be users that can be only served by VNs belonging to the set V_u, users allowing associations to several VNs from the set V_u, and users not admitting connections to VNs from the set V_u. Since the requirements (and flexibility for context adaptation) of the user u are

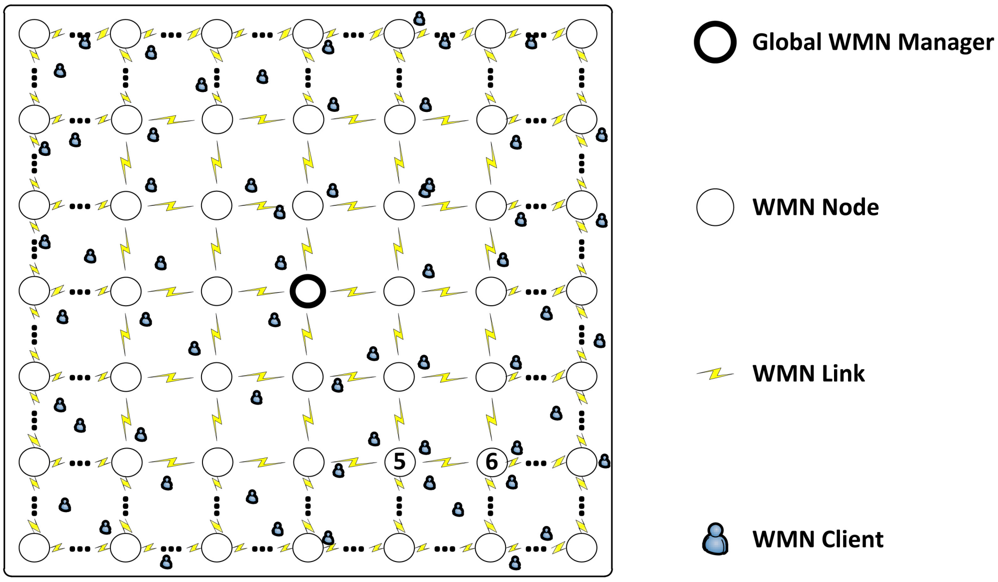


Figure 5.2: Grid-Based WMN Topology.

randomly assigned, the probability of a VN available in the WMN, to be part of the set V_u, P_u, is then defined by:

$$P_u = \frac{|V_u|}{|V|}, \tag{5.13}$$

where the number of possible VNs, $|V|$, is defined by $|M_c|^{|C|}$, since a VN can be created to meet any set of user context levels, and the number of those VNs that are fitting ones for the user u, $|V_u|$, is derived from the sets R_u and T_u.

Considering $\overline{V}$ as the mean number of VNs available in the WMN at the time of arrival of the user u (an expression for $\overline{V}$ will be defined below), the probability of the VN v'_u to be created in the WMN, P_{Cr_u}, can then be defined by the probability of none of these $\overline{V}$ VNs to be part of the set V_u, which is represented by $P_{Cr_u} = (1 - P_u)^{\overline{V}}$. Finally, considering any possible user $u \in U$ to access the WMN, each one characterized by the sets of requirements R_u and T_u, the probability of a VN creation process to be started in the WMN, P_{Cr}, is calculated by averaging all the probabilities P_{Cr_u} associated to the elements of the set U. P_{Cr} is then defined by:

$$P_{Cr} = \frac{1}{|U|} \times \sum_{u \in U} P_{Cr_u} = \frac{1}{|U|} \times \sum_{u \in U} (1 - P_u)^{\overline{V}}. \tag{5.14}$$

As previously referred, $\overline{V}$ is the mean number of VNs available in the WMN at the time of arrival of the user u at the WMN. $\overline{V}$ would be equal to $\overline{U}$ (mean number of ongoing users in the WMN), if all of these $\overline{U}$ users were associated to distinct VNs. However, each of these $\overline{U}$ users could have triggered a VN update or extension at the time of his/her arrival at the WMN (with the probability $1 - P_{Cr}$, since a VN update or extension is performed instead of a VN creation), which will reduce $\overline{V}$.

Moreover, since these $\overline{U}$ users arrived at the WMN at a different time, the probability of one of these $\overline{U}$ users to have triggered a VN update or extension, needs to be considered different from the same probability associated to another of these $\overline{U}$ users. This difference is related to the need to only consider the extensions or updates triggered in VNs that contain any of these $\overline{U}$ users (updates or extensions of VNs that contained users that already left the WMN cannot decrease $\overline{V}$, since such VNs were already removed). For instance, if $\overline{U} = 3$ users (identified by $\{u_1, u_2, u_3\}$) are connected to the WMN at the time of arrival of a new user, the number of possibilities to one of these 3 users to have triggered a VN update or extension is different: *(i)* the user u_1, which was the first user to arrive at the WMN, could not trigger the update or extension of any VN that contains users still connected to the WMN; *(ii)* the user u_2 could have triggered the update or extension of the VN in which the user u_1 is connected to; *(iii)* the user u_3 could have triggered the update or extension of the VNs associated to the users u_1 or u_2. According to this example, the mean number of possibilities, associated to each of these $\overline{U}$ users, to have triggered a VN update or extension, $P_{\overline{U}}$, is given by:

$$P_{\overline{U}} = \frac{1}{\overline{U}} \times \sum_{i=0}^{\overline{U}-1} i = \frac{\overline{U}-1}{2}. \tag{5.15}$$

Therefore, due to the VN updates or extensions that can be triggered in the WMN, the mean number of VNs available in the WMN, $\overline{V}$, has to be represented by the difference between: *(i)* the mean number of users accessing the WMN ($\overline{U}$); and *(ii)* the mean number of possibilities, associated to each of these $\overline{U}$ users, to have triggered a VN update or extension ($P_{\overline{U}}$), times the probability of occurrence of such VN update or extension processes $(1 - P_{Cr})$. $\overline{V}$ is then defined by:

$$\overline{V} = \overline{U} - P_{\overline{U}} \times (1 - P_{Cr}). \tag{5.16}$$

From (5.14) and (5.16), we can easily derive P_{Cr} and $\overline{V}$.

- **VN Update**

When the user $u \in U$ arrives at the node $n_u \in N$, the user u may trigger the update of any VN available in the node n_u, that belongs to the set of fitting VNs for the user u (V_u).

Due to the $\overline{U}$ users that are already accessing the WMN at the time of arrival of the user u, a VN update can occur if any of these $\overline{U}$ users is both: *(i)* connected to the node n_u, which has the probability $\frac{1}{|N|}$ (since these $\overline{U}$ users are randomly distributed in the WMN); *(ii)* associated to any VN belonging to the set V_u, which has the probability P_u (already defined in expression (5.13)). Moreover, a VN update can occur if any of these $\overline{U}$ users is assigned to any VN belonging to the set V_u, and the node n_u is within the path of such VN up to the global WMN manager (where the VN controllers are located), which has the probability α_n (an example for this probability will be provided below). According to this discussion, we can define the probability of one of these $\overline{U}$ users to allow the possibility to trigger a VN update at the time of arrival of the user u at node n_u; it is derived from the probability of any of the two situations described above to occur, being represented by $\frac{1}{N} \times P_u \times (1 + \alpha_n)$.

Considering any possible user from the ones already connected to the WMN ($\overline{U}$), the probability to trigger a VN update in the WMN at the time of arrival of the user u at node n_u, $P_{Up_{(n,u)}}$, is given by the complementary probability of any of these $\overline{U}$ users to not allow the possibility to trigger such VN update. $P_{Up_{(n,u)}}$ is then defined by:

$$P_{Up_{(n,u)}} = 1 - \left(1 - \frac{1}{N} \times P_u \times (1 + \alpha_n)\right)^{\overline{U}}. \tag{5.17}$$

As previously referred, α_n is the probability of the node n_u to be within the path of a fitting VN for the user u. Considering the 5×5 WMN of Fig. 5.2, α_5=0.5 is the probability of node 5 to be within the path of a fitting VN for the user u available in node 6, since there are two available shortest paths to establish a VN between node 6 and the global WMN manager, and only one of such paths makes use of node 5.

Considering any possible user $u \in U$, each one characterized by the sets of requirements R_u and T_u, and any possible node $n_u \in N$, the probability of a VN update process to be started in the WMN, P_{Up}, is calculated by averaging all the probabilities $P_{Up_{(n,u)}}$ (see expression (5.17)) associated to the elements of the sets U and N. P_{Up} is then defined by:

$$P_{Up} = \frac{1}{|N| \times |U|} \times \sum_{n \in N} \sum_{u \in U} P_{Up_{(n,u)}}. \tag{5.18}$$

- **VN Extension**

When the user $u \in U$ arrives at the node $n_u \in N$, and a VN belonging to the set of fitting VNs for the user u (V_u) is not available in the node n_u, but it exists in another node $n \in N$, such VN needs to be extended up to the node n_u.

Based on the presented formulation for the VN creation and VN update processes, it is easily inferred that the probability of a VN extension to be triggered by the user u arriving at the node n, $P_{Ex_{(n,u)}}$, is given by the probability of the user u to not trigger a VN creation (with probability P_{Cr_u}) or a VN update (with probability $P_{Up_{(n,u)}}$) at the node n. $P_{Ex_{(n,u)}}$ is then defined by:

$$P_{Ex_{(n,u)}} = 1 - P_{Cr_u} - P_{Up_{(n,u)}}. \tag{5.19}$$

Considering any possible user $u \in U$, each one characterized by the sets of requirements R_u and T_u, and any possible node $n \in N$, the probability of a VN extension process to be started in the WMN, P_{Ex}, is calculated by averaging all the probabilities $P_{Ex_{(n,u)}}$ (see expression (5.19)) associated to the elements of the sets U and N. P_{Ex} is then defined by:

$$P_{Ex} = \frac{1}{|N| \times |U|} \times \sum_{n \in N} \sum_{u \in U} P_{Ex_{(n,u)}}. \tag{5.20}$$

5.3.4 Evaluation & Discussion

In the following, we resort to the analytical model defined in Section 5.3.3, to evaluate the impact of using distinct levels of user context flexibility on the decrease of the need to create new VNs in the WMN.

Using the simulation framework presented in Section 3.4, we also extend the distributed VN control and management mechanism proposed in Chapter 4, to incorporate the context-aware VN selection metric defined in expression (5.10), and the basic processes for VN creation and remotion. This way, we can assess the accuracy of the analytical results by comparing them against the simulation ones.

Finally, we also make use of the simulation platform to implement several of the context mapping rules presented in Section 5.2.1, and to compare our distributed VN control and management mechanism against other distributed approaches.

Table 5.3: Modeling & Simulation Details: Impact of User Context on VN Selection.

Parameter	**Value**
$\lvert N\rvert$	25 (5×5 Grid WMN)
WMN Node Interface Range	$100m$ ($500m \times 500m$ of Simulated Area)
WMN Node Interface Capacity	$54Mb/s$ ($1Mb/s$ is for Control Purposes)
WMN Node Buffer Size	500
n_u	Randomly chosen from the set N
$/U$	$\{5, 10, 15, 20, 25, 30\}$
User Connectivity Time	Exponential Distribution with Mean $1/\mu = 1/20$
User Arrival Rate	Poisson Distribution with Mean Inter-Arrival Time of $1/(\overline{U}\times\mu)$
$\lvert C\rvert/M_c$ $(c \in C)$	$4/\{1, 2, 3, 4, 5\}$
$r_{u_c} \in R_u$ $(c \in C)$	Randomly chosen from the set M_c
$t_{u_c} \in T_u$ $(c \in C)$	Randomly chosen from the set $\{0, 1, 2\}$
c_1=User Service Rate	Arrival Rate of VN Flows: $\{64, 96, 128, 256, 512\}Kb/s$
c_2=User Service Payload	Payload of VN Flows: $\{64, 96, 128, 256, 512\}Bytes$
c_3=User Security Requirements	Security Header of VN Packets (EH): $\{0, 22, 34, 54, 74\}Bytes$
	Extra E2E delay of VN flows (ED): $\{0, 5, 10, 15, 20\}ms$
c_4=User Price Preferences	MAC priority level assigned to a VN (P): $\{0, 1, 2, 3, 4\}$
VN Creation/Remotion Timeout	$300ms/500ms$
Simulation Time/Runs	$1000s/20$

5.3.4.1 Evaluation Scenario: Details & Variables

We start by defining all the variables and details of this study, which are summarized in Table 5.3. We create a grid-based WMN topology with $|N| = 5 \times 5$ nodes. Each WMN node has a transmission radius of $100m$ ($500m \times 500m$ of simulated area), one physical interface of $54Mb/s$ and buffer size of 500 packets.

According to the discussion of Section 3.4.3, in which we presented the details of the multi-VN architecture implemented in NS-2, the network virtualization is emulated by dynamically assigning a distinct number of "virtual" interfaces to each WMN node according to the number of VNs that it supports. Then, the capacity and buffer size of each physical WMN node are shared among its VN nodes (or "virtual" interfaces) according to the resource needs of such VN nodes. Moreover, the interface assigned to a VN uses its dedicated wireless channel. Finally, one "virtual" interface of $1Mb/s$ is assigned to all WMN nodes for control purposes, and the AODV routing protocol [36] is used for both data and control.

The arrival rate of users at the WMN is a Poisson process with mean inter-arrival time of $1/\lambda$. Each user $u \in U$ arrives at a random WMN node $n_u \in N$, and he/she is connected for a time following an exponential distribution with mean $1/\mu = 1/20$. From the queueing theory, the mean number of users in the WMN is then defined by $\overline{U} = \lambda/\mu = \lambda/20$; thus, we configure λ to perform tests for a mean number $\overline{U} = \{5, 10, 15, 20, 25, 30\}$ of users accessing the WMN. Finally, in the simulation environment, the user communication target is randomly chosen among the nodes from the VN in which he/she is connected to.

We consider $|C| = 4$ user context parameters to drive the architecture. The set of levels of each context parameter $c \in C$ is $M_c = \{1, 2, 3, 4, 5\}$. For each user $u \in U$, the requested level on the context parameter $c \in C$ ($r_{u_c} \in R_u$) is randomly chosen from the set M_c of levels, and the flexibility associated to the demand of such context parameter ($t_{u_c} \in T_u$) is randomly chosen from the set $\{0, 1, 2\}$.

We consider the set C=\{*Service Rate*, *Service Payload*, *Security Requirements*, *Price Preferences*\} of user context parameters. Based on the discussion of Section 5.2.1, we implement, in the simulation platform, rules to model the user context requirements in levels, and to map such levels in proper VN features. These rules are described in the paragraph below.

First, we consider 5 levels of rate and payload of the user requested services in order to differentiate the type of flows running at each VN (please see Table 5.3). With respect to the user security requirements, we consider 5 IPSec-compliant security levels, each one characterized by distinct authentication and/or encryption mechanisms: \{*No_Security*, *DES*, *3DES+SHA*, *3DES+SHA+IPv4_Tunnel*, *3DES+SHA+IPv6_Tunnel*\}. Following the strategy presented in Section 5.2.1.2, we emulate a VN security level by configuring the VN packets with an extra security header (EH, as defined in expression (5.5)) and the VN communications with an extra E2E delay (ED, as defined in expression (5.6)); therefore, Table 5.3 presents the values assigned to the EH and ED parameters associated to each one of the 5 VN security levels, being these values derived from the information available in [299]. Finally, according to the discussion of Section 5.2.1.4, the NS-2 IEEE 802.11e module [284] is used to assign a distinct MAC priority level (P, as defined in expression (5.8)) to the WMN channel assigned to a VN based on the price preferences of the users connected to such VN; in the simulation platform, $P = 4$ is the highest MAC priority level, as can be seen in Table 5.3.

Still in the NS-2 environment, the distributed VN control and management mechanism proposed in Chapter 4, has now the following features: *(i)* each WMN node is able to select any of its supported VNs to be updated according to the VN selection metric defined in expression (5.10); *(ii)* the candidate set of fitting VNs for the user $u \in U$ to be locally extended up to the node $n_u \in N$ in which the user $u \in U$ is connected to, is limited to the VNs located at a maximum distance of 2 hops to the node n_u; *(iii)* the global WMN manager is the central WMN node, and it can decide to trigger a global VN extension or a creation; *(iv)* the creation of the exactly fitting VN for the user $u \in U$, v'_u, is triggered when no reply is received by a distributed VN discovery in the timeout of $300ms$; *(v)* the timeout to trigger the remotion of VNs when there are no attached users is $500ms$.

In order to compare our distributed VN discovery mechanism against other distributed approaches, we also implement two other distributed VN discovery mechanisms. First, we implement a mechanism to discover a point of attachment of any fitting VN to assign to an user, by using a similar strategy than the AODV routing protocol [36]. In this AODV-based VN discovery mechanism, if an user arrives at a WMN node that does not support any fitting VN for the user, this WMN node broadcasts a VN discovery request in the WMN. The VN discovery request message includes the user context, and therefore, any WMN node, after receiving the request, is able to check if it supports any fitting VN for the user (selected according to the metric defined in expression (5.10)); if it is true, this WMN node sends a VN discovery reply, in a unicast manner, to the WMN node where the user is connected to. Second, we implement a distributed VN discovery mechanism based on the original NS-2 Bamboo-DHT [285], where a set of Bamboo keys are representative of the VNs available in the WMN; then, the Bamboo overlay is used to discover a point of attachment of any fitting VN for an user, if the user arrives at a WMN node that does not support any fitting VN for the user. In both AODV- and Bamboo-based schemes, the discovered VN is extended up to the user location. Such extension is performed through an available shortest path between the user location and the nearest WMN node that supports the discovered VN.

Finally, each simulation run lasts $1000s$, and each obtained value is the mean of 20 runs with a confidence degree of 90%.

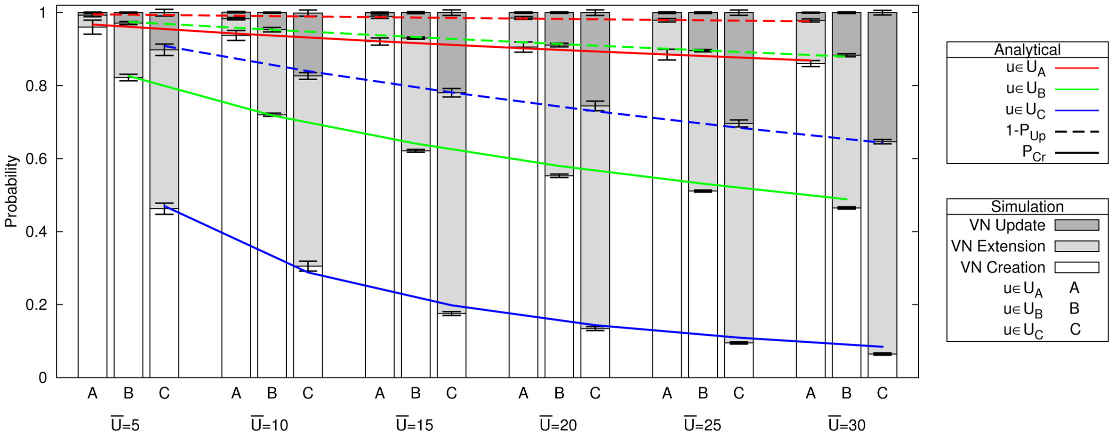


Figure 5.3: Analytical & Simulation Models: Probability of a VN Update, Extension or Creation to be triggered in the WMN, when varying the Mean Number of Users in the WMN, being considered Distinct Levels of User Context Flexibility.

5.3.4.2 Model Validation

In this Section, we validate the analytical results against the ones obtained from the simulation study.

- **Results**

In Fig. 5.3, we show the probability of each VN control process (VN update, VN extension or VN creation) to be triggered when a specific user arrives at a WMN, where a distinct number of $\overline{U}$ users are already connected to.

In detail, Fig. 5.3 presents, for each $\overline{U}$ value, the simulation probabilities in cumulative bars, where a distinct color (dark gray, light gray and white) is representative of the probability associated to each VN control process (VN update, VN extension or VN creation, respectively). In addition, Fig. 5.3 presents, for each $\overline{U}$ value, the analytical probabilities in lines, where the solid line represents the VN creation probability (P_{Cr}), and the dotted line represents the cumulative VN creation and extension probability (or $1 - P_{Up}$).

In order to assess the impact of using distinct levels of context-aware flexibility of users (represented by the set T_u) to be connected to VNs that do not exactly fit their context requirements (represented by the set R_u), we also split the set of possible users to be connected in the WMN, U, in three distinct sub-sets: U_A, U_B and U_C. To form these three sub-sets, we first define the overall context-aware flexibility of a specific user $u \in U$ by $\sum t_{u_c}$ ($\forall c \in C$); since these experiments assume that each $t_{u_c} \in T_u$ can take any value from the set $\{0, 1, 2\}$, the minimum value of $\sum t_{u_c}$ ($\forall c \in C$) is 0 and the maximum is 8 (for $|C| = 4$). Therefore, we make use of the distinct values that $\sum t_{u_c}$ ($\forall c \in C$) can take, in order to characterize the sub-sets U_A, U_B and U_C, as the ones that respectively group users that are characterized by small, medium and high overall context-aware flexibility. This differentiation of the set U of users, based on the overall context-aware flexibility of users, is then defined by:

$$U_A = \{u \in U | 0 \leq \sum_{c \in C} t_{u_c} \leq 2\}, \tag{5.21}$$

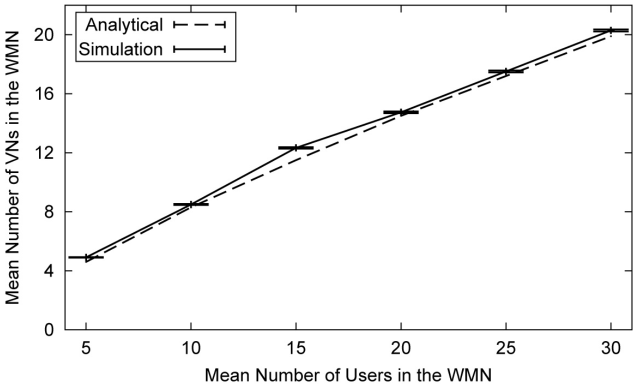


Figure 5.4: Analytical & Simulation Models: Mean Number of VNs in the WMN, when varying the Mean Number of Users in the WMN.

$$U_B = \{u \in U | 3 \leq \sum_{c \in C} t_{u_c} \leq 5\}, \tag{5.22}$$

$$U_C = \{u \in U | 6 \leq \sum_{c \in C} t_{u_c} \leq 8\}. \tag{5.23}$$

In the analytical model, the probability of each VN control process to be triggered by users that belong to the sub-sets U_A, U_B and U_C, is differentiated by considering the number of users that belong to these sub-sets, instead of the total number of elements of the set U. In the simulation environment, this differentiation is performed as following: for instance, considering the sub-set U_A, we collect the number of times that each VN control process is triggered when any user $u_A \in U_A$ arrives at the WMN, and divide such value by the number of elements from the set U_A that access the WMN during the simulation.

Finally, Fig. 5.4 shows the mean number of VNs available in the WMN when an user arrives at the WMN, where a distinct number of $\overline{U}$ users are already connected to.

- **Discussion**

From the obtained results, we can clearly validate the analytical model. Its results are very similar to the simulation framework, and the minimal discrepancies between them were expected due to the: *(i)* randomness of the considered variables; *(ii)* collisions between packets in NS-2 that are not considered in the model; *(iii)* finite number of values per simulation run which may not exactly match the probability of each process.

We can state that the increase of the mean number of users accessing the WMN, $\overline{U}$, will decrease the probability of VN creation (see Fig. 5.3), since there are more VNs in the WMN, and therefore, the probability to update or extend fitting VNs to assign to arriving users will be higher. This fact can also be observed in Fig. 5.4, since the tendency of increasing of the mean number of VNs, $\overline{V}$, is lower when there are more users $\overline{U}$ accessing the WMN. By increasing the $\overline{U}$ value, the increase of the VN update probability is higher than the one of VN extension probability, since updates have priority over extensions.

Moreover, it should be noticed that the VN creation probability should be small, especially in a dynamic WMN environment where a high number of users are constantly arriving and

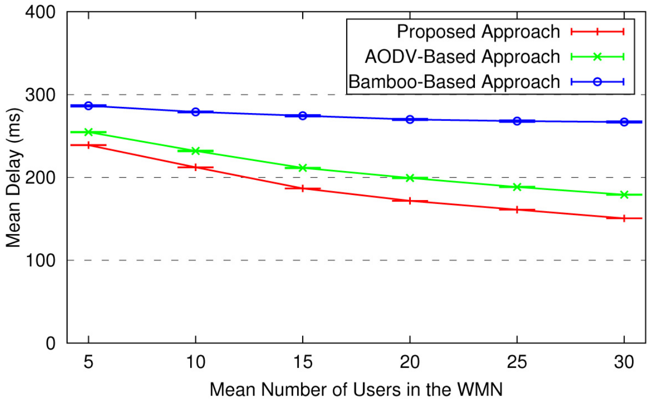


Figure 5.5: Simulation Model: Mean Signaling Delay introduced by each Control Approach, when varying the Mean Number of Users in the WMN.

leaving the WMN, since it is a complex process that imposes the need to create a high number of new virtual nodes. In Fig. 5.3, we can observe that the overall context-aware flexibility of users, to be connected to partially fitting VNs, influences the VN creation probability: *(i)* users with strict requirements ($u_A \in U_A$) will very often trigger the creation of exactly fitting VNs, since they only accept associations to a small number of VNs; *(ii)* users allowing high variation ranges of their demands ($u_C \in U_C$) will trigger a small number of VN creations. More specifically, in Fig. 5.3, we can see that the probability of a VN creation to be triggered by an user $u_c \in U_C$ is very small (less than 0.1), when a mean number of 25 or 30 users are connected to the WMN. Therefore, these results are able to show the advantages of considering a high level of user context-aware flexibility in the VN control and management mechanism, since it allows to increase the VN usability in dynamic WMN environments, while decreasing the need to create new VNs to serve single users and its associated complexity.

5.3.4.3 Comparison of Distributed Approaches for VN Control & Management

In this Section, we evaluate the signaling delay and overhead, as well as the reconfiguration costs, of the proposed distributed control mechanism to update, extend or create a VN to assign to an user, according to the simulation setup defined in Section 5.3.4.1.

Moreover, we compare these results against the ones obtained with the two other distributed approaches for VN control and management that we implement in the NS-2 simulator. These approaches consider other schemes to discover a point of attachment of any fitting VN to assign to an user. One scheme uses a similar strategy than the AODV routing protocol, while the other uses the Bamboo overlay for VN discovery (please see more details of these schemes in Section 5.3.4.1).

- **Results**

In Fig. 5.5, we show the mean signaling delay of the three distributed control mechanisms to update, extend or create a VN to assign to an user that arrives at the WMN, where a distinct number of $\overline{U}$ users are already accessing the WMN. These delays do not contemplate the processing delays associated to the WMN nodes in which it is required to create new virtual nodes to extend or create VNs, since we are not able to measure them in NS-2.

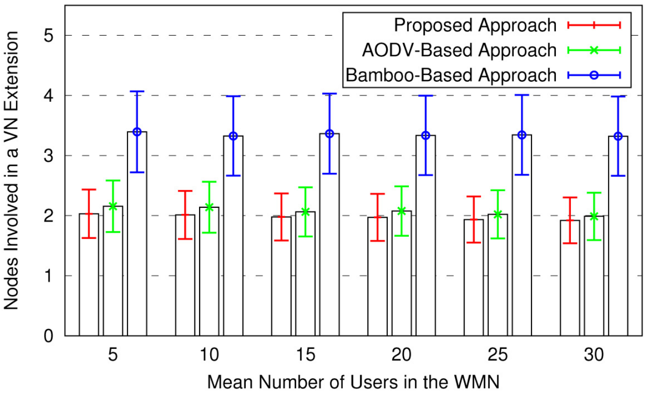


Figure 5.6: Simulation Model: Mean Number of Nodes involved in a VN Extension, when varying the Mean Number of Users in the WMN.

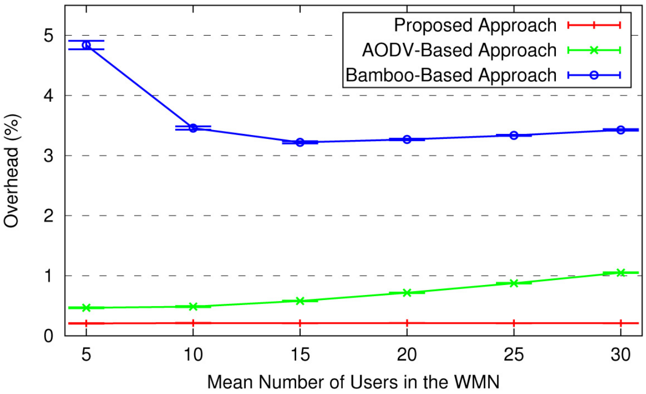


Figure 5.7: Simulation Model: Mean Signaling Overhead introduced by each Control Approach, when varying the Mean Number of Users in the WMN.

In Fig. 5.6, for each distributed approach for VN control and management, we present the mean number of WMN nodes that need to be reconfigured when it is triggered a VN extension in the WMN. Finally, Fig. 5.7 shows the overhead introduced by the signaling control processes in the three distributed approaches for VN control and management, which is calculated in relation to the total amount of traffic that flows in the WMN.

- **Discussion**

The mean signaling delay to discover and adapt (or create) a fitting VN to assign to an user is smaller in our control approach than in the AODV- or Bamboo-based solutions (see Fig. 5.5). In our approach, each WMN node can quickly inspect the VNs available in the physical neighbors located at the maximum distance of 2 hops, instead of starting a AODV-based or Bamboo-based VN discovery process in the WMN. The AODV-based VN discovery process has a broadcast nature that can lead to a higher number of collisions, and so, higher delays. The Bamboo-based VN discovery process uses the Bamboo overlay that can lead to a higher number of communications to find the desired VN.

Moreover, in our approach and in the AODV-based solution, the user association delays decrease with the increase of the mean number of users accessing the WMN, $\overline{U}$. With the increase of $\overline{U}$, there are more VNs available in the WMN, and so, there is a higher probability to find a fitting VN in a node near the user location, which decreases the VN discovery delays of these two schemes. However, the tendency of delay decreasing is lower, when a high number of users access the WMN ($\overline{U} = 25$ or $\overline{U} = 30$), since a higher number of control processes are simultaneously triggered in the WMN, which increases the number of network collisions.

Concerning the Bamboo-based solution, the user association delays also decrease when there are more users $\overline{U}$ accessing the WMN, since there are more VNs available in the WMN, which increases the probability to discover the desired VN in a lower number of communications. Nevertheless, in the Bamboo-based solution, the tendency of delay decreasing is lower than in the two other solutions, since the number of communications of the Bamboo-based VN discovery process is variable, increasing the VN discovery delays.

According to the results of Fig. 5.6, we can conclude that our control mechanism requires the reconfiguration of a smaller number of WMN nodes to perform VN extensions than the others, which reflects the reduction of the VN reconfiguration costs. However, while the results of the AODV-based solution are very similar to our approach, the mean number of reconfigured WMN nodes in the Bamboo-based solution is extremely high. Due to the random distribution of the Bamboo keys in the WMN, the point of attachment of a fitting VN for the user cannot be discovered near the user location, which can lead to the reconfiguration of a higher number of nodes to extend the discovered VN up to the user location.

Finally, focusing on the signaling control overhead (see Fig. 5.7), we can see that the proposed approach for VN control and management requires a lower amount of control overhead than the AODV-based solution, which is explained by the broadcast nature (and its associated network flooding) of the AODV-based VN discovery process. Moreover, the signaling control overhead is clearly higher for the Bamboo-based solution than for the others, mainly due to the higher amount of maintenance traffic of the Bamboo overlay. Finally, in the Bamboo-based solution, the signaling control overhead has a decreasing tendency when there is a small number of users accessing the WMN. This behavior reflects the higher importance of the control traffic required to maintain the Bamboo overlay, when compared to the data traffic that flows in a WMN with a small number of mesh clients.

5.4 QoE-Aware Reinforcement Learning VN Routing

Tightly coupled with the proper configuration of the context-aware mechanisms and protocols in a specific VN, this Section presents and evaluates a reinforcement learning strategy to be applied in the definition of a QoE-aware VN routing scheme (published in [19]). Our main goal is to provide a routing strategy that autonomously adjusts the routes among two distinct virtual nodes from a VN, in order to maximize the overall QoE for VN users in an adaptive fashion, and in response to VN topology or WMN resource availability dynamics.

Reinforcement learning [300] is an unsupervised learning technique that allows an agent to autonomously perform actions based on the sensed system information. The result of these actions is then communicated to the agent, as a reinforcement signal or reward. The main purpose of reinforcement learning is to automatically find a policy that maximizes the long-term reward for the agent. Routing can be seen as a multi-agent reinforcement learning problem: *(i)* initially, the routing agent of a network node has no (topology) information available, starting an exploration phase, where it learns about available network paths and their quality; *(ii)* these paths will then be used in the exploitation phase, where the routing agent makes use of the high quality network paths; *(iii)* sporadically, the routing agent switches to the exploration phase in order to update the quality of other available paths.

In the context of the VN routing strategy, we bring together the potentials of reinforcement learning and QoE-awareness. In Section 5.2.1, we discussed how to derive the QoE perceived by an user based on the context features of the VN where such user is connected to (please see expression (5.9)). At this stage, we now aim to enable the exchange, among the virtual nodes from a VN, of light-weight feedback signals related to the QoE perceived by the ongoing VN users. These signals are conveyed in the headers of data or control packets that flow in the VN, and they are then used by a reinforcement learning technique, in order to allow each VN node to preferably forward VN flows through the next hops that offer the best expected QoE towards a specific destination. Finally, we consider an exploration probability of using other VN paths, rather than the expected best ones, to obtain a broader VN knowledge and to dynamically explore better paths in terms of QoE support.

5.4.1 Double QoE-Aware Reinforcement Learning Mechanism

In our approach, the virtual nodes from each VN cooperate through a light-weight knowledge dissemination process to acquire and update specific QoE-aware learning information. This information then allows each VN node to select, on-demand, the most suitable next hop for a certain flow, enabling the fast detection of the quality degradation of VN links.

In this QoE-aware VN routing strategy, we consider a double learning mechanism [301] to disseminate the QoE-related information in the VN, which allows each VN node to acquire a higher level of QoE-related information available in the VN. More specifically, each VN node, within a path of a VN flow from the source s to the destination d, is able to: *(i)* be informed by its selected next hop about the expected QoE of the path towards d, which is denoted by *Backward Learning* (BL) mechanism; *(ii)* inform its selected next hop about the expected QoE of the path towards s, which is denoted by the *Forward Learning* (FL) mechanism.

With this broader knowledge of QoE-related information, each VN node, within a path of a VN flow from the source s to the destination d, can take advantage of the BL mechanism to perform a more accurate selection of the best next hop to reach the destination d. In addition, a specific VN node can take advantage of the FL mechanism to immediately select a path to assign to a new VN flow, if the destination of such flow is the source of another flow already established in the VN. In the following, we detail the processes and functions involved in this double QoE-aware reinforcement learning mechanism for VN routing.

5.4.1.1 Reinforcement Function

In a traditional reinforcement learning strategy, an agent interacts with its system in discrete time steps. At each step, the agent receives an observation of the system, and based on the system information, it has to chose an action from the set of actions available. This action will move the system to a different state, and in the following system observation, the agent can determine the reward associated to the transition of the system from one state to another. This reward is derived from the application of a reinforcement function.

In our reinforcement learning VN routing strategy, we define the reward associated to the new state t of VN node i when VN node i performs a certain action, as "*the expected QoE* *value of the remaining path from VN node i to the destination d, when VN node i selects the VN node h as the next hop towards the destination d*", being represented by $QoE^{t}_{i\to d,h}$. To obtain this reward, we make use of the expression (5.9) to derive the QoE perceived by an user connected to a specific VN. Moreover, it is important to derive the reward associated to the selection of any possible next hop of VN node i towards the destination d, in order to allow VN node i to select the best next hop towards the destination d; in our strategy, we define the entire set of possible next hops of VN node i towards the destination d as $H_{i\to d}$. Finally, the reinforcement function employed to calculate the reward $QoE^{t}_{i\to d,h}$ associated to a specific state transition, is derived from the original Q-Learning algorithm [300], which has the advantage that it does not need a model of the system to learn from delayed reinforcement. Therefore, for each $h \in H_{i\to d}$, the function to derive $QoE^{t}_{i\to d,h}$ is defined by:

$$QoE^{t}_{i\to d,h} = QoE^{t-1}_{i\to d,h} + \alpha\big[(QoE^{t}_{i\to h} + QoE^{aux}_{h\to d}) - QoE^{t-1}_{i\to d,h}\big], \qquad (5.24)$$

where:

- $QoE^{t-1}_{i\to d,h}$: expected QoE value to reach the destination d based on the previous state $t-1$ of VN node i, when VN node i selected VN node h as the next hop towards the destination d;

- $QoE^{t}_{i\to h}$: local QoE value of the connection between VN node i and the respective next hop (VN node h) for the current state t;

- α: *Learning Rate*, which is associated to the importance given to a future action made by VN node i. α is intrinsically related to the convergence level of the learning strategy, as will be described in Section 5.4.1.3;

- $QoE^{aux}_{h\to d}$: *Discount Factor*, which considers the previous and current expected QoE values to reach the destination d, when VN node i selected VN node h as the next hop towards the destination d. More details about the discount factor will be provided in Section 5.4.1.4.

With the QoE-related information acquired by VN node i (the required control mechanism to measure and disseminate this QoE-related information will be detailed in Section 5.4.1.2), it can then use expression (5.24) to determine the set of expected QoE values of the remaining paths towards the destination d when selecting any $h \in H_{i\to d}$. In the following, VN node i can update the reinforcement learning strategy within the VN, since it can: *(i)* determine the best next hop to reach the destination d; *(ii)* inform its previous hop, within the flow from the source s to the destination d, about the best expected QoE that it can offer to reach the destination d. To perform these two processes, VN node i has to select the best next hop to reach the destination d as the one that maximizes the reward $QoE^{t}_{i\to d,h}$ (see expression (5.24)), which is defined by:

$$QoE^{t}_{i\to d} = \max_{h\in H_{i\to d}} \{QoE^{t}_{i\to d,h}\}. \qquad (5.25)$$

Please note that the above explanation is focused on the BL mechanism, in which a VN node i, within a path of a VN flow from the source s to the destination d, can calculate the expected QoE values to reach the destination d, based on the QoE-related information sent by the selected next hops towards the destination d. However, a very similar scheme can be

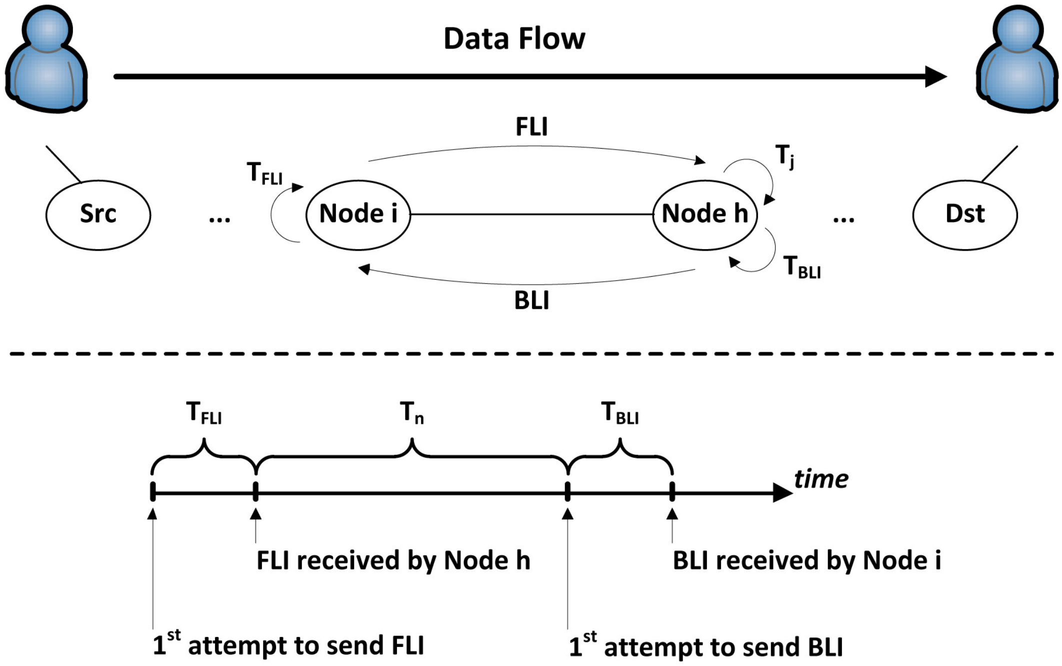


Figure 5.8: Double QoE-Aware Reinforcement Learning Mechanism.

in place by using the FL mechanism. In this case, the VN node i can derive the expected QoE values to reach the source s, based on the QoE-related information sent by its previous hops within the path from the source s to the destination d. In the following, we detail these BL and FL mechanisms.

5.4.1.2 QoE Measurement & BL/FL Schemes

In order to implement the double reinforcement learning approach, a VN node i, within a path used by a VN flow with source s and destination d, periodically updates the information about the: *(i)* quality of the path towards the source s and disseminates this information to its selected next hop (FL mechanism); *(ii)* quality of the path towards the destination d and disseminates this information to its previous hop (BL mechanism). Fig. 5.8 depicts the proposed learning approach that is explained below.

In the beginning, VN node i sends to its selected next hop (VN node h) an empty *Forward Learning Information* (FLI) element that triggers the reinforcement learning mechanism for the VN flow in the VN link $L_{i \to h}$. Such FLI element is carried in the header of the first VN data packet sent by VN node i to VN node h. Due to possible packet retransmissions, the process of sending the FLI element is associated to the time-window T_{FLI}.

From the moment that VN node i first attempts to send the FLI element to VN node h, it begins to store local QoS-related metrics: buffer delay, data transmission and reception times, number and importance of lost packets. This information is then used to determine the expected QoE in the VN link $L_{i \to h}$, $QoE^t_{i \to h}$. The same action is performed by VN node h after receiving the FLI element, and during a period T_n that may depend on the VN context features. In this case, the local QoS-aware knowledge will be used to determine the expected

QoE in the reverse VN link $L_{h\to i}$, $QoE^t_{h\to i}$.

After the period T_n, VN node h sends to VN node i the information about the delay of its buffer, as well as the number and importance values of lost packets. Moreover, VN node h sends the best expected QoE value to reach the destination d, $QoE^t_{h\to d}$, that it has stored. This QoE value is determined through the expression (5.25), after applying the QoE-aware learning scheme over its next hops towards the destination d ($H_{h\to d}$). In this process, VN node h encapsulates this information in a *Backward Learning Information* (*BLI*) element, which is carried as an additional header field in the MAC-ACK of the last received VN data packet from VN node i. If the next VN data packet, received by VN node h from VN node i, does not include a *FLI* element, VN node h updates its local information and tries to send the *BLI* element again (this process lasts the time-window T_{BLI}).

As soon as VN node i receives the *BLI* element, it determines the expected QoE of the path to reach the destination d through VN node h, $QoE^t_{i\to d,h}$, using the reinforcement function defined in expression (5.24). In the following, VN node i uses the expression (5.25) to update the best expected QoE value to reach the destination d, $QoE^t_{i\to d}$, taking into account the rewards received after applying the QoE-aware learning scheme over its next hops towards the destination d ($H_{i\to d}$).

Finally, the overall mechanism described above is repeated. However, the *FLI* element sent by VN node i to its selected next hop (VN node h) also includes the information related to the quality of the path between VN node i and the source s, $QoE^t_{i\to s}$. After receiving the *FLI* element, VN node h can apply the expressions (5.24) and (5.25) to respectively refresh the quality of the path to reach the source s through VN node i, $QoE^t_{h\to s,i}$, and the best expected QoE value to reach the source s, $QoE^t_{h\to s}$.

5.4.1.3 Learning Rate

In the Q-learning algorithm, the learning rate (α) determines the importance given to a future action based on the knowledge acquired by a learning agent, which leads to a compromise between risky decisions or more grounded ones. Therefore, α is directly related to the convergence level of the learning strategy, and, in the context of the proposed VN routing strategy, α is directly related to the convergence of stable paths.

Considering our QoE-aware routing strategy, we suggest that α must be dependent on the difference among the actual expected QoE values of the possible next hops towards a certain destination. If the available VN paths have equal qualities, then exploiting different paths does not make much sense; in this case, α should be higher. On the other hand, if the VN path quality is significantly different, we should more aggressively exploit the best path; in this case, α should be decreased in order to give more importance to the best next hop.

According to the above discussion, please consider that a certain VN flow is using VN node i to reach the destination d, and VN node i has the set of possible next hops $H_{i\to d}$ to reach the destination d. Therefore, the formula used by VN node i to determine α, which is part of the reinforcement function defined in expression (5.24), is defined by:

$$\alpha = 1 - \frac{\max\limits_{h\in H_{i\to d}} \left\{ exp\left(QoE_{i\to d,h}\right)\right\}}{\sum\limits_{h\in H_{i\to d}} exp\left(QoE^t_{i\to d,h}\right)}, \tag{5.26}$$

$$QoE_{i\to d,h} = QoE^t_{i\to h} + QoE^t_{h\to d}, \tag{5.27}$$

where $QoE^{t}_{i \to h}$ is the last QoE value of the VN link $L_{i \to h}$ determined by VN node i, and $QoE^{t}_{h \to d}$ refers to the last received information of the expected QoE of the path to reach the destination d, when using VN node h.

5.4.1.4 Discount Factor

In the Q-learning algorithm, the discount factor allows a learning agent to infer the reliability of a future action, which could be computed based on the previous actions.

In our QoE-aware reinforcement learning VN routing strategy, each VN node i, within a path of a VN flow from the source s to the destination d, stores the last Z received information updates of the expected QoE to reach the destination d when using a certain next hop (VN node h). This way, VN node i can update its reinforcement learning strategy by considering the background of VN node h, which can increase the reliability of a future action.

According to the above discussion, the formula used by VN node i to determine the discount factor associated to the selection of a specific next hop (VN node h) towards the destination d, $QoE^{aux}_{h \to d}$, which is part of the reinforcement function defined in expression (5.24), is based on the mean of the last Z received information updates, being defined by:

$$QoE^{aux}_{h \to d} = \sum_{z=0}^{Z-1} \frac{QoE^{t-z}_{h \to d}}{Z}, \tag{5.28}$$

where $QoE^{t-z}_{h \to d}$ refers to the received information of the expected QoE of the path to reach the destination d when using VN node h, in the previous $z < Z$ intervals (or states).

5.4.1.5 Exploration Probability

A reinforcement learning agent, beyond exploiting effective actions, also explores new ones in order to make better actions in the future. This way, we consider exploration to reach a specific destination using other feasible paths rather than the expected best one. Using exploration, new VN routes are evaluated, propelling the dissemination of VN information about possible better VN links in order to gather a more comprehensive VN knowledge.

Considering our QoE-aware routing strategy, each VN node, beyond the shortest path towards a specific destination, also has to maintain a set of alternative VN routes to the same destination, in order to support a probabilistic exploration according to their quality. However, when VN routes become too long, excessive delays will occur and packet loss probability will also increase. Therefore, we define an upper bound for the hop count of each possible VN path that a packet may take, depending on the number of hops of the shortest path between the source and the destination of such path.

According to the above discussion, each VN node i, within a path of a VN flow from the source s to the destination d, has only available a set of candidate next hops towards the destination d. Each candidate next hop (VN node h) is part of a VN path that must satisfy the following condition:

$$h \in H_{i \to d} : |(s \to d, h)| \leq (1 + \beta_1)|(s \to d)'| \ , \ \ 0 \leq \beta_1 \leq 1, \tag{5.29}$$

where: *(i)* $|(s \to d, h)|$ is the number of hops already traversed between the source s and VN node i, plus the number of hops of the path between VN node i the destination d, when VN node i selects VN node h as next hop; *(ii)* $|(s \to d)'|$ is the number of hops of the shortest path between the source s and the destination d; *(iii)* β_1 is the factor that allows to determine the maximum tolerance in the number of hops of the selected VN path between the source s and the destination d, with respect to the number of hops of the shortest path between the source s and the destination d.

Considering this approach, when VN node i has to forward a packet towards the destination d, it must consider an exploration probability to send the packet through all the possible next hops that belong to the set $H_{i\to d}$ and satisfy the condition defined in expression (5.29). Therefore, the probability of VN node i to forward a VN packet towards the destination d via VN node h, $p_{i\to d,h}$, is defined by:

$$p_{i\to d,h} = \frac{exp\left(\frac{QoE^t_{i\to d,h}}{T_e}\right)}{\sum\limits_{h\in H_{i\to d}} exp\left(\frac{QoE^t_{i\to d,h}}{T_e}\right)}, \tag{5.30}$$

which is a probability adapted from the Boltzmann-action selection [302]. Following the same principle, T_e represents the temperature that allows to control the exploration probability according to the VN context features. Moreover, since the routes assigned to a specific VN flow tend to converge during the flow lifetime (which is defined by Δt), we decrease the exploration level over time. Therefore, T_e is defined based on the following decay function:

$$T_e = T_{e0} \times \rho^{-\Delta t} \ \ , \ \ \rho > 1, \tag{5.31}$$

where: *(i)* T_{e0} refers to the initial value assigned to T_e, when the flow is established in the VN; *(ii)* ρ is the scaling factor that sets the rate of decay of T_e.

5.4.2 Evaluation & Discussion

In the following, we resort to the simulation framework presented in Section 3.4, to evaluate the impact of the QoE-aware learning routing strategy on the mean perceived QoE of a VN user and on the VN routing control overhead.

We first perform several experiments to optimize the tunable parameters of the learning routing scheme when applied to different context-aware VNs. We then compare the obtained results against the AODV routing protocol [36], which is the base for the routing approach in the IEEE 802.11s [30] (the IEEE 802.11 amendment for wireless mesh networking).

5.4.2.1 Evaluation Scenario: Details & Variables

In this Section, we detail the used simulation setup, which is summarized in Table 5.4.

We create a grid-based 802.11 WMN topology with $|N|$=8×8 nodes. Each node has a transmission radius of $100m$ ($800m\times800m$ of simulated area), one physical interface of $54Mb/s$ and buffer size of 500 packets.

According to the discussion of Section 3.4.3, in which we presented the details of how to emulate network virtualization in NS-2, we consider three types of VNs, each one using its own "virtual" interface on each WMN node. Then, the capacity and buffer size of each physical WMN node are shared among the three "virtual" interfaces according to the VN resource requirements. Moreover, the interface assigned to a VN uses its dedicated wireless channel. Finally, one "virtual" interface of $1Mb/s$ is assigned to all WMN nodes for control purposes.

Table 5.4: Simulation Details: QoE-Aware Reinforcement Learning Mechanism.

Parameter	**Value**
$\|N\|$	64 (8×8 Grid WMN)
WMN Node Interface Range	$100m$ ($800m \times 800m$ of Simulated Area)
WMN Node Interface Capacity	$54Mb/s$ ($1Mb/s$ is for Control Purposes)
WMN Node Buffer Size	500
VN Services	{Audio, Video and Data Transfer}
Number of Flows per VN	$\{1, 2, 3, 4, 5, 6\}$
Flow Source and Destination	Nodes randomly chosen with Physical Distance less than 7 hops
Flow Lifetime (Δt)	$20s$
User Mobility Type	Each user moves in the WMN at a vehicular velocity of $15m/s$
Learning Mechanism Variables	β_1=0.5, Z=3, ρ=1.1, T_n/T_{e0} (Values to be evaluated)
Simulation Runs	50

In this study, the context-aware VNs are differentiated according to the type of services that they provide support. Therefore, each VN is specialized to run audio, video or data transfer services, and the formulas used to determine the perceived QoE by an user connected to such VNs are respectively defined by the expressions (5.2), (5.3) and (5.4).

The audio, video or data transfer models adopted are based on the ones used in [239]. This way, audio is configured to be transmitted at $64Kb/s$ with a payload size of $160Bytes$. Video uses the H.264 codec sampled at 30 frames per Group of Picture (GoP), 10 of which are P-frames (the used video traces may be found in [303]). Data transfer is modeled to return a maximum MOS for $450Kb/s$ and a minimum MOS for $10Kb/s$.

In each simulation, 1 to 6 users simultaneously access each VN, and each user generates one VN flow that lasts $\Delta t = 20s$ (which is also the simulation duration time). The source node of the user flow is randomly chosen from all the VN nodes, and the destination node of the user flow is randomly chosen from the VN nodes with shortest path lower than 7 hops to the user flow source. To model WMN dynamics, users randomly move among 1-hop neighboring WMN nodes at a vehicular velocity of $15m/s$.

In the QoE-aware learning routing, we fix β_1=0.5, Z=3, ρ=1.1. Finally, each obtained value corresponds to the mean of 50 simulation runs with a confidence degree of 95%. In this study, we increase the number of simulations runs, in order to have more confidence in the results, since we observe small discrepancies in the evaluation of the user perceived QoE.

5.4.2.2 Impact of Learning Period

In this Section, we determine the best value for the learning period, T_n, in which each VN node acquires the QoE-related information of the VN links with its possible 1-hop neighbors towards a specific VN destination (that are selected according to expression (5.29)). This information is then used by each VN node to update the learning routing strategy according to expressions (5.24) and (5.25). With this purpose in mind, we perform a set of experiments for T_n=$\{30, 100, 200, 500\}ms$. In this first set of experiments, we consider T_{e0}=0.5 to control the probability of VN nodes to explore other feasible paths, rather than the expected best one, to reach a specific VN destination (please refer to expression (5.30)).

The results of this evaluation are presented in Fig. 5.9, in which we show the mean perceived QoE of VN users and the VN routing control overhead, when varying the number of users (or flows) per VN. Here, we can observe that a lower T_n gives a better result for the

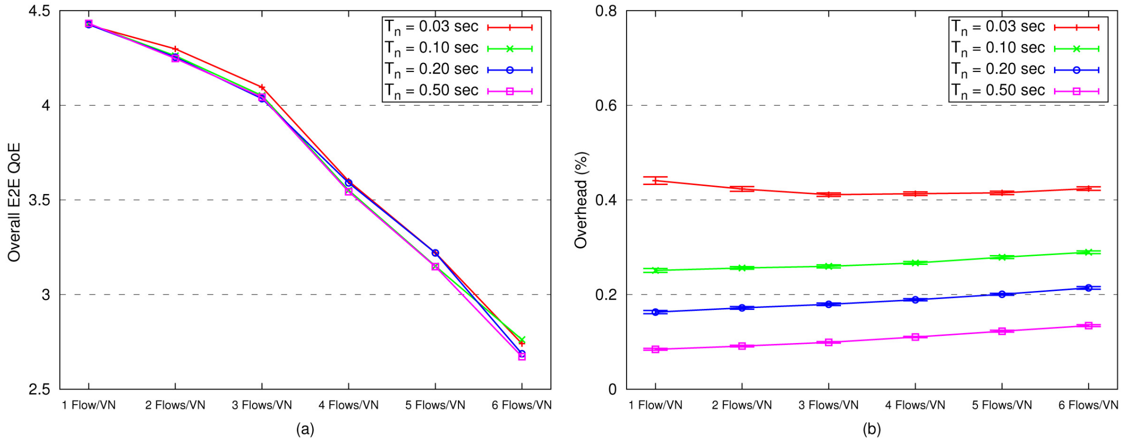


Figure 5.9: Impact of Learning Period on the QoE (a) and Routing Control Overhead (b), when varying the Number of Flows per VN.

overall user perceived QoE. In this case, each VN node, within a VN path traversed by a VN flow towards a specific VN destination, can update the learning scheme faster. Consequently, when a VN link gets congested, it can select a next hop that provides a better expected QoE towards the destination. However, the QoE benefits of using T_n=30ms are very small when compared to the other T_n values, and we can observe that the control overhead is the highest for T_n=30ms. As a compromise between the QoE gains and overhead drawbacks, we will assume T_n=100ms in the next experiments.

5.4.2.3 Impact of Exploration Probability

The following set of experiments is meant to determine the best values for the probability of each VN node to explore all the possible next hops (selected according to expression (5.29)) to reach a specific VN destination. This evaluation is performed by controlling the T_{e0} parameter that directly influences such exploration probability (please refer to expression (5.30)). This way, we run several simulations, in which each VN node, within a VN path traversed by a VN flow towards a specific VN destination, can assign a small (T_{e0}=0.2), medium (T_{e0}=0.4) or high (T_{e0}=0.6) level of exploration to use any of its possible next hops to reach such destination.

In Fig. 5.10, we show the mean perceived QoE of VN users and the VN routing control overhead, when there are 4 users (or flows) per VN. These results are presented for all the possible combinations of assigning a distinct level of exploration probability to all the flows of each VN type (VNs supporting audio, video or data transfer flows).

We observe that the best combination of T_{e0} values regarding the trade-off between QoE and overhead is: T_{e0}=0.2 for VNs supporting audio flows, T_{e0}=0.4 for VNs supporting video flows, and T_{e0}=0.2 for VNs supporting data flows. First, we need to assign low values of path exploration (or high values of exploitation of the best expected paths) to audio flows, since these flows need to follow short delay routes to not suffer high QoE decays. The same happens with the TCP-based data transfer flows, which do not tolerate packet reordering, and so, they benefit from low path exploration. Finally, it is useful to assign high exploration probability to video flows by taking advantage of the higher VN path diversity, since they

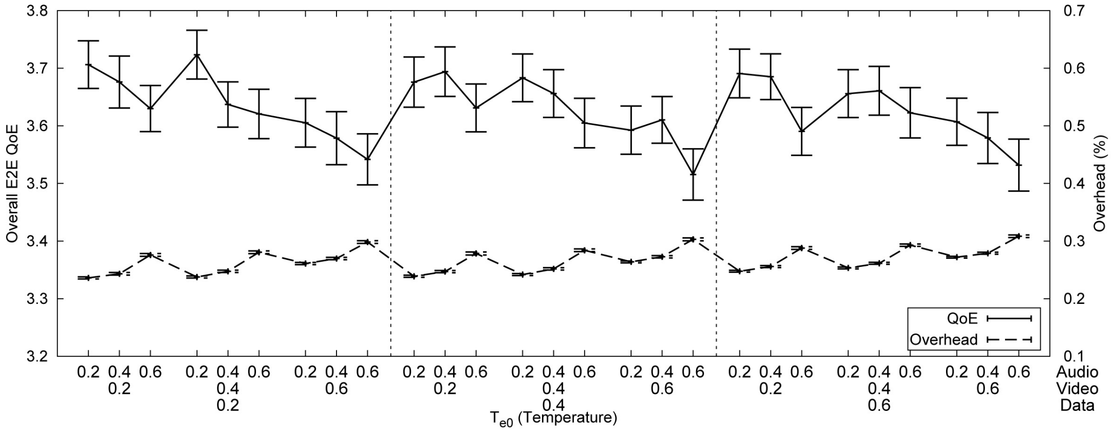


Figure 5.10: Impact of Exploration Level on the QoE and Routing Control Overhead, when there are 4 Flows/VN.

can tolerate more packet reordering. In the remaining experiments, we will use the above combination of T_{e0} values to all flows of each VN type.

5.4.2.4 QoE-Aware Routing vs AODV

We now compare the performance of the QoE-aware learning routing mechanism against the AODV routing protocol, when applied to different context-aware VNs.

In Fig. 5.11, we present the mean perceived QoE by an user connected to a VN supporting audio, video or data transfer services, obtained with both routing schemes and with a variable number of flows per VN. From these results, we can see that the VNs supporting video flows present the highest QoE values when applying the learning routing strategy. This is because, once a VN path becomes congested, the reinforcement learning mechanism leads to more frequent exploration of alternative next hops, which enable the balancing of these flows in the VN along different paths. Therefore, the overall congestion level is decreased, which reduces the packet loss rate for video packets at the VN nodes.

Still from the results of Fig. 5.11, we can observe that the QoE gains of the learning routing strategy in comparison to the AODV routing protocol are visible for all VN types, and they are more visible when the VNs tend to be saturated. For instance, in the VNs supporting video and audio services, Fig. 5.11 shows improvements of more than 1 QoE point when there are 6 flows per VN type.

In Fig. 5.12, it is depicted the relation between the E2E QoS metrics that significantly affect the QoE of audio, video and data transfer flows, which respectively are the delay, loss rate and throughput (please refer to expressions (5.2), (5.3) and (5.4)). For both routing approaches, these results are relative to the first 20 simulations with 4 users (or flows) per VN type, and for each value of user perceived QoE, we plot the associated QoS metric that significantly affects such value. Therefore, we can see that $\approx$ 20 audio flows present delays above $500ms$ with the AODV routing protocol, while in our approach this has been reduced to half. Similarly, the number of video flows that experience packet losses above 5% has been reduced from 12 to 4 when using our scheme. Moreover, the QoS metrics that do not directly influence the QoE measures are not much affected by our approach; for instance, video traffic

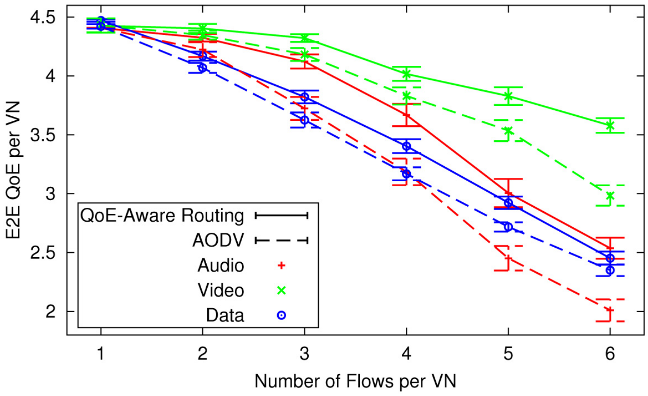


Figure 5.11: Impact of Network Load on the E2E QoE per VN, when varying the Number of Flows per VN.

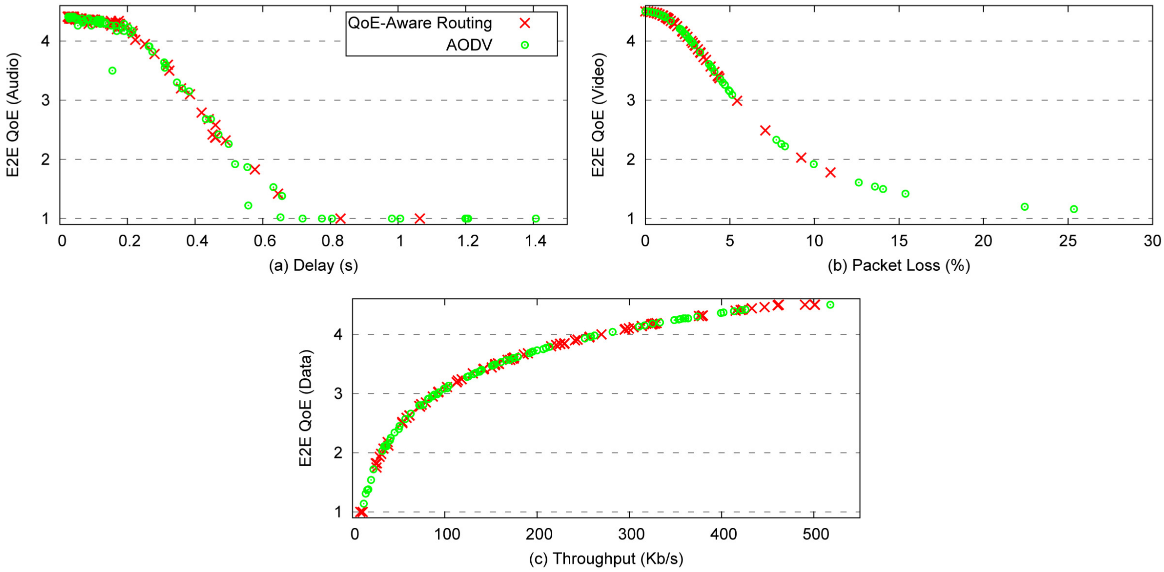


Figure 5.12: Dependency between QoE and Audio Delay (a), Video Loss (b) and Data Throughput, when there are 4 Flows/VN.

even experiences smaller delays when using the QoE-aware learning routing rather than the AODV routing protocol.

Finally, Fig. 5.13 presents the control overhead of the QoE-aware learning and AODV routing schemes, when the number of flows increases in each VN. This figure clearly demonstrates that the AODV routing protocol introduces a higher amount of control traffic when compared to the proposed QoE-aware learning routing mechanism, and the difference between the two schemes almost reaches 0.6% when there are 6 flows per VN. Therefore, our approach significantly increases the user perceived QoE over different types of context-aware VNs, while still reducing the overhead compared to traditional routing approaches.

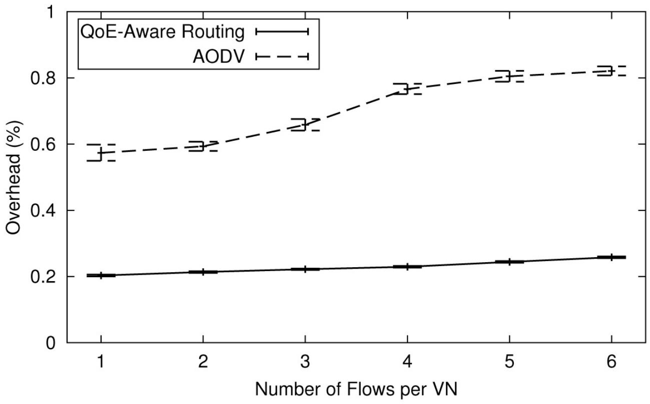


Figure 5.13: Impact of Network Load on the Routing Control Overhead, when varying the Number of Flows per VN.

5.5 Conclusions

Due to the strong emphasis on context and context-awareness in the presented Thesis, this Chapter was focused on the understanding and modeling of several user and VN context information, and on the subsequent handling of such information to define intelligent context-aware mechanisms to be applied to the VN control and management. Such mechanisms can span from the: *(i)* proper configuration of the resources and protocols to run in a specific VN, in order to meet a specific set of user context requirements; *(ii)* definition of the QoE perceived by an user that is connected to a specific context-aware VN; *(iii)* selection of the best VN to be assigned to an user, from the candidate set of fitting VNs, which needs to consider the distinct levels of user context flexibility to reduce the number of VN re-associations during the user connectivity time; *(iv)* adaptation of the VN routes on-demand based on the QoE feedbacks of ongoing VN users.

Given the above discussion, this Chapter started by considering a demonstrative set of user context requirements in order to define possible rules to automatically model such requirements in specific context levels, and then to map such levels in proper VN context features. These VN features can span from distinct resource levels assigned to VN nodes and links, mobility protocols or security mechanisms, or different priority levels for channel assignment. We also discussed the impact of the VN context features on the performance of VN flows, and consequently, on the QoE perceived by VN users.

In the following, we moved into the definition of the metric to select the best VN to assign to a particular user. This metric was designed based on a combined view of the context information characterizing users and VNs, since it considered: *(i)* the context requirements of users, and their context flexibility to be assigned to VNs that do not exactly fit all their requirements, but may incorporate these users if their context flexibility allows to; *(ii)* the context features of the candidate VNs to assign to users, which can also include the overall quality status of these VNs expressed by the mean QoE perceived by the ongoing VN users.

To evaluate this VN selection metric, this Chapter proposed an analytical model that was validated against a simulation study. This model enabled us to assess the impact of considering distinct levels of user context flexibility on the increase of the usability of the

VNs already available in the WMN, while avoiding the constant creation of new VNs to serve single users. From the obtained results, we showed that the probability to create a new VN decreases with the increase of the mean number of users accessing the WMN, especially when these users admit high levels of flexibility on their requirements. In the simulation platform, we incorporated the new context-aware VN selection metric in the distributed VN control and management mechanism proposed in Chapter 4, as well as basic processes for VN creation and remotion. This way, we assessed the accuracy of the analytical results, and we also observed that in the proposed distributed approach for VN control, the VN signaling control delays, overhead and the number of nodes involved in a VN reconfiguration are lower than in other distributed control approaches.

Despite these encouraging results, it is still required to propose and evaluate a metric that considers resource-aware indicators to select non-congested WMN paths to dynamically perform VN extensions or creations, which can increase the overall WMN capacity. The potential of joining context- and resource-aware metrics in the VN and WMN path selection procedures requires the evolvement of the distributed VN control and management mechanism proposed in the previous Chapter. This way, it is important to define a framework that enables the distribution of several user, VN and WMN control knowledge along the different architectural entities, which can then cooperate to autonomously discover, select, create, adapt and remove VNs on-demand in order to cope with user, VN and WMN context changes. Moreover, the distributed context-aware ring proposed in the previous Chapter needs to be enhanced by considering the distinct levels of user context flexibility to design a more intelligent ring structure. This structure should enable the fast cooperation among controllers of VNs that are more probable to deal with user context requirements and their variations.

Finally, this Chapter defined a QoE-aware learning scheme to dynamically select and update the routes between two distinct virtual nodes from a VN. Its key component is a double reinforcement learning strategy that allows the VN nodes to autonomously learn the paths that maximize the QoE perceived by the VN users; in such strategy, the QoE metric can be derived by the particular context features of such VN. Our technique adapts to VN dynamics by switching between an exploitation phase, when VN path quality can maintain a sufficiently high QoE level, and an exploration phase, when VN path quality degrades due to increased congestion. We performed comprehensive simulations to demonstrate the substantial QoE performance and control overhead improvement of the QoE-aware reinforcement learning routing strategy over conventional AODV-based routing techniques, when applied to different context-aware VNs.

In the following Chapter, we will propose and evaluate an overall distributed and context-aware framework to discover, select, create, adapt and remove VNs on-demand. Such framework will embed, and sometimes enhance, the distributed and context-aware control rules, mechanisms and metrics proposed so far in this Thesis.

Chapter 6

Distributed Context-Aware VN Control & Management

This Chapter defines an overall distributed and context-aware framework to autonomously discover, select, create, adapt and remove VNs on-demand. Such framework is based on the distribution of control knowledge and functionalities along the distinct architectural entities, which can then cooperate to control and manage the VNs in a distributed and autonomous way, and in response to any type of context change.

Later in the Chapter, we propose and validate an analytical model to evaluate several performance aspects of using the distributed framework for VN control and management, when compared to centralized or decentralized solutions.

Finally, through simulation, this Chapter presents an overall evaluation of the data and control planes of the multi-VN architecture, and the obtained results are compared against other possibilities to implement the architecture presented in this Thesis.

6.1 Introduction

The last two Chapters explored the idea of employing a distributed mechanism to perform several VN control and management processes, complemented by the context-aware design of several of these processes to favor the flexible support and adaptation of users and VNs.

The envisioned distributed VN control and management requires the autonomous cooperation among the WMN nodes, and their supported virtual nodes, as a mean to: *(i)* quickly discover fitting VNs to assign to users that are constantly arriving, accessing and leaving the WMN, or changing their context requirements; *(ii)* perform the extension or creation of the discovered VNs up to the locations of users, and the remotion of the unused VNs.

The context-aware architectural design, which was mainly targeted in Chapter 5, focuses on the definition of rules to quantify and map user context requirements in suitable VN characteristics and protocols, and on the importance of taking into account the context of users, VNs and WMN nodes, as a mean towards selecting VNs to connect users and paths to build such VNs. This context-aware design also aims to reduce the number of VN re-associations during the user connectivity time, which is done by selecting the VNs that are most probable to deal with the specific levels of user context flexibility. Moreover, this flexibility decreases the need to create new VNs to serve single users and its associated complexity.

In the two previous Chapters, we already showed several benefits of the distributed control and context-awareness, such as: *(i)* the small delays and overhead introduced by the VN update and local VN extension processes, even in the presence of mobility of mesh clients; *(ii)* the reduction on the number of VN creations triggered in the WMN and the number of nodes involved in a VN extension, thus decreasing the number of network reconfigurations required. Despite these encouraging results, it is important to enhance some of the concepts already proposed in this Thesis, such as the mechanisms that are intrinsic to the dynamic building and adaptation of the distributed context-aware ring to enable the cooperation among the distinct VN controllers, or the used criteria to decide when to create a new VN. Moreover, it is important to consider the WMN resource availability to select non-congested WMN paths to perform VN extensions or creations, which can increase the WMN capacity. Finally, we are still missing the comparison between considering a distributed solution for architectural control and a centralized or decentralized one.

In this context, Section 6.2 proposes an overall framework to distribute the control knowledge (e.g., VN context and topology information, WMN resource availability) and functionalities along the distinct architectural entities, and to enable their cooperation to autonomously control and manage the VNs on-demand. In this overall framework for VN control and management, we incorporate the context-aware VN selection metric defined in Chapter 5, and we define a new resource-aware metric to select the WMN path to perform a VN extension or creation. This Section also considers the different levels of user context flexibility in the design and organization of the ring structure, enabling fast communications among controllers of VNs that are most probable to deal with user context requirements and their variations. Moreover, it defines a metric to dynamically adapt the location of each VN controller in order to improve the performance of the control and management functionalities inside the VN, and to avoid topological mismatching problems between the physical WMN infrastructure and the ring structure. Finally, it provides the mechanisms to dynamically insert, update and remove the connections that are established among VN controllers.

In the following, Section 6.3 proposes a control mechanism (with the detailed signaling control processes and messages) to autonomously discover, create, adapt and remove VNs on-demand. In such mechanism, we consider the previously defined VN and WMN path selection metrics, and we introduce a new one to optimize the global selection of fitting VNs to assign to users, which is performed with the help of the distributed context-aware ring.

In order to assess the impact of the distributed and context-aware framework for VN control and management, Section 6.4 defines an analytical model to evaluate several performance aspects of such framework when applied to specific WMN scenarios, and compares it against centralized and decentralized control schemes. For each control approach (centralized, decentralized or distributed), the model is able to measure: *(i)* delay of the user association to a fitting VN available in the WMN; *(ii)* overhead of each control approach.

In the following, Section 6.5 provides an extensive evaluation of the data and control planes of the multi-VN architecture. The obtained results are also compared against other possibilities to control and manage the architecture in a centralized or distributed way.

Finally, Section 6.6 summarizes the main important conclusions of the evaluations performed in this Chapter.

6.2 Distribution of Control Knowledge & Functionalities

In the envisioned WMN scenario, users are constantly changing their context requirements and locations, and VNs need to be frequently adapted to deal with user and WMN context change. This dynamics raises the need to define intelligent control and management mechanisms for the fast and scalable VN discovery, adaptation, creation or remotion. In order to leave behind the scalability limitations of centralized control approaches, we aim to perform these mechanisms in a distributed and autonomous way.

In such approach, the distinct architectural entities are endowed with specific control functionalities that allow them to cooperate in the distributed VN control and management, without the need to constantly notify and update a central entity. First, each WMN node needs to be aware of the VN and WMN context information available in the 1-hop neighborhood, which can enable the fast local discovery and selection of: *(i)* the best fitting VN to assign to an user; *(ii)* the WMN path characterized by high resource availability to smoothly reconfigure such VN. Second, the virtual nodes from a specific VN are expected to cooperate in the selection of the best location for the VN controller, and in the selection of high quality routes between two distinct VN nodes. Third, the existence of multiple VNs, each one meeting particular context features and having its own control administration, requires the dynamic cooperation among VNs characterized by similar context features, in order to accelerate the global discovery and adaptation of fitting VNs to assign to users.

Several solutions proposed in the last two Chapters already focused on the distribution of control responsibilities along the different WMN nodes, VN nodes and VNs, which can then cooperate to provide a flexible and autonomous VN control and management. However, these solutions still lack several aspects. For instance: *(i)* WMN nodes have to be endowed with the possibility to gather information available in the 2-hop or 3-hop neighborhood, and not only in the 1-hop neighborhood as addressed in the work of Chapter 4, which can increase the probability of local VN extensions to be triggered in the WMN; *(ii)* we need to consider resource-aware metrics to select non-congested WMN paths to perform VN adaptations; *(iii)* it is required to detail the processes to update the stored information and the location of each VN controller; *(iv)* it is important to define the mechanisms for the dynamic building and adaptation of the distributed context-aware ring proposed in Section 4.2.1.

Algorithm 2: Distribution of Control Knowledge & Functionalities.

```
1  foreach ($n \in N$) do
2      foreach ($u \in U$) do
3          • Node $n$ stores the context requirements of the user $u$;
4          • Node $n$ can select any VN $v \in V_u$ available on it according to expression (5.10);
5          • Node $n$ can select any VN $v \in V_u$ available on its physical neighborhood according to
           expression (6.2);
6          • Node $n$ can trigger the global discovery of any VN $v \in V_u$ in the WMN;
7      • Receives the information of VNs and resources available on it or on its neighborhood;
8      • Forwards the gathered information to its 1-hop physical neighbors using the tuples
       defined by expression (6.3);
9  foreach ($n_v \in N_v$) do
10     • Sends the perceived QoE of its users to $O_v$, and receives the $QoE_v$ value (please refer to
       expression (5.11)) from $O_v$;
11     • Sends the $QoE_v$ value to the WMN node that supports it;
12     • Informs $O_v$ about its intentions to leave the VN $v$;
13 foreach ($O_v$) do
14     • Sends the $QoE_v$ value to all the nodes of the VN $v$, and receives the perceived QoE by all
       the users of the VN $v$;
15     • Updates the location of $O_v$ according to expression (6.4);
16     • Informs $O_{v_s}$ and $O_{v_p}$ of its actual location and of its context features;
17     • Informs $O_{v_s}$ and $O_{v_s}$ in case of the complete remotion of the VN $v$;
```

To address these open issues, this Section proposes an overall framework to enable the coordination and cooperation among the different architectural entities that were introduced in Section 3.2.3, to control and manage the VNs in a distributed way (published in [20] and [21]). We will make use of the variables already defined in Table 5.2 to describe the most important control functionalities of each architectural entity. Moreover, such description will be properly complemented by Algorithm 2.

6.2.1 WMN Node

When the user $u \in U$ arrives at the WMN node $n \in N$ and wants to be connected to a VN that belongs to the set of fitting VNs for the user u (V_u), the node n first stores the context requirements sent by the user u (Alg.2-line3).

In the following, the node n inspects the information of the VNs that it supports. If it supports any VN $v \in V_u$, it selects the VN to be updated and assigned to the user u, according to the context-aware VN selection metric defined in expression (5.10) (Alg.2-line4).

Nevertheless, if the node n does not support any fitting VN to assign to the user u, it can trigger the local extension of any VN $v \in V_u$ available in its physical neighborhood, in order to allow the fast association of the user u. Based on the VN and WMN knowledge that the node n stores about its physical neighborhood, it can select the best VN from the candidate set V_u of fitting VNs to assign to the user u, and the WMN path to trigger the local extension of the selected VN up to the node n (Alg.2-line5). We now detail this VN and WMN path selection process.

First, the node n limits the candidate set of VNs to the ones accessible in a minimum number of hops, enabling the creation of a lower number of virtual nodes in a future VN extension. Second, for each possible VN $v \in V_u$ available in its physical neighborhood, the node n has to consider a weighted combination between: *(i)* the score given by the context-aware VN selection metric defined in expression (5.10), in order to more frequently select VNs that are configured to exactly meet the context requirements of the user u (R_u), while decreasing the number of user re-associations to other fitting VNs due to variations of the user context demands (T_u); *(ii)* the resource availability (e.g., bandwidth) in the path between the WMN node supporting a VN $v \in V_u$ (n_v) and the WMN node n, $R_{L_{n_v \to n}}$, in order to select non-congested WMN paths to perform VN extensions, increasing the WMN capacity.

In order to directly compare the resources available in heterogeneous WMN paths, we normalize the level of resources available in a path between the WMN node n_v supporting a VN $v \in V_u$ and the WMN node n, $R_{L_{n_v \to n}}$, based on the total resource capacity of such path, $max\{R_{L_{n_v \to n}}\}$. This normalized level of resources, RA_v, is then defined by:

$$RA_v = \frac{R_{L_{n_v \to n_u}}}{max\{R_{L_{n_v \to n_u}}\}} \ \ , \ \ 0 \le RA_v \le 1. \tag{6.1}$$

According to the above discussion, the node n can inspect the VN and resource information that it gathered from its physical neighborhood, in order to select the VN $v_u \in V_u$ to be locally extended up to the location of the user u. This VN is selected according to:

$$v_u \in V_u \ \ s.t. \ \ \min_{v \in V_u}\{|L_{n_v \to n}|\} \ \ \&\& \ \ \min_{v \in V_u}\{w_1 \times CD_{(u,v)} + w_2 \times (1 - RA_v)\} \ \ , \ \ w_1 + w_2 = 1, \tag{6.2}$$

where: *(i)* $|L_{n_v \to n}|$ is the number of hops in a possible WMN path to perform the extension of the VN $v \in V_u$ up to the node n; *(ii)* w_1 is the weight assigned to the **C**ontext-aware **D**ifference (CD) among the context needs of the user u (R_u, T_u) and the context features of the VN $v \in V_u$ (R_v); *(iii)* w_2 is the weight assigned to the level of resources available in a possible WMN path to perform the extension of the VN $v \in V_u$ up to the node n.

If these two local VN control processes cannot be performed due to unavailability of fitting VNs for the user u in the physical neighborhood, the node n can trigger the global discovery of any VN $v \in V_u$ in the WMN (Alg.2-line6), which will be detailed in Section 6.2.3.3.

In order to perform the VN control processes presented above, the node n periodically receives the information of each VN $v \in V$ that it supports (T_1 is the periodicity of this process), and then forwards this information to its 1-hop physical neighbors (Alg.2-line7/8). In the following, such information is disseminated by the neighbors of the node n along a maximum number of hops TTL_m. The information tuple associated to each virtual node from the VN $v \in V$ that is announced in the WMN, is defined by:

$$t_v = \{n_{ID}, v_{ID}, TTL, QoE_v, RA_v\}, \tag{6.3}$$

where: *(i)* n_{ID} is the identifier of the WMN node that supports the VN v; *(ii)* v_{ID} is the identifier of the VN v; *(iii)* TTL is the maximum number of hops to still forward the tuple t_v in the WMN; *(iv)* QoE_v is the mean perceived QoE of ongoing users in the VN v, which represents the overall quality level of the VN v (as defined in expression (5.11)); *(v)* RA_v, defined in expression (6.1), is the level of resources available (e.g., bandwidth) in the path between the WMN node n_{ID} and the WMN node that receives the tuple t_v, and it is updated at each WMN node of the path traversed by the tuple t_v.

6.2.2 VN Node & VN Controller

As referred in Section 3.2.3.3, one of the nodes of each VN $v \in V$ is selected to be the controller of the VN v, which is defined by O_v. O_v coordinates the control functionalities performed in the VN v, and it also configures the same context-aware mechanisms and protocols in all the virtual nodes from the VN v.

At the time of the creation of the VN v (more details about the VN creation process will be provided in Section 6.3.4), it is randomly selected a virtual node from the VN v to be O_v. The location of O_v is then dynamically updated to deal with topology changes of the VN v (this O_v selection process will be detailed in Section 6.2.3.2).

As described in Alg.2-line10/14, O_v periodically sends the current QoE_v value to all the virtual nodes from the VN v, and these nodes reply with the new QoE-related information of the users connected to the VN v (T_2 is the periodicity of this process).

Each virtual node from the VN v can then send this QoE_v value to the WMN node that supports it (Alg.2-line11). In case of considering the user QoE expectations as a context parameter to drive the architecture, this WMN node can use this QoE_v value (or the overall quality level provided by the VN $v \in V$) to perform the context-aware VN selection process according to the metric defined in expression (5.10).

Finally, each end-point of the VN $v \in V$ periodically inspects the connectivity status of the users connected to it. If there are no attached users in a specific timeout, such end-point notifies its 1-hop neighbor node in the VN v about its intentions to leave the VN v, and such notification in then forwarded to O_v (Alg.2-line12). More details about this VN remotion process will be provided in Section 6.3.5.

6.2.3 Enhanced Distributed Context-Aware Ring

In this Section, we enhance the DHT-based context-aware ring to interconnect VN controllers proposed in Section 4.2.1. Our main aim is to decrease the time spent in the global discovery of VNs available in the WMN, as well as the signaling control overhead introduced by a Chord-based solution to maintain the ring connectivity.

This way, we propose a new ring structure to enable the fast and context-driven discovery of a fitting VN to assign to an user, and the fast re-association of the user to other fitting VN in case of user context change. To deal with VN dynamics, we also define the processes to dynamically update the location of each VN controller, and to automatically insert, update and remove the DHT-based routing entries that are established among the VN controllers in the ring structure.

6.2.3.1 Context Space Partition & Ring Organization

We aim to optimize the global discovery of VNs in the WMN. With this purpose in mind, we consider the distinct levels of user context flexibility to define a ring structure that enables fast communications among the controllers of VNs that are most probable to deal with such context flexibility. Please note that, this user context flexibility was already defined in Chapter 5, as the maximum variation ranges that the user demands can admit during the user connectivity time, which were modeled by $T_u=\{t_{u_c} \mid t_{u_c} \in M_c, c \in C\}$. In such approach, the user context requirements that are more probable to change, are the ones admitting high variation ranges (or high t_{u_c}).

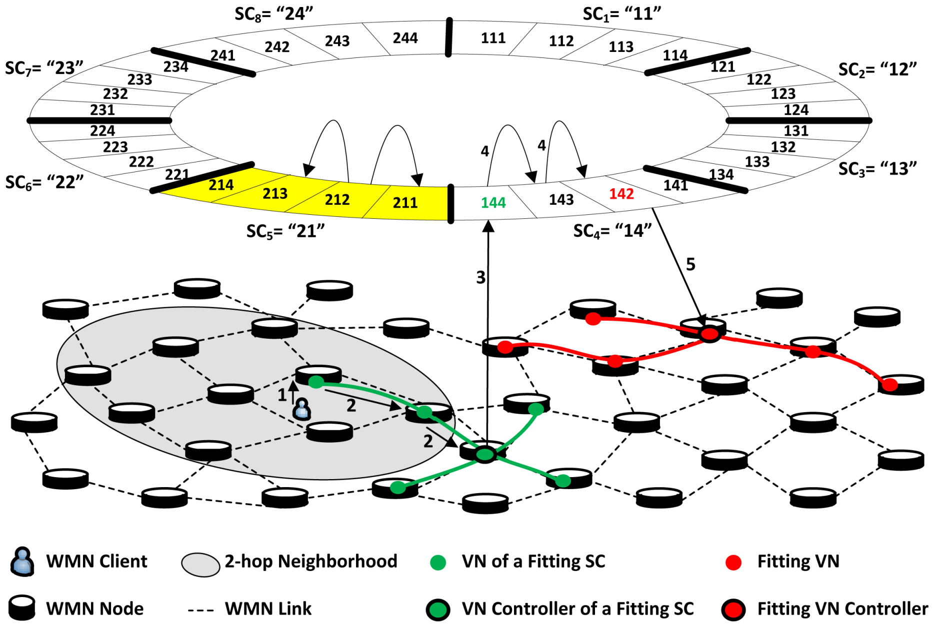


Figure 6.1: Enhanced DHT-Based Context-Aware Ring: Structure & Organization.

Our approach is the following. We split the set of context features C in two disjoint sets, $C = C' \cup \neg C'$, according to the distinct levels of flexibility admitted by the users for such features. Here, the set C' contains the context features that do not frequently admit variations from the user side (low t_{u_c}), and do not usually trigger the need to re-associate an user to another VN; on the other hand, the set $\neg C'$ contains the remaining context features that generally admit large variations (high t_{u_c}), and constantly trigger VN adaptations (please refer to the discussion of Section 5.2.2). This distinction is very important, since we can have a first notion of the user context requirements that have more probability to change. Then, we can use this probability to enhance the global distributed VN discovery in the WMN.

Following this approach, we propose a new DHT-based context-aware ring structure that is logically split in a set of semantic clusters (SCs), where each SC groups the controllers of VNs characterized by a specific combination of the levels that can be admitted by the context features of the set C'. Please note that, as in the ring structure proposed in Section 4.2.1, the key that identifies each VN controller in the ring embeds the set of context features (C) and the levels of these features (M_c, $\forall c \in C$) that characterize such VN.

According to this strategy, we aim to enable the possibility of the controllers of VNs, that are most probable to deal with the most frequent user context changes, to be closely located in the ring structure. This is explained by the fact that in the envisioned ring structure, each SC includes the controllers of VNs characterized by the same levels in the set of context features C', and by distinct levels in the set of context features $\neg C'$. Therefore, by establishing direct DHT-based connections between the controllers of VNs that belong to the same SC (these connections will be detailed below), we ensure the possibility to quickly react to the most frequent user context changes (the ones associated to the set of context features $\neg C'$); another VN that copes with such context changes can be discovered in a small number of communications, which decreases the time spent in the global distributed VN discovery.

Concerning the DHT-based connectivity among the VN controllers that belong to the same SC, the context parameter $c \in \neg C'$ with the highest t_{u_c} has the most important role on the proximity among VN controllers. Such parameter is the one with the highest probability to change from the user side, and so, it will more frequently trigger the need to discover another VN to assign to the user. Therefore, the controllers of VNs that only differ in one level in the context parameter $c \in \neg C'$ with the highest t_{u_c} are 1-hop logical neighbors in the ring, storing a direct DHT-based routing entry to communicate to each other. This enables the possibility to quickly discover a point of attachment of a VN to cope with the more frequent user context changes. In our approach, each O_v stores a DHT-based routing entry to communicate to O_{v_s} (which is the controller of the successor VN of the VN $v \in V$ in the ring structure, $v_s \in V$) and another one to communicate to O_{v_p} (which is the controller of the predecessor VN of the VN $v \in V$ in the ring structure, $v_p \in V$).

Fig. 6.1 presents an example of the ring structure by considering $C = \{c_1, c_2, c_3\}$, $C' = \{c_1, c_2\}$, $\neg C' = \{c_3\}$, $|M_{c_1}| = 2$, $|M_{c_2}| = 4$ and $|M_{c_3}| = 4$. First, the total number of possible SCs available in the ring is given by $|M_{c_1}||M_{c_2}|$=8; e.g., SC_3 groups the controllers of VNs characterized by the level 1 on the context parameter $c_1 \in C'$ and the level 3 on context parameter $c_2 \in C'$. Second, the SCs, and the VN controllers within the same SC, are organized in a consecutive order of the levels of the context parameters that are part of the sets C' and $\neg C'$, respectively. Third, the context parameter $c_3 \in \neg C'$ has associated the highest t_{u_c}; thus, the controllers of two VNs that only differ in one level in this context parameter are 1-hop logical neighbors in the ring (in Fig. 6.1, the controller of the VN with ID 212 stores direct links to contact the controllers of the VNs with IDs 211 and 213).

Still in Fig. 6.1, we provide an example on how to make use of the ring structure to quickly find another VN to assign to an user that changed his/her context requirements. The user was already connected to the VN with ID 144, but suddenly changed the demand of the context parameter $c_3 \in \neg C'$, which is the most probable to change, from 4 to 2; therefore, it is required to find the VN with ID 142 in the WMN. After the reception of the new context requirements of the user (Fig. 6.1→1), the WMN node where the user is connected to, inspects its stored information tuples and detects that it does not have information of any node of the desired VN in its physical neighborhood. Therefore, it triggers a global VN discovery in the WMN, which will be performed with the help of the ring structure. Again using the information of its received tuples, the WMN node first tries to contact a VN belonging to the same SC of the desired VN. Since the VN with ID 144, in which the user was connected to, is already available in this WMN node, the controller of such VN is notified (Fig. 6.1→2). Within the ring structure (Fig. 6.1→3), the discovery is performed by using the DHT-based connections established among consecutive VN controllers (Fig. 6.1→4). As can be seen in Fig. 6.1, the controller of the desired VN is found after performing two ring communications (Fig. 6.1→5); this reflects the benefits of incorporating the levels of user context flexibility in the ring organization, in order to decrease the number of communications (and the delay) required to discover the new VN to be assigned to the user.

6.2.3.2 Maintenance of the Ring Connectivity

This Section details the signaling control processes and messages that allow the controller of each VN $v \in V$, O_v, to dynamically update its location to react to topology changes of the VN v, and to automatically create, update and remove the DHT-based routing entries used to contact its 1-hop logical neighbors in the distributed ring structure.

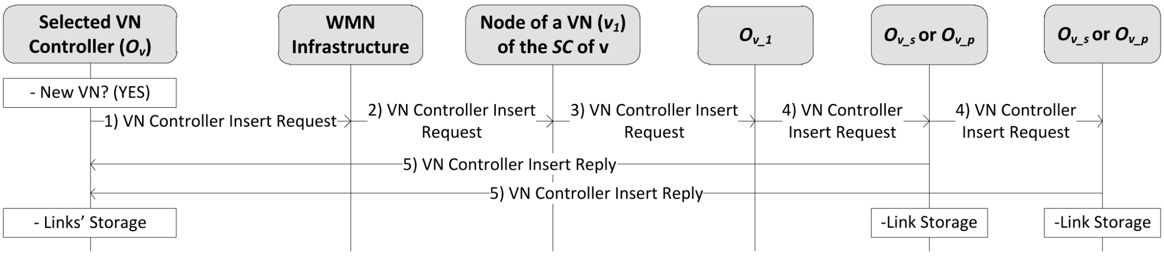


Figure 6.2: Ring Maintenance: Creation of Links among VN Controllers.

- **Links Creation**

At the time of the creation of the VN $v \in V$ (more details about the VN creation process will be provided in Section 6.3.4), the selected virtual node from the VN v to be its controller, O_v, has to be integrated in the ring through the establishment of the routes to contact O_{v_s} and O_{v_p}. This control process is detailed in Fig. 6.2.

First, O_v contacts its substrate WMN node by sending a control message that includes the context features characterizing the VN v (Fig. 6.2→1). This WMN node then inspects its stored information tuples about the VNs available in its physical neighborhood, in order to contact the nearest WMN node that provides support for a VN with controller belonging to the *SC* of O_v (Fig. 6.2→2); such VN (e.g., the VN v_1) is randomly selected from the VNs characterized by the same levels for the set of context features C'. After being contacted by a virtual node from the VN v_1 (Fig. 6.2→3), O_{v_1} contacts its 1-hop ring neighbor that is closely located to O_{v_s} or O_{v_p} (Fig. 6.2→4). Such process is performed in a DHT-alike way until finding O_{v_s} and O_{v_p}, which then reply to O_v (Fig. 6.2→5). Finally, O_v stores the routes to contact O_{v_s} and O_{v_p}.

- **O_v Selection & Links Update**

The location of the controller of the VN $v \in V$, O_v, has to be periodically updated to deal with the dynamics of the multi-VN architecture (Alg.2-line15). In the one hand, O_v has to be closely located to every virtual node from the VN v, in order to reduce the delays of the control communications performed inside the VN v. On the other hand, to avoid topology mismatching problems between the WMN infrastructure and the ring structure, the mean hop distance among consecutive VN controllers needs to be minimized, which can reduce the delays of ring communications. According to these two factors, the O_v location can be dynamically selected according to:

$$O_v = \{n_v \in N_v \mid \min_{n_v \in N_v} \{L_{n_v} + L'_{n_v}\}\}, \tag{6.4}$$

where,

$$L_{n_v} = \frac{1}{|N_v|} \times \sum_{n_{v_j} \in N_v} |L_{n_v \to n_{v_j}}|, \tag{6.5}$$

and,

$$L'_{n_v} = \frac{|L_{n_v \to O_{v_s}}| + |L_{n_v \to O_{v_p}}|}{2}. \tag{6.6}$$

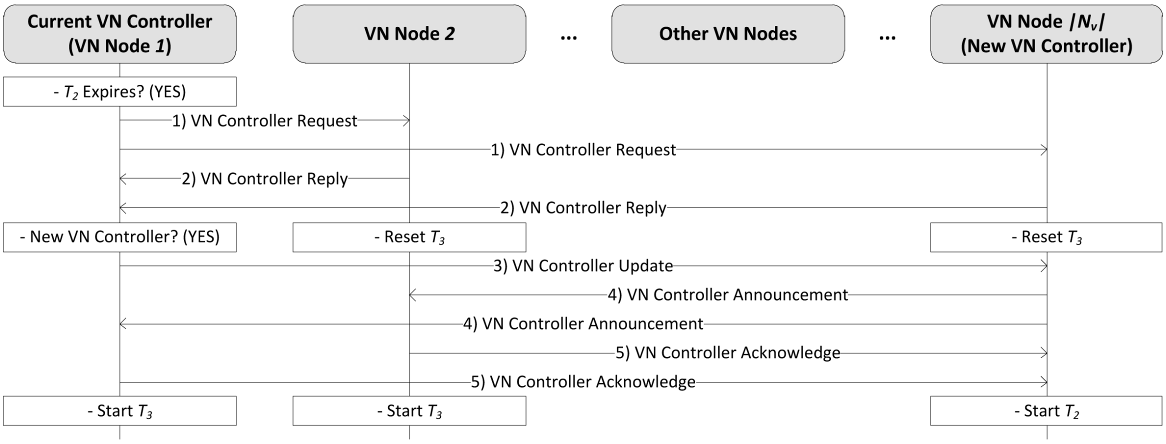


Figure 6.3: VN Controller Update.

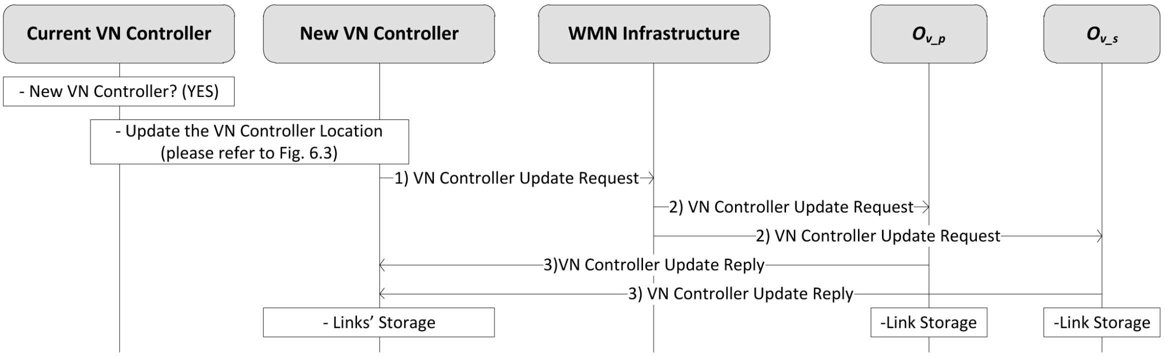


Figure 6.4: Ring Maintenance: Update of Links among VN Controllers.

From expression (6.4), the selected O_v is the virtual node from the VN v that both minimizes the mean hop distance to every other virtual node from the VN v (L_{n_v}) and the mean hop distance to O_{v_s} or O_{v_p} (L'_{n_v}). Other metrics to select O_v, such as the resource availability and stability of each virtual node from the VN v, can be incorporated in the future.

In Fig. 6.3, we describe the process to update, if necessary, the location of O_v in the VN v. As previously referred, O_v periodically sends the current QoE_v value to all the virtual nodes from the VN v (Fig. 6.3→1), and these nodes reply with the new QoE-related information of the users connected to the VN v (Fig. 6.3→2). With the received replies, the current O_v performs the selection process described in expression (6.4), in order to possible select another virtual node from the VN v to be O_v. If it is true, the new O_v is notified by the current O_v (Fig. 6.3→3), and then, the new O_v announces itself in the VN v (Fig. 6.3→4/5).

In this control process, if a specific virtual node from the VN v did not receive any message from O_v in the timeout T_3 due to a misbehavior of O_v, a new O_v has to be selected. In order to easily select another O_v in a synchronized way, O_v periodically provides an ordered list of the virtual nodes from the VN v according to their rankings to be designated as O_v (obtained through the expression (6.4)). Each virtual node from the VN v ($n_v \in N_v$) then configures its timer T_3 based on its position po_{n_v} in such list ($T_{3_{n_v}}$=T_3×po_{n_v}). When a problem occurs

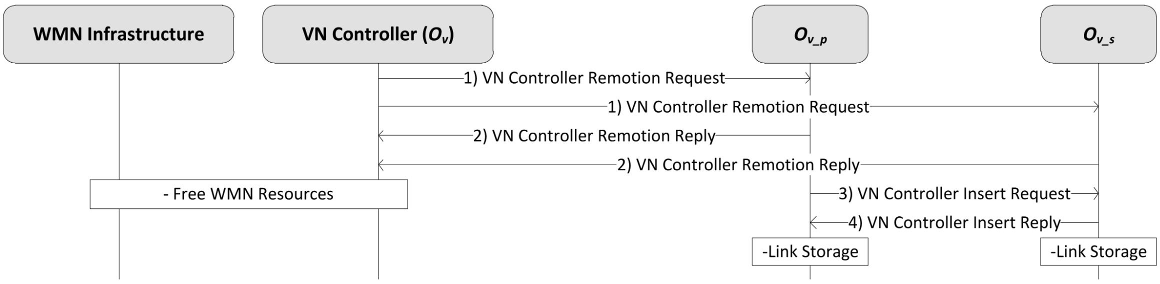


Figure 6.5: Ring Maintenance: Remotion of Links among VN Controllers.

with O_v, the first node of the list provided by O_v is able to auto-designate and announce itself as the new O_v. Through this approach, we can even deal with partitions in VNs, allowing each VN partition to autonomously select its VN controller.

Finally, the location of O_v has to be periodically announced to O_{v_s} and O_{v_p} ir order to be refreshed or updated (Alg.2-line16). In this control process, which is depicted in Fig. 6.4, O_v first contacts its substrate WMN node (Fig. 6.4→1). This WMN node then notifies the WMN nodes that provide support to O_{v_s} and O_{v_p} (Fig. 6.4→2), which then reply to O_v (Fig. 6.4→3); please note that the messages exchanged among VN controllers also convey the VN context information in order to optimize the global selection of fitting VNs to assign to users (detailed in Section 6.3.3.2). Finally, O_v updates the routes to contact O_{v_s} and O_{v_p}.

- **Links Remotion**

If the controller of the VN $v \in V$, O_v, has to leave the WMN due to the complete remotion of the VN v (more details about this VN remotion process will be provided in Section 6.3.5), it notifies such intention to O_{v_s} and O_{v_p} in order to adapt the ring structure accordingly (Fig. 6.5→1/2 and Alg.2-line17). These notification messages convey the information related to the location of O_{v_s} and O_{v_p} to allow these VN controllers to become 1-hop logical neighbors in the ring structure. They can then announce themselves to each other (Fig. 6.5→3/4). This ring adaptation process can also be triggered by O_{v_s} or O_{v_p}, if they did not receive any update message from O_v in the timeout T_4 (Fig. 6.4→1).

6.2.3.3 Real Implementation: Signaling Protocols

In the real implementation of the enhanced DHT-based context-aware ring proposed above, we can resort to the code of any DHT available in the literature (e.g., Bamboo-DHT [285] or Chord-DHT [287]). However, such code needs to be correctly understood in order to be adapted and/or extended to deal with the particularities and requirements of our DHT-based approach. In the following, we present a brief discussion on how to use and/or adapt the Bamboo-DHT to implement the proposed DHT-based context-aware ring.

First, we need to enable the automatic assignment of a context-aware key to each VN controller based on the context parameters, and the levels of these parameters, that characterize such VN. Moreover, it is important to allow the possibility to dynamically update the location of the key that characterizes a specific VN in the control ring, since the location of the VN controller can be updated to react to VN topology changes.

With respect to the update of the location of the controller of a specific VN, it can be triggered after the exchanging of QoE-related information between the VN controller and all the virtual nodes from such VN (described in Fig. 6.3→1/2). This exchange of QoE-related information can be performed by resorting to the Context Transfer Protocol (CXTP) [286].

In the following, we can adapt the basic principles of the control processes associated to the *BAMBOO_PUT* and *BAMBOO_PUT_OK* messages [285], in order to dynamically create, update and remove the DHT-based routing entries established among the VN controllers that belong to the same SC of the context-aware ring structure. Finally, the methods associated to the *BAMBOO_GET* and *BAMBOO_GET_OK* messages [285] can be adapted to implement the context-driven global VN discovery performed within the ring structure.

6.2.4 Global WMN Manager

In our distributed control framework, the global WMN manager knows the ongoing VNs available in the WMN, and stores the context features of each VN and a link to access to one of the nodes of each VN. This minimum control knowledge of the VNs available in the WMN can assist the distributed VN control and management. For instance, in case of a small number or high sparse VNs, a specific WMN node may not store any route to contact SC_1, where a new O_v has to be inserted. In this case, the global WMN manager is notified to redirect the process to SC_1. The global WMN manager can also help to decide if a new VN should be created in the WMN, and can provide the initial support for its controller.

6.3 User Association & VN Control

Using the previous framework to enable the cooperation among the architectural entities to control and manage VNs in a distributed way, this Section defines a signaling control mechanism to autonomously discover, select, adapt, create or remove VNs on-demand, and in response to any type of context change (published in [20] and [21]).

In the description of this control mechanism, we use the variables already defined in Table 5.2, and we consider that the user $u \in U$ is arriving at the WMN node $n_u \in N$ and wants to be connected to a fitting VN $v \in V_u$. Thus, the user u can be automatically connected to any VN $v \in V_u$ already available in the node n_u, triggering a VN update. Otherwise, it can be triggered the local selection and extension of any VN $v \in V_u$ available in the physical neighborhood of the node n_u, or the global discovery, selection and extension of any VN $v \in V_u$ available in the WMN, with the help of the ring structure presented in Section 6.2.3.

In the definition of these three VN control processes, we extend the ones presented in Section 4.2.2 by considering the VN and WMN path selection metrics, respectively defined in Sections 5.3.2 (see expression (5.10)) and 6.2.1 (see expression (6.2)). Moreover, we introduce a scheme to optimize the global discovery and selection of fitting VNs to assign to users.

If none of the VNs available in the WMN fits the requirements of the user u, we extend the criteria defined in Section 5.3.3.2, to decide when to create a new VN. We also provide more detail on the process to dynamically remove unused VN nodes or VNs.

6.3.1 VN Update

According to the top part of Fig. 6.6, when the user $u \in U$ arrives at the WMN node $n_u \in N$, user u sends his/her context requirements to node n_u (Fig. 6.6→1).

In the following, node n_1 stores the context requirements of user u (R_u, T_u), and matches them against the features of the VNs that it supports (R_v). If node n_u supports more than one VN that belongs to the set of fitting VNs for user u (V_u), the VN to be assigned to user u is selected as the one that minimizes the $CD_{(u,v)}$ metric defined in expression (5.10).

Considering that it is selected the VN $v_2 \in V_u$, node n_u contacts the virtual node from VN v_2 that it supports, in order to notify the controller of VN v_2, O_{v_2}. This notification is described in Fig. 6.6→2, and the used control message also includes the context of user u.

After storing the context of user u, O_{v_2} cooperates with the global WMN manager to update the resources that node n_u has to assign to its virtual node from VN v_2. This new resource information is then conveyed in the message sent by O_{v_2} to node n_u (Fig. 6.6→3). Finally, user u is connected to VN v_2 (Fig. 6.6→4).

6.3.2 Local VN Selection & Extension

Focusing now in the middle part of Fig. 6.6, if node n_u does not support any VN that belongs to the set of fitting VNs for user u (V_u), it inspects the VN information tuples that it gathered from its physical neighborhood.

If node n_u has information of more than one VN $v \in V_u$ available in its physical neighborhood, node n_u has to select the best fitting VN and WMN path to trigger a local VN extension up to node n_u. This VN and WMN path selection process is performed according to the metric defined in expression (6.2), in which it is considered: *(i)* the **C**ontext-aware **D**ifference (CD) among the context needs of user u (R_u, T_u) and the context features of each possible VN $v \in V_u$ (R_v); and *(ii)* the number of WMN nodes that need to be involved in a possible VN extension and the level of WMN resources available to perform such extension.

Supposing that it is selected the VN $v_3 \in V_u$ to be extended up to node n_u, node n_u then contacts the WMN node that supports the virtual node from VN v_3 (Fig. 6.6→6), in order to notify the controller of VN v_3, O_{v_3}, about the required extension of VN v_3 up to node n_u (Fig. 6.6→7). These two notification messages also include the context of user u.

After storing the context of user u, O_{v_3} cooperates with the global WMN manager to update the topology of VN v_3, and to derive the resources that need to be assigned to the new virtual link of VN v_3. This topology and resource information is then conveyed in the message sent by O_{v_3} to the WMN node where the extension of VN v_3 will start (Fig. 6.6→8); this WMN node can update the resources assigned to its virtual node from VN v_3. In the following, this WMN node notifies node n_u to create a new virtual node from VN v_3, in order to extend VN v_3 (Fig. 6.6→9). Finally, user u is connected to VN v_3 (Fig. 6.6→10).

6.3.3 Global VN Discovery, Selection & Extension

If node n_u does not support and does not have information of any VN that belongs to the set of fitting VNs for user u (V_u), node n_u triggers a global VN discovery in the WMN (which is depicted in the bottom part of Fig. 6.6). The global VN discovery process of any VN $v \in V_u$ has two major steps: *(i)* the redirection of the process to a proper SC of the context-aware distributed ring structure; *(ii)* the discovery performed within the selected SC. To aid the description of the global VN discovery process, we first define the set SC_u as the set of SCs that are part of the ring structure, and contain controllers of VNs characterized by features (R_v) that fit the requirements of user u (R_u, T_u), when considering the set of context parameters C'. This way, SC_u can be designated as the set of fitting SCs for user u.

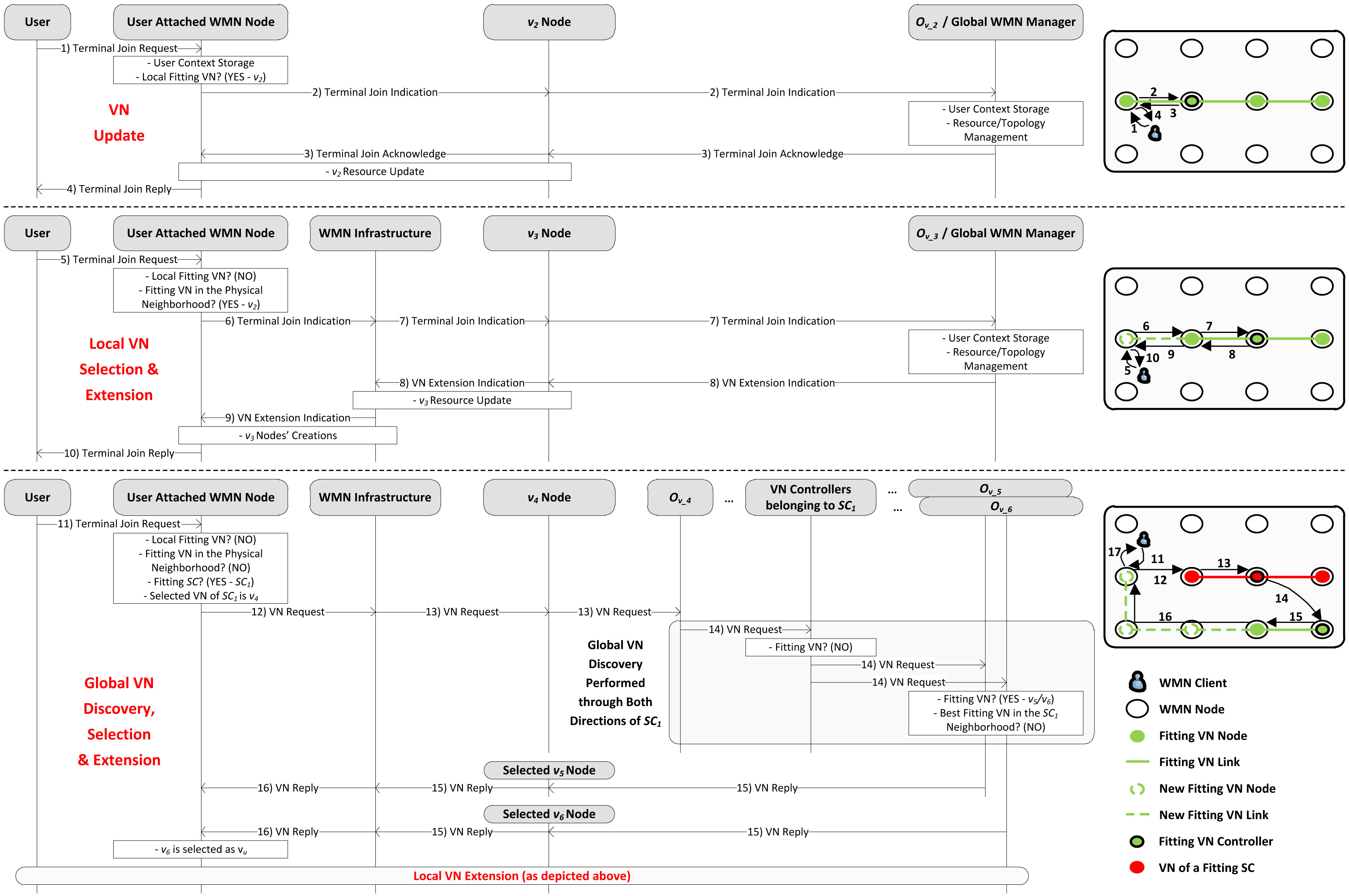


Figure 6.6: User Association & VN Control: VN Update, or Local/Global VN Extension.

6.3.3.1 Redirecting the Global Discovery to a SC of the Ring Structure

In order to start the global VN discovery in the WMN, node n_u first inspects the stored information tuples about the VNs available on it or on its physical neighborhood. This way, node n_u can redirect the global VN discovery to any $SC \in SC_u$, in which can be found a point of attachment of any VN $v \in V_u$.

Based on its VN information tuples, node n_u selects a random tuple to access, in a minimum number of hops, to a VN $v \in V$ belonging to the $SC \in SC_u$ that minimizes the $CD_{(u,v)}$ metric (defined in expression (5.10)), when applied to the set of context parameters C'. This way, the metric to select the VN to initiate the global VN discovery is defined by:

$$v \in V \ \ s.t. \ \min_{v \in V}\{CD_{(u,v)}[|C'|]\} \ \&\& \ \min_{v \in V}\{|L_{n_u \rightarrow n_v}|\}. \tag{6.7}$$

Considering that it is selected the VN $v_4 \in V$ to access to the $SC_1 \in SC_u$, node n_u then contacts the WMN node supporting the virtual node from VN v_4 (Fig. 6.6→12). This virtual node then notifies the controller of VN v_4, O_{v_4}, in order to start a global VN discovery in the SC_1 of the ring structure (Fig. 6.6→13). These two notification messages also include the context of user u.

6.3.3.2 Global Discovery within the Selected SC of the Ring Structure

Within the SC_1 of the ring structure, O_{v_4} uses the routes to contact its 1-hop logical neighbors ($O_{v_{4_s}}$ and $O_{v_{4_p}}$) to forward the global VN discovery through both clockwise and counterclockwise directions of SC_1 (Fig. 6.6→14). Such process is performed along SC_1 in a DHT-alike way, using the connections established among consecutive VN controllers, until finding the controller of any VN $v \in V_u$. Please note that at each VN controller, the VN context features are compared against the context of user u (using the $CD_{(u,v)}$ metric defined in expression (5.10)); therefore, it is important to convey the context of user u in the global VN discovery request.

Focusing on a specific SC_1 direction, after reaching the controller of any VN $v \in V_u$, it may be followed by the controller of another VN $v' \in V_u$, due to the context-aware organization of the ring structure. In this case, the controller of the VN $v \in V_u$ will only redirect the global VN discovery to the controller of the VN $v' \in V_u$, if the VN $v' \in V_u$ achieves a best score than the VN $v \in V_u$, when applying the $CD_{(u,v)}$ metric defined in expression (5.10); this process is performed by taking advantage of the context-aware information exchanged among VN controllers (please remember the discussion of Section 6.2.3.2). This way, we can optimize the global selection of fitting VNs at the cost of one more ring communication.

If the controller of any VN $v \in V_u$ is not available in the SC_1 of the ring structure, the WMN nodes supporting the VN controllers of SC_1, in which the global VN discovery stops, are notified to redirect the discovery to another $SC \in SC_u$. These WMN nodes have then to make use of a random tuple to access to a VN that belongs to the following $SC \in SC_u$, selected according to the metric defined in expression (6.7) (this process is repeated until finding the controller of any VN $v \in V_u$ in the ring structure).

6.3.3.3 Global VN Selection & Extension

After reaching the controller of any VN $v \in V_u$, and considering that the VNs $v_5 \in V_u$ and $v_6 \in V_u$ are the VNs selected in the clockwise and counterclockwise directions of SC_1, O_{v_5}

and O_{v_6} respectively notify the virtual nodes from the VNs v_5 and v_6 that are closely located to node n_u (Fig. 6.6→15).

Then, the WMN nodes that provide support for these virtual nodes notify node n_u about the VNs discovered in the WMN. These notifications are described in Fig. 6.6→16, and the messages received by node n_u need to include: *(i)* the context features of the discovered VNs; and *(ii)* the levels of resources available (e.g., bandwidth) in the traversed paths between the node n_u and the WMN nodes supporting the virtual nodes from the VNs v_5 and v_6 (which was already defined by RA_v in expression (6.1)).

From the obtained replies, node n_u can select the VN $v \in V_u$ to be assigned to user u and the WMN path to perform the VN extension up to node n_u, according to the metric defined in expression (6.2). Finally, a VN extension process similar to the one explained in Section 6.3.2 is started in the WMN.

6.3.4 VN Creation

If node n_u (or another node $n \in N$ in the WMN) cannot redirect the global VN discovery to any $SC \in SC_u$, due to the unavailability of VN information tuples to access to a VN belonging to any $SC \in SC_u$, node n_u notifies the global WMN manager. Based on the knowledge stored by the global WMN manager about the VNs available in the WMN (please remember the discussion of Section 6.2.4), it can: *(i)* select the VN $v \in V_u$ to be extended up to node n_u, as the one that minimizes the $CD_{(u,v)}$ metric defined in expression (5.10); *(ii)* redirect the control mechanism to a virtual node from the selected VN. In this case, a global VN extension process similar to the one explained in Section 6.3.3.3 is started in the WMN.

Nevertheless, if the global WMN manager detects that the WMN does not provide support for any VN $v \in V_u$, the possibility to create the VN that exactly fits the context of user u (R_u) will be evaluated; this VN was already defined by the VN $v'_u \in V_u$ (please see Table 5.2). This VN creation control process is depicted in Fig. 6.7.

In this control process, user u sends his/her context requirements to node n_u (Fig. 6.7→1). Using the $CD_{(u,v)}$ metric defined in expression (5.10), node n_u can then detect that it does not store information tuples to enable the selection of any VN $v \in V_u$ to be assigned to user u, or to start a global VN discovery in the WMN. Therefore, node n_u contacts the global VN manager by sending the context of user u (Fig. 6.7→2).

In order to evaluate the possibility to create the VN v'_u, the global WMN manager starts a process performed through a set L of WMN paths between the WMN gateway (the node $n_O \in N$) and node n_u (Fig. 6.7→3). Following a shortest path strategy, the number of hops of each candidate WMN path between node n_O and node n_u must satisfy the following condition:

$$l \in L : |l| \leq (1 + \beta_2) \times |L_{n_O \to n_u}| \;,\; 0 \leq \beta_2 \leq 1, \tag{6.8}$$

where: *(i)* $|L_{n_O \to n_u}|$ is the number of hops of the shortest WMN path between node n_O and node n_u; *(ii)* β_2 is the factor that allows to determine the maximum tolerance in the number of hops of a candidate WMN path between node n_O and node n_u, with respect to the number of hops of the shortest WMN path between these two nodes.

In our approach, the VN v'_u will be only created if there are a sufficiently large number of users, which we define by U_1, that would be better served if change their connections to the new VN v'_u, in order to avoid the creation of new VNs to serve single users; otherwise, and if user u declines to be connected to a default VN available in node n_u, user u will not be allowed to access the WMN. According to this strategy, within each traversed WMN path between

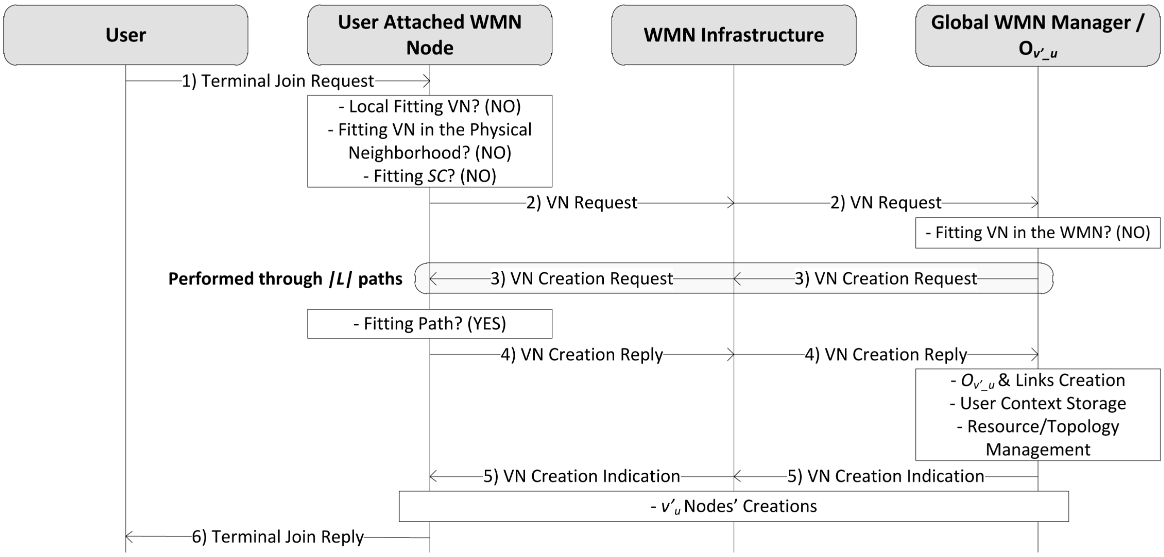


Figure 6.7: User Association & VN Control: VN Creation.

node n_O and node n_u, each WMN node inspects the number of users that would be better served by the VN v'_u, which is defined by $|U_{v'_u}|$. This process is performed by comparing the **C**ontext-aware **D**ifference (CD) among the context needs of these users against the context features of the VNs in which they are currently connected to, and against the context features of the VN v'_u (using the metric defined in expression (5.10)).

Based on the received replies, node n_u first limits the set of candidate WMN paths to the ones characterized by the highest value of $|U_{v'_u}| \geq U_1$. In the following, the WMN path to instantiate the VN v'_u is selected according to the metric defined in expression (6.2). Due to the resource-aware WMN path selection, the message described in Fig. 6.7→3 has to convey the level of resources available (e.g., bandwidth) in the traversed WMN path between node n_O and node n_u (which was already defined by RA_v in expression (6.1)).

In the following, node n_u notifies node n_O through the WMN path selected to instantiate the VN v'_u (Fig. 6.7→4). In this VN creation process, node n_O provides the initial support for the controller of the VN v'_u, $O_{v'_u}$, which has to be integrated in the distributed control ring according to the process depicted in Fig. 6.2. The location of $O_{v'_u}$ can be later updated based on the metric defined in expression (6.4).

After storing the context of user u, $O_{v'_u}$ cooperates with the global WMN manager to derive the resources that need to be assigned to the new virtual nodes from the VN v'_u. This resource information is then conveyed in the message sent by $O_{v'_u}$ to the WMN nodes where the creation of VN v'_u will take place (Fig. 6.7→5). At each of these WMN nodes, the users that are better served by the VN v'_u are notified to change their connections. Finally, user u is connected to VN v'_u (Fig. 6.7→6).

6.3.5 VN Remotion

As previously referred, several nodes from the VN $v \in V$ (or even the whole VN v) can be dynamically removed to react to the user and WMN dynamics. The VN remotion control

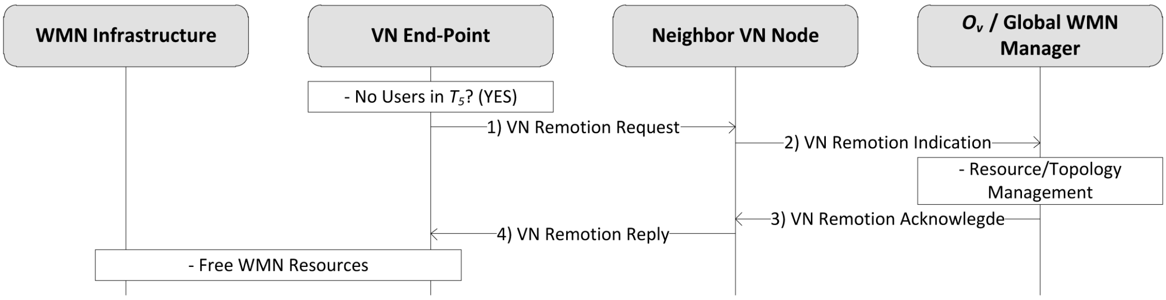


Figure 6.8: User Association & VN Control: VN Remotion.

process is described in Fig. 6.8.

In this control process, each end-point from the VN v (a virtual node with only one virtual connection) periodically inspects the connectivity status of the users connected to it. If there are no attached users in the timeout T_5, this end-point notifies its neighbor virtual node about its intention to leave the VN v (Fig. 6.8→1), and this virtual node forwards such notification to the controller of the VN v, O_v (Fig. 6.8→2). O_v then cooperates with the global WMN manager to free the wireless resources assigned to the end-point from the VN v, in order to be used by other VNs available in the WMN. After this process, the end-point from the VN v is notified that it will be completely removed (Fig. 6.8→3/4).

In case of the complete remotion of the VN v, the context-aware distributed ring structure is adapted to the leaving of O_v according to the process depicted in Fig. 6.5.

6.3.6 Real Implementation: Signaling Protocols

In Section 4.2.3, we provided a discussion on the signaling protocols available in the literature that can be used to implement the distributed VN discovery and extension mechanism proposed in Chapter 4. Similarly to that discussion, we will now present some guidelines for the real implementation of the distributed control mechanism proposed along this Section, to autonomously discover, select, adapt, create or remove VNs on-demand.

First, CXTP [286] can again be used to transfer the user context requirements between the user terminal and the WMN node where the user is connected to. Moreover, CXTP can also enable the transference of context information between the virtual nodes from a specific VN and the respective VN controller (this information can be related to the context of the users that are connected to such VN, or it can include quality indicators of the VN status).

The NSIS modules (GIST [291] and QoS NSLP [292]) can be installed in all WMN routers, which can then establish and manipulate E2E QoS-aware resource reservations in the WMN. As referred in Section 4.2.3, these resource reservations are important to allow the dynamic extension of VNs to be included in WMN nodes that do not support them. In the scope of the VN control mechanism proposed along this Section, the QoS NSLP signaling can also be used to dynamically create new VNs, and remove the unused ones to free their assigned resources.

Finally, we can resort to the Diameter protocol [289] to contact the AAA infrastructure, in order to allow an user to be authenticated in the WMN and in the controller of his/her associated VN, and also to authorize (or not) a NSIS-aware resource reservation.

6.4 Analytical Model: User Association to a Fitting VN

6.4.1 Goals & Methodology

After defining a mechanism to control and manage the VNs in a distributed way, it is important to assess the gains of such mechanism in comparison to centralized and decentralized control schemes. In this context, this Section defines an analytical model to compare several performance aspects of the centralized, decentralized and distributed VN control and management in specific WMN scenarios (published in [21]). First, the model analyzes the signaling load of each control mechanism to associate an user to a fitting VN available in the WMN, which can give us some insights on the delay of the user association to a fitting VN. Second, the model analyzes the overhead introduced by the signaling processes in the centralized, decentralized and distributed approaches for VN control and management.

6.4.2 Model Definition

6.4.2.1 Model Preliminaries

In the model definition, we will make use of the variables defined in Table 5.2, and the ones used along Sections 6.2 and 6.3 for the description of the distributed control approach.

In the first part of the model, which will be detailed in Section 6.4.2.2, we consider that an user $u \in U$ arrives at a random node $n_u \in N$. We also consider that the context level ($r_{u_c} \in R_u$), and the allowed variation range for such level ($t_{u_c} \in T_u$), requested by user u are randomly chosen from the set of possible levels M_c ($\forall c \in C$). At the time of arrival of user u at node n_u, we consider that $\overline{V}$ is the mean number of VNs available in the WMN, and $\overline{N_v}$ is the mean number of nodes per VN, which are uniformly and independently distributed in the WMN. Thus, user u can be automatically connected to any VN $v \in V_u$ already available in node n_u, being only triggered a VN update; otherwise, it can be triggered the local extension of any VN $v \in V_u$ available in one of the nodes located at a maximum distance of TTL_m hops, which is designated by the TTL_m-hop neighborhood of node n_u; finally, the global extension of any VN $v \in V_u$ available in the WMN can occur. Following this idea, we first derive the probability of a VN update, local VN extension and global VN extension to be triggered in the WMN. For the centralized, decentralized and distributed VN control approaches, we then derive the mean number of hops of each VN control process. Based on the probabilities to trigger each VN control process and on their mean number of hops, we derive closed-form expressions that measure the effort of user association in each approach.

In the second part of the model, which will be detailed in Section 6.4.2.3, we focus on the messages that are required to keep the VN knowledge updated in the architectural entities of the centralized, decentralized and distributed VN control approaches. By defining an expression that embeds the number of required messages, the length and the mean number of hops each message travels, we derive closed-form expressions for the overhead associated to each VN control approach.

Similarly to the models of the previous Chapters, we consider a grid-based topology to place $|N|$ WMN nodes, where each WMN node can directly communicate with other 4 nodes (except the border nodes). Since in this Chapter we allow the possibility of each node $n \in N$ to gather VN information tuples that are not only available on its 1-hop physical neighborhood, we consider that the mean number of nodes located at l hops of distance to the node $n \in N$, which compose the set $K_{(n,l)}$, is defined by (please refer to [304]):

$$\overline{|K_{(n,l)}|} = 4 \times l - \frac{4 \times l^2}{\sqrt{|N|}} + \frac{2 \times l \times (l^2 - 1)}{3 \times |N|}. \tag{6.9}$$

Finally, in the centralized VN control approach, the central WMN node, $O_c \in N$, controls the architecture; this optimal position aims to decrease the overall control delays. In the decentralized VN control approach, a set of WMN nodes composes an overlay to control the architecture, which is the defined by O_{de}; in such approach, p is the probability of node $n \in N$ to be part of O_{de}, and the overlay node located near to node $n \in N$ is defined by $o_n \in O_{de}$.

6.4.2.2 User Association to a Fitting VN

- **VN Control Processes Probabilities**

A) VN Update. When the user $u \in U$ arrives at the node $n_u \in N$, user u can trigger the update of any VN $v \in V_u$ available in node n_u.

As previously referred, this model considers that node n_u is randomly selected from the set N of WMN nodes, and the context requirements (R_u) and flexibility for context adaptation (T_u) of user u are also randomly assigned. Moreover, the $\overline{V}$ VNs are uniformly and independently distributed in the WMN. Therefore, the probability of a virtual node available in the physical node n_u to be part of any VN $v \in V_u$, P_u, is defined by:

$$P_u = \frac{|V_u|}{|V|}, \tag{6.10}$$

where the number of possible VNs, $|V|$, is defined by $|M_c|^{|C|}$, since a VN can be created to meet any set of user context levels, and the number of those VNs that are fitting ones for user u, $|V_u|$, is derived from the sets R_u and T_u.

Since the $\overline{V}$ VNs are uniformly and independently distributed in the WMN, the mean number of virtual nodes per physical node can be defined by $\overline{n_v}$=$(\overline{V}\times\overline{N_v})/|N|$. Thus, the probability of user u to trigger a VN update can be given by the complementary probability of none of the $\overline{n_v}$ virtual nodes available in the physical node n_u, to be part of any VN $v \in V_u$, which is defined by:

$$P_{UP_u} = 1 - (1 - P_u)^{\overline{n_v}}. \tag{6.11}$$

Considering any possible user $u \in U$ to access the WMN, each one characterized by the sets of requirements R_u and T_u, the probability of a VN update process to be started in the WMN, P_{UP}, is calculated by averaging all the probabilities P_{UP_u} associated to the elements of the set U. P_{UP} is then defined by:

$$P_{UP} = \frac{1}{|U|} \times \sum_{u \in U} P_{UP_u}. \tag{6.12}$$

B) Local VN Extension. If node n_u does not contain a virtual node from any VN $v \in V_u$, a local VN extension can be triggered if any VN $v \in V_u$ is available in any of the neighbors of node n_u located at a maximum distance of TTL_m hops. According to the definition of expression (6.9) for the mean number of nodes located at l hops of distance to the node $n \in N$, the mean number of neighbors of node n_u located at a maximum distance of TTL_m hops, is defined by:

$$\overline{K} = \sum_{l=1}^{TTL_m} \overline{|K_{(n,l)}|}. \tag{6.13}$$

Therefore, the probability of user u to trigger a local VN extension, P_{LE_u}, can be given by the probability of not triggering a VN update $(1 - P_{UP_u} = (1 - P_u)^{\overline{n_v}})$, times the probability of at least one of the $\overline{n_v}$ virtual nodes available in any of the $\overline{K}$ neighbors of node n_u, to be part of any VN $v \in V_u$. This last probability can be given by the complementary probability of none of the $\overline{n_v}$ virtual nodes available in any of the $\overline{K}$ neighbors of node n_u, to be part of any VN $v \in V_u$, which is defined by $1 - (1 - P_u)^{\overline{n_v} \times \overline{K}}$. P_{LE_u} is then defined by:

$$P_{LE_u} = (1 - P_u)^{\overline{n_v}} \times (1 - (1 - P_u)^{\overline{n_v} \times \overline{K}}). \tag{6.14}$$

Considering any possible user $u \in U$ to access the WMN, each one characterized by the sets of requirements R_u and T_u, the probability of a local VN extension process to be started in the WMN, P_{LE}, is calculated by averaging all the probabilities P_{LE_u} associated to the elements of the set U. P_{LE} is then defined by:

$$P_{LE} = \frac{1}{|U|} \times \sum_{u \in U} P_{LE_u}. \tag{6.15}$$

C) Global VN Extension. If a VN $v \in V_u$ is not available in node n_u, neither in the TTL_m-hop neighborhood of node n_u, a global VN discovery is started in the WMN.

According to the global VN discovery process described in Section 6.3.3, node n_u first inspects the stored information tuples about the VNs available on it or on its TTL_m-hop neighborhood; this can enable the redirection of the global discovery process to any $SC \in SC_u$ in which can be found a point of attachment from any VN $v \in V_u$. Since the VNs are uniformly and independently distributed in the WMN, the probability of a VN available in the node $n \in N$, to be part of any $SC \in SC_u$, P'_u, is defined by:

$$P'_u = \frac{|SC_u|}{|SC|} = \frac{|SC_u|}{\prod_{c \in C'} |M_c|}, \tag{6.16}$$

where $|SC|$ is the number of possible SCs that compose the ring structure, and $|SC_u|$ is the number of fitting SCs for user u, which is derived from the sets R_u and T_u when considering the set of context features C'.

Therefore, the probability of user u to trigger a global VN extension, P_{GE_u}, can be given by the product of three terms: *(i)* the probability of not triggering a VN update in node n_u $(1 - P_{UP_u} = (1 - P_u)^{\overline{n_v}})$; *(ii)* the probability of not triggering a local VN extension of any of the VNs available in the TTL_m-hop neighborhood of node n_u $((1 - P_u)^{\overline{n_v} \times \overline{K}})$; *(iii)* the probability of at least one of the VNs available in node n_u, on its TTL_m-hop neighborhood, to be part of any $SC \in SC_u$. This last probability can be given by the complementary probability of none of the $\overline{n_v}$ virtual nodes available in the physical node n_u or in any of its $\overline{K}$ neighbors, to be part of a VN belonging to any $SC \in SC_u$, which is defined by $1 - (1 - P'_u)^{[\overline{n_v} \times (1 + \overline{K})]}$. P_{GE_u} is then defined by:

$$P_{GE_u} = (1 - P_u)^{\overline{n_v}} \times (1 - P_u)^{\overline{n_v} \times \overline{K}} \times (1 - (1 - P'_u)^{[\overline{n_v} \times (1 + \overline{K})]}). \tag{6.17}$$

Considering any possible user $u \in U$ to access the WMN, each one characterized by the sets of requirements R_u and T_u, the probability of a global VN extension process to be started in the WMN, P_{GE}, is calculated by averaging all the probabilities P_{GE_u} associated to the elements of the set U. P_{GE} is then defined by:

$$P_{GE} = \frac{1}{|U|} \times \sum_{u \in U} P_{GE_u}. \tag{6.18}$$

In the following, we define closed-form expressions for the mean number of hops of a VN update, local VN extension and global VN extension when opting for the centralized, decentralized and distributed VN control. Such expressions are presented in Algorithm 3, and they are based on the sum of the mean number of hops traveled by the messages involved in each process in the respective control approach, which are described in Fig. 6.9.

- **Centralized Control**

When user u arrives at node n_u, node n_u contacts the central WMN controller, which was already defined in Section 6.4.2.1 by O_c. If O_c has information of any VN $v \in V_u$ available in node n_u, O_c triggers a VN update in node n_u. This process then involves two messages performed between node n_u and O_c (Fig. 6.9→1/2), each one traveling L_c hops in the WMN (Alg.3-line2). Due to the random location of node n_u, we consider that L_c is the mean hop distance between any WMN node $n \in N$ and the central WMN node, where O_c is located.

In case of a local VN extension (Alg.3-line3), O_c notifies one of the $\overline{K}$ neighbors of node n_u located at a maximum distance of TTL_m hops, that supports the VN $v \in V_u$ selected according to the metric defined in expression (6.2) (this notification is described in Fig. 6.9→3, and it again travels L_c hops). Since this model does not focus on data communications, it will not consider the effects of the resource-aware metric RA_v in the expression (6.2); however, since the VNs, VN nodes and users are uniformly and independently distributed in the WMN, this simplification will not have a high impact on the model results. Considering that any of the $\overline{K}$ neighbors of node n_u can be selected with equal probability, the selected neighbor then contacts node n_u (Fig. 6.9→4) by sending a message that travels the mean number of hops:

$$L_k = \frac{1}{\overline{K}} \times \sum_{l=1}^{TTL_m} \left(\overline{|K_{(n,l)}|} \times l \right). \tag{6.19}$$

If a global VN extension occurs (Alg.3-line4), we consider that it is performed from one of the nodes available at TTL_m+1 hops of node n_u, since VN extensions involving more nodes are very expensive. Therefore, O_c notifies one of these nodes by sending a message that travels L_c hops (Fig. 6.9→5), and finally, node n_u is contacted (Fig. 6.9→6).

Based on the probability and mean number of hops of each VN control process, Alg.3-line5 presents the closed-from expression for the mean number of hops of the centralized mechanism for user association, which is defined by L_{ce}.

- **Decentralized Control**

When user u arrives at node n_u, node n_u contacts its associated overlay node, which was already defined in Section 6.4.2.1 by $o_{n_u} \in O_{de}$. In case of a VN update, o_{n_u} then replies to node n_u. Thus, a VN update requires two messages (Fig. 6.9→7/8) performed between

Algorithm 3: Mean Number of Hops of the Control Mechanism for User Association.

```
1  if (Centralized Approach) then
2      if (VN Update) then $L_{UPce} = 2 \times L_c$;
3      if (Local VN Extension) then $L_{LEce} = 2 \times L_c + L_k$;
4      if (Global VN Extension) then $L_{GEce} = 2 \times L_c + TTL_m + 1$;
5      $L_{ce} = P_{UP} \times L_{UPce} + P_{LE} \times L_{LEce} + P_{GE} \times L_{GEce}$;
6  else
7      if (Decentralized Approach) then
8          if (VN Update) then $L_{UPde} = 2 \times L_d$;
9          if (Local VN Extension) then $L_{LEde} = 2 \times L_d + L_k$;
10         if (Global VN Extension) then $L_{GEde} = 2 \times L_d + 2 \times L_{c'} + TTL_m + 1$;
11         $L_{de} = P_{UP} \times L_{UPde} + P_{LE} \times L_{LEde} + P_{GE} \times L_{GEde}$;
12     else
13         if (Distributed Approach) then
14             if (VN Update) then $L_{UPdi} = 2 \times L_v$;
15             if (Local VN Extension) then $L_{LEdi} = 2 \times L_k + 2 \times L_v$;
16             if (Global VN Extension) then $L_{GEdi} = M_2 \times (L_{k_1} + L_v + M_1 \times L_r) + L_v + TTL_m + 1$;
17             $L_{di} = P_{UP} \times L_{UPdi} + P_{LE} \times L_{LEdi} + P_{GE} \times L_{GEdi}$;
```

node n_u and the overlay node o_{n_u}, each one traveling L_d hops (Alg.3-line8). Since p is the probability of node $n \in N$ to be part of O_{de}, the mean hop distance between node n_u and its associated overlay node, L_d, is then defined by:

$$L_d = \sum_{l=1}^{L_{d'}} \prod_{j=0}^{l-1} \Big((1-p)^{\overline{|K_{(n,j)}|}} \times p \times \overline{|K_{(n,l)}|} \times l \Big), \tag{6.20}$$

where $L_{d'}$ is the maximum hop distance, that this model considers, between node n_u and its associated overlay node. To define $L_{d'}$, we know that the probability of node $n \in N$ to not be located at more than $L_{d'}$ hops of any O_{de} node is $P'{=}1{-}(1{-}p)^{[\overline{|K_{(n,0)}|}+\overline{|K_{(n,1)}|}+\ldots+\overline{|K_{(n,L_{d'})}|}]}$. Considering $|N|{=}100$, $L_{d'}{=}3$ implies $P'{\approx}0.9$; therefore, we assume $L_{d'}{=}3$ as the maximum hop distance between node $n \in N$ and its associated overlay node.

The local VN extension process is very similar to the one of a centralized approach (Alg.3-line9). In this process, the overlay node o_{n_u} notifies one of the $\overline{K}$ neighbors of node n_u located at a maximum distance of TTL_m hops, that supports the VN $v \in V_u$ selected according to the metric defined in expression (6.2) (this notification is described in Fig. 6.9→9, and it again travels L_d hops). Considering that any of the $\overline{K}$ neighbors of node n_u can be selected with equal probability, the selected neighbor then contacts node n_u (Fig. 6.9→10) by sending a message that travels L_k hops.

In a global VN extension (Alg.3-line10), the overlay node o_{n_u} uses the control overlay O_{de} to find the point of attachment of any VN $v \in V_u$ in other part of the WMN (Fig. 6.9→11). Due to the random selection of the nodes that are part of O_{de}, the mean number of hops traveled by an overlay communication, $L_{c'}$, is defined by the mean hop distance between any pair of nodes in a grid WMN as $L_{c'}{=}(2{\times}\sqrt{|N|})/3$ (please refer to [304]). Similarly to the centralized approach, after receiving the reply from another overlay node (Fig. 6.9→12), the overlay node o_{n_u} contacts one of the nodes available at $TTL_m{+}1$ hops of node n_u (Fig. 6.9→13). The contacted node then notifies node n_u (Fig. 6.9→14).

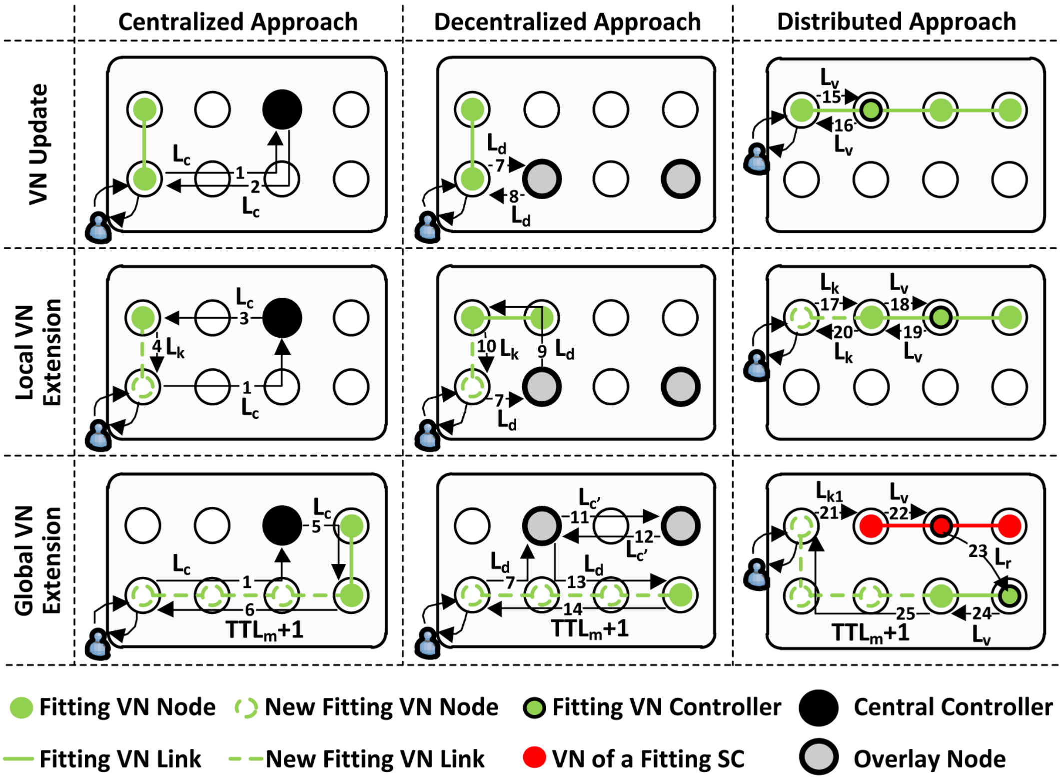


Figure 6.9: VN Update, or Local/Global VN Extension according to the Centralized, Decentralized and Distributed VN Control Approaches.

Based on the probability and mean number of hops of each VN control process, Alg.3-line11 presents the closed-from expression for the mean number of hops of the decentralized mechanism for user association, which is defined by L_{de}.

- **Distributed Control**

When user u arrives at node n_u, and node n_u supports any VN $v \in V_u$, a VN update is triggered. In this case, the virtual node from the selected VN $v \in V_u$ contacts the controller of this VN, O_v, which then replies to the virtual node from the VN $v \in V_u$ supported by node n_u. Such process then requires two messages between a VN node and the respective VN controller (Fig. 6.9→15/16), each one traveling L_v hops (Alg.3-line14). Since this model focuses on random VN topologies, L_v is the mean number of hops between two arbitrary nodes in $\overline{N_v}$ from a VN $v \in V$, which is defined by $L_v = \sqrt{(\overline{N_v}/\log \overline{N_v})}$ (please refer to [279]).

If a local VN extension occurs, node n_u contacts one of its $\overline{K}$ neighbors located at a maximum distance of TTL_m hops, that supports any VN $v \in V_u$ selected according to the metric defined in expression (6.2). Then, the controller of the selected VN is contacted, and the process returns to node n_u. Similarly to the centralized and decentralized approaches, we consider that any of the $\overline{K}$ neighbors of node n_u can be selected with equal probability, and thus, the local VN extension implies four messages (Alg.3-line15): two of them travel L_k hops (Fig. 6.9→17/20), and the remaining travel L_v hops (Fig. 6.9→18/19).

In case of a global VN extension (Alg.3-line16), node n_u first inspects the stored information tuples about the VNs available on it or on its TTL_m-hop neighborhood; this can enable the redirection of the global discovery process to any $SC \in SC_u$ in which can be found a point of attachment from any VN $v \in V_u$. Considering that node n_u, or one of its $\overline{K}$ neighbors located at a maximum distance of TTL_m hops, can be selected with equal probability to start the global VN discovery process within the ring structure, the message used to perform this process (Fig. 6.9→21) travels the mean number of hops:

$$L_{k_1} = \frac{1}{1+\overline{K}} \times \sum_{l=0}^{TTL_m} \left(\overline{|K_{(n,l)}|} \times l \right). \tag{6.21}$$

In the following, the virtual node from the selected VN notifies the respective VN controller by sending a message that travels L_v hops (Fig. 6.9→22).

Within the selected $SC \in SC_u$, the global VN discovery is redirected through the available VN controllers until finding the controller of any VN $v \in V_u$ (Fig. 6.9→23). Since in this model, VNs are uniformly and independently distributed in the WMN, the mean number of hops between two consecutive VN controllers, L_r, has to be considered as the mean hop distance between any pair of WMN nodes, which was already defined by $L_{c'}$; however, to be aware of the intelligent VN controller selection of expression (6.4), we consider L_r=$L_{c'}/2$.

Due to the independent selection of the VNs available in the WMN, they are uniformly distributed among the SCs. Therefore, the mean number of communications performed through each SC direction until finding the controller of any VN $v \in V_u$, M_1, is defined by:

$$M_1 = \frac{1}{2} \times \left(\frac{\overline{V}}{|SC|} - 1 \right) = \frac{1}{2} \times \left(\frac{\overline{V}}{\prod\limits_{c \in C'} |M_c|} - 1 \right). \tag{6.22}$$

If the controller of any VN $v \in V_u$ is not available in the selected SC, the process is redirected to another $SC \in SC_u$. Considering any possible user $u \in U$ to access the WMN, each one characterized by the sets of requirements R_u and T_u, the number fitting SCs for the user u that are accessed in a distributed VN discovery until finding any VN $v \in V_u$, M_2, can be modeled by:

$$M_2 = \frac{1}{|U|} \times \sum_{u \in U} \left(\frac{|SC_u| + 1}{2} \right). \tag{6.23}$$

After discovering any VN $v \in V_u$, it is contacted the virtual node from the discovered VN $v \in V_u$ that is located near to node n_u (Fig. 6.9→24). Similarly to the centralized and decentralized approaches, we again assume that this node is located at TTL_m+1 hops of node n_u. Finally, node n_u is notified (Fig. 6.9→25). The mean number of hops traveled by the messages of these two processes were already modeled by L_v and TTL_m+1, respectively.

Based on the probability and mean number of hops of each VN control process, Alg.3-line17 presents the closed-from expression for the mean number of hops of the distributed mechanism for user association, which is defined by L_{di}.

6.4.2.3 Overhead of VN Control

In the following, we define closed-form expressions for the overhead associated to the centralized, decentralized and distributed approaches for VN control (see Algorithm 4).

Algorithm 4: Overhead of a VN Control Approach.

```
1  if (Centralized Approach) then
2      $Z_{ce} = \underbrace{|N|}_{a} \times \underbrace{(2 \times B + P \times \overline{n_v})}_{b} \times \underbrace{L_c}_{c};$
3  else if (Decentralized Approach) then
4      $Z_{de} = \underbrace{|N|}_{a} \times \underbrace{(2 \times B + P \times \overline{n_v})}_{b} \times \underbrace{L_d}_{c} +$
5      $\sum_{l=1}^{TTL_m} (\underbrace{|O_{de}| \times \overline{|K_{(n,L_{d'}+l)}|}}_{a} \times \underbrace{(2 \times B + P \times \overline{n_v})}_{b} \times \underbrace{(L_{d'} + l)}_{c})+$
6      $\underbrace{|O_{de}| \times 2}_{a} \times \underbrace{B}_{b} \times \underbrace{L_{c'}}_{c};$
7  else if (Distributed Approach) then
8      $Z_{di} = \sum_{l=1}^{TTL_m} (\underbrace{|N| \times \overline{|K_{(n,l)}|}}_{a} \times \underbrace{(B + P \times \overline{n_v})}_{b} \times \underbrace{l}_{c})+$
9      $\underbrace{\overline{V} \times (\overline{N_v} - 1)}_{a} \times \underbrace{(2 \times B + P)}_{b} \times \underbrace{L_v}_{c} +$
10     $\underbrace{\overline{V} \times 2}_{a} \times \underbrace{B}_{b} \times \underbrace{L_r}_{c};$
```

These overhead expressions represent the total amount of traffic required to periodically update the VN knowledge in the architectural elements with control responsibilities. According to Algorithm 4, each portion of such expressions represents the traffic imposed by a single control maintenance process which is modeled as the product of three terms: *(a)* the total amount of instances of such process; *(b)* the length of the messages of such process; *(c)* the mean number of hops each message travels.

- **Centralized Control**

In the centralized control approach, the WMN controller O_c periodically contacts each node $n \in N$ to refresh its VN knowledge, and, as a reply, each node sends the information of the $\overline{n_v}$ virtual nodes that it supports; both messages travel L_c hops (Alg.4-line2). In this model, we consider that every control packet has the size of B bytes, and the ones conveying information of virtual nodes have an extra size of $P{=}B/2$ bytes per virtual node.

- **Decentralized Control**

In the decentralized control approach, the overhead is a result of three distinct network processes. First, each node $n \in N$ is periodically notified by its associated overlay node, $o_n \in O_{de}$, and replies by sending the information of the $\overline{n_v}$ virtual nodes that it supports; both messages travel L_d hops (Alg.4-line4). Second, each overlay node $o \in O_{de}$ ($|O_{de}|{=}|N|{\times}p$) contacts the WMN nodes that are under the TTL_m-hop neighborhood of the WMN nodes of its covered area, but have associated another overlay node. These WMN nodes then reply with the information of the virtual nodes that they support (Alg.4-line5); this allows each overlay node to trigger local VN extensions from any of the $\overline{K}$ neighbors, located at a maximum distance of TTL_m hops, of the WMN nodes of its covered area. Third, each overlay node

$o \in O_{de}$ has to refresh the routes with its 1-hop overlay neighbors. In this model, each overlay node stores routes to contact 2 other random overlay nodes, and the mean number of hops traveled by an overlay communication was already defined by $L_{c'}$ (Alg.4-line6).

- **Distributed Control**

In the distributed control approach, the overhead is again a result of three parts. First, each node $n \in N$ has to disseminate the information of the $\overline{n_v}$ virtual nodes that it supports to its $\overline{K}$ neighbors located at a maximum distance of TTL_m hops (Alg.4-line8). Second, the controller of each VN $v \in V$, O_v, periodically contacts the other virtual nodes from the VN v, by sending a message that travels L_v hops; these virtual nodes then reply to O_v with its information (Alg.4-line9). Third, O_v announces itself to O_{v_s} and O_{v_p} by sending a message that travels L_r hops, in order to update the distributed ring structure (Alg.4-line10).

6.4.3 Evaluation & Discussion

In this Section, we resort to the analytical model defined in Section 6.4.2 to evaluate the mean signaling load of the centralized, decentralized and distributed control mechanism to associate an user to a fitting VN in specific WMN scenarios. The signaling effort for user association can give us some insights on the mean signaling delay of these three control mechanisms. Moreover, we analyze the signaling overhead associated to the centralized, decentralized and distributed approaches for VN control and management.

In the evaluation of these two performance aspects of each VN control approach, we measure the impact of varying: *(i)* the size of the WMN; *(ii)* the WMN load in terms of the mean number of VNs available in the WMN at the time of arrival of a new user, and the mean number of virtual nodes per VN; *(iii)* the number of context features that drive the multi-VN architecture; *(iv)* the TTL_m parameter that influences the probability to trigger a local or a global VN extension in each control approach, as well as the amount of control information disseminated in the WMN in the decentralized and distributed VN control approaches; *(v)* the number of context features that compose the sets C' and $\neg C'$, which influences the number of context parameters that are more or less flexible from the user side, and the organization of the control ring structure in the distributed VN control approach.

Finally, we make use of the simulation framework presented in Section 3.4, in order to assess the accuracy of the analytical results when compared against the ones obtained from a specific simulation setup.

6.4.3.1 Model Evaluation

We start by defining all the variables and details of this analytical study, which are summarized in Table 6.1.

We apply the analytical model in grid-based WMN topologies with $|N| = \{12 \times 12, 14 \times 14, 16 \times 16\}$ nodes. We vary the number of context parameters that drive the architecture, $|C|$, and the number of such parameters that admit low values of context flexibility from the user side (t_{u_c}), $|C'|$, performing distinct tests for the combinations: $(|C|, |C'|) = \{(4,2), (4,3), (5,2)\}$. The set of levels of each context parameter $c \in C$ is $M_c = \{1,2,3,4\}$.

The results are obtained by considering the set of users U, where each user $u \in U$ arrives at a random node $n_u \in N$, and is characterized by particular context requirements (R_u and T_u): *(i)* $r_{u_c} \in R_u$ can take any value of from the set M_c; *(ii)* $t_{u_c} \in T_u$ ($c \in C'$) can take any

Table 6.1: Analytical Model Details: User Association to a Fitting VN.

Parameter	**Value**
$\|N\|$	$\{12 \times 12, 14 \times 14, 16 \times 16\}$ Grid WMN
n_u	Randomly chosen from the set N
$(\|C\|, \|C'\|)$	$\{(4,2),(4,3),(5,2)\}$
M_c $(c \in C)$	$\{1,2,3,4\}$
$r_{u_c} \in R_u$ $(c \in C)$	Randomly chosen from the set M_c
$t_{u_c} \in T_u$ $(c \in C')$	Randomly chosen from the set $\{0,1\}$
$t_{u_c} \in T_u$ $(c \in \neg C')$	Randomly chosen from the set $\{1,2\}$
$/V$	$\{30,35,40,45,50,55,60\}$
$/N_v$	$\{6,7,8\}$
TTL_m	$\{1,2,3\}$
p	0.1

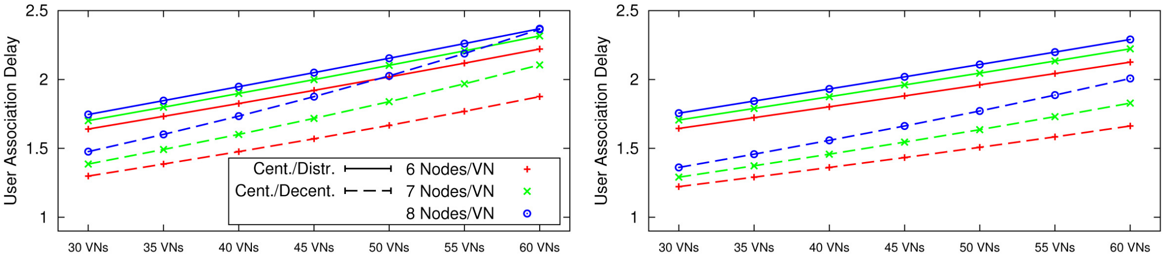
value from the set $\{0,1\}$, since these are the parameters that admit low variation ranges for their demands during the user connectivity time; *(iii)* $t_{u_c} \in T_u$ $(c \in \neg C')$ can take any value from the set $\{1,2\}$, since these are the parameters that admit more distinct levels during the user connectivity time. Moreover, we consider $\overline{V} = \{30,35,40,45,50,55,60\}$ for the mean number of VNs in the WMN, and $\overline{N_v} = \{6,7,8\}$ for the mean number of nodes per VN. Finally, we assume $p = 0.1$ and perform distinct tests for $TTL_m = \{1,2,3\}$.

- **Results & Discussion**

In Fig. 6.10, we show the ratio between the signaling effort (or the signaling delay) of the centralized control mechanism to associate an user to a fitting VN, and the same effort obtained with the distributed and decentralized control mechanisms. In Fig. 6.11, we present the ratio between the overhead associated to the centralized VN control approach, and the overhead associated to the distributed and decentralized VN control approaches. The results of these two figures are respectively defined by $\frac{L_{ce}}{L_{di}}/\frac{L_{ce}}{L_{de}}$ and $\frac{Z_{ce}}{Z_{di}}/\frac{Z_{ce}}{Z_{de}}$ according to the variables of the analytical model, and they are obtained for $|C|/|C'| = 4/2$, $TTL_m = 2$ and varying the number WMN nodes ($|N|$), mean number of VNs ($\overline{V}$) and virtual nodes per VN ($\overline{N_v}$).

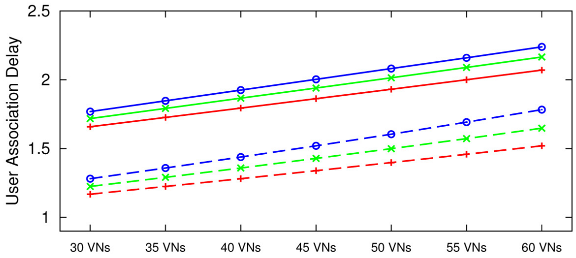
From the results obtained in Fig. 6.10, we observe the improvement of both distributed and decentralized control approaches over the centralized one in terms of the time required to connect an user to a fitting VN available in the WMN; the user association delays in the distributed approach can be lower than half of the ones obtained in the centralized approach. This is explained by the possibility to take advantage of the disseminated knowledge to perform local VN associations without always contacting a central unit. Concerning the overhead results presented in Fig. 6.11, the centralized scheme imposes a lower amount of control traffic than the other schemes, which is expected due to the need to update the control overlays in the latter schemes. However, the overhead of the non-centralized schemes is lower than twice the one of the centralized scheme. There is then a trade-off between delay gains and overhead drawbacks of non-centralized schemes with respect to the centralized one.

Still from Fig. 6.10, we conclude that the distributed approach is characterized by lower user association delays than the decentralized approach, mostly because of the fast redirection of global VN discoveries to proper *SCs* of the ring. These results are relative to the initial user association; in case of the need to perform the association of the user to another VN due



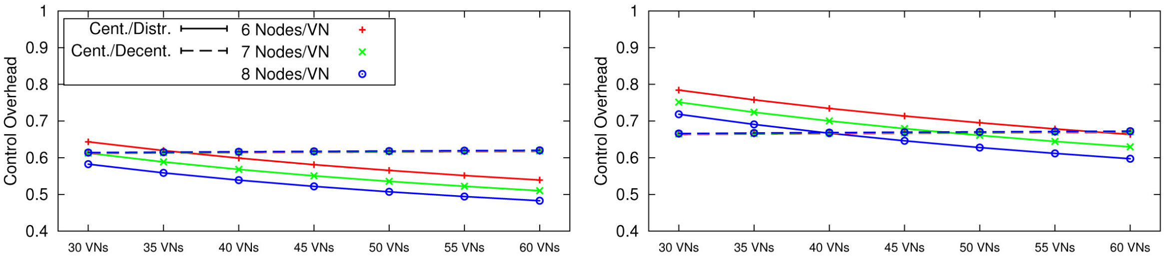
(a) 12x12 WMN Nodes

(b) 14x14 WMN Nodes



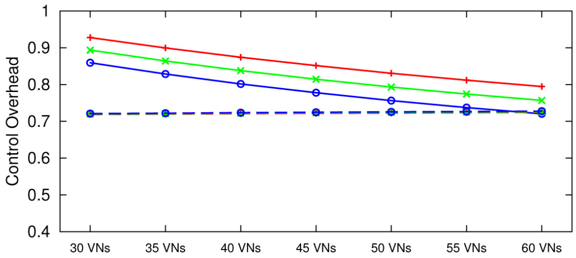
(c) 16x16 WMN Nodes

Figure 6.10: Analytical Model: User Association Delay of the Distributed or Decentralized Mechanism in comparison to the Centralized One, when using $|C|/|C'| = 4/2$ and $TTL_m = 2$ for a Grid-Based WMN with 12×12 Nodes (a), 14×14 Nodes (b) and 16×16 Nodes (c), and varying the Number of VNs in the WMN and the Number of Nodes per VN.



(a) 12x12 WMN Nodes

(b) 14x14 WMN Nodes



(c) 16x16 WMN Nodes

Figure 6.11: Analytical Model: Signaling Overhead of the Distributed or Decentralized VN Control Approach in comparison to the Centralized One, when using $|C|/|C'| = 4/2$ and $TTL_m = 2$ for a Grid-Based WMN with 12×12 Nodes (a), 14×14 Nodes (b) and 16×16 Nodes (c), and varying the Number of VNs in the WMN and the Number of Nodes per VN.

to user context change, we believe that the context-aware organization of the ring will further increase the performance of the distributed approach. With respect to the overhead (see Fig. 6.11), it is slightly lower in the decentralized approach than in the distributed one in the case of a 12 × 12 WMN; however, such tendency is not observed for large size WMNs, mainly because of the path length increase among overlay nodes in the decentralized approach. In smaller WMNs, the decentralized approach yields similar values (sometimes even smaller) in terms of both delay and overhead. As expected, distributed approaches are more relevant in larger WMNs.

The increase of the WMN size has small effect on the decrease of the user association delays of the non-centralized schemes with respect to the centralized one. Such is explained since the increase of WMN size implies a high dispersion of VNs in the WMN. This increases the probability to start global VN extensions that take more time to be performed than VN updates or local VN extensions. With regard to overhead, we observe the decrease of the gains of the centralized scheme over the distributed scheme when the WMN size increases. This leads us to conclude that the increase of the path length to each node $n \in N$ access to the WMN controller O_c has a worse effect than the increase of the path length between VN controllers in the ring.

For the same WMN size, the increase of the mean number of VNs ($\overline{V}$) and the mean number of virtual nodes per VN ($\overline{N_v}$) enables the increase of the gains of the non-centralized schemes. This is explained since the high VN density provides the existence of more VN updates or local VN extensions, which are less complex than global VN extensions.

In the following, Fig. 6.12 and Fig. 6.13 show the signaling delay and overhead ratios between the centralized and the distributed and decentralized control approaches, when we respectively use distinct values of TTL_m and combinations of $|C|/|C'|$. The results of these two figures are obtained for a WMN with $|N| = 14 \times 14$ nodes and varying the mean number of VNs with $\overline{N_v} = 7$ virtual nodes.

From the results obtained in Fig. 6.12, we observe that the increase of the number of hops to forward VN information in the WMN ($TTL_m = 3$) clearly allows to obtain lower user association delays in the non-centralized schemes, since there is more information available to start local VN extensions that take less time than global VN extensions. However, the overhead highly increases, being the overhead of the distributed scheme slightly higher than the one of the decentralized scheme. When $TTL_m = 1$, the worse behavior of the non-centralized schemes in the user association is explained by the need to perform more global VN extensions; however, these schemes now present overhead gains over the centralized one.

Fig. 6.13 shows that for the same number of context parameters that drive the multi-VN architecture ($|C|$), the existence of a higher number of user context features admitting low values of flexibility (high $|C'|$) leads to lower user association delays of the non-decentralized schemes over the centralized one. This happens due to the decrease of the probability to update or locally extend VNs (decrease of P_u, as described in expression (6.10)), and the consequent increase of the number of global VN extensions. Moreover, for the same WMN size and mean number of VNs ($\overline{V}$), the increase of the number of context parameters that drive the multi-VN architecture ($|C|$) reduces the user association delays of both distributed or decentralized schemes over the centralized one. Such is explained by the increase of the probability to have more VNs in the user physical neighborhood that do not fit the user context requirements, leading to a high number of global VN extensions. Finally, the overhead is not affected by varying the number of context parameters that characterize the sets C or C', since the mean number of virtual nodes in the WMN is the same.

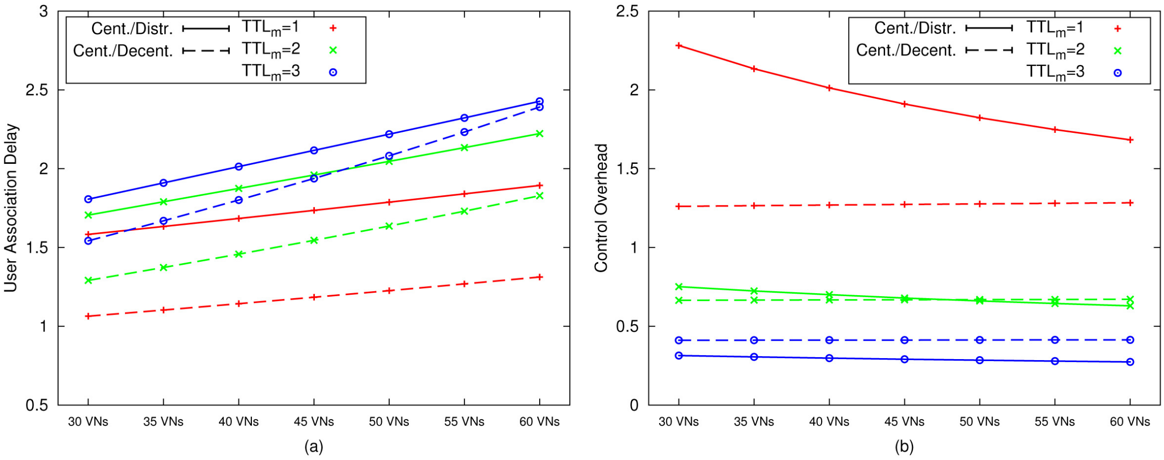


Figure 6.12: Impact of using Distinct Values of TTL_m on User Association Delay (a) and Control Overhead (b) of the Distributed or Decentralized VN Control Approach in comparison to the Centralized One, when using $|C|/|C'| = 4/2$ for a Grid-Based WMN with 14×14 Nodes, and varying the Number of VNs with $\overline{N_v} = 7$ Virtual Nodes.

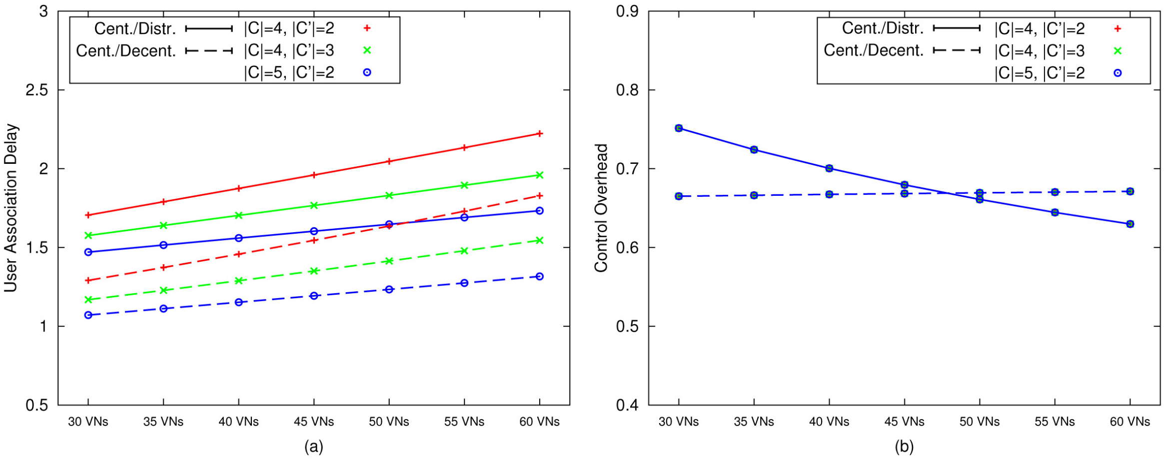


Figure 6.13: Analytical Model: Impact of using Distinct Combinations of $|C|/|C'|$ on User Association Delay (a) and Control Overhead (b) of the Distributed or Decentralized VN Control Approach in comparison to the Centralized One, when using TTL_m=2 for a Grid-Based WMN with 14×14 Nodes, and varying the Number of VNs with $\overline{N_v}$=7 Virtual Nodes.

6.4.3.2 Model Validation

In order to evaluate the accuracy of the analytical model, we implemented the centralized VN control approach, and a preliminary version of the distributed VN control approach described in Sections 6.2 and 6.3, in the simulation platform presented in Section 3.4. These two implementations are meant to be directly comparable with the ones modeled in Section 6.4.2, which contemplate the VN update, local VN extension and global VN extension control processes. Therefore, in the implementation of the distributed VN control mechanism, we do not include the resource-aware metric of the WMN path selection function defined in expression (6.2), since it is not considered in the analytical model.

To this purpose, we create a grid-based WMN topology with 10×10 nodes, where each node has a transmission radius of $100m$. To model the WMN dynamics, the arrival rate of the users at the WMN is a Poisson process with mean inter-arrival time of $1/\lambda$. We consider $|C|/|C|' = 4/2$, and each user $u \in U$ is characterized by a random set of context requirements (R_u and T_u) defined according to Table 6.1. Each user $u \in U$ arrives at a random node $n_u \in N$, and he/she is connected for a time following an exponential distribution with mean $1/\mu = 1/20$. From the queueing theory, the mean number of users in the WMN is defined by $\overline{U} = \lambda/\mu$; thus, we configure λ to perform tests for a mean number $\overline{U} = \{40, 45, 50, 55, 60, 65, 70, 75, 80\}$ of users accessing the WMN.

The processes to update the VN knowledge among the distinct architectural elements with control responsibilities in the centralized and distributed approaches (these processes were described in Section 6.4.2.3) are performed every $200ms$, and we consider $TTL_m = 2$. We also implement the VN creation and remotion processes to deal with the user dynamics: *(i)* when an user arrives at the WMN, and there are no fitting VNs available, the exactly fitting VN for such user is created in the shortest path between the central WMN node and the node where the user is connected to (in these simulations, we do not incorporate the VN creation criteria proposed in Section 6.3.4); *(ii)* the timeout to trigger the remotion of VNs when there are no attached users is $500ms$.

By running the simulation setup during $1000s$ for each control approach (each obtained value is the mean of 20 runs with a confidence degree of 90%), we determine the number and the mean signaling delay of VN updates, local VN extensions and global VN extensions. The probability of each VN control process is then defined as the ratio between the number of times that it occurs in relation to the total number of the three VN control processes. In the following, we derive L_{ce} and L_{di}, and compare such values against the analytical results.

Finally, the analytical model considers the mean number of VNs available in the WMN and the mean number of nodes per VN. To directly compare the analytical results against the simulation ones, we first access, every $10ms$, to the number of VNs and nodes per VN, and then derive their mean values for each simulation.

- **Results & Discussion**

The analytical and simulation ratios between the signaling delay of the centralized and the distributed control approaches are presented in Fig. 6.14, in which we can observe the accuracy of the analytical model through the closeness between the results.

Small discrepancies between the results were already expected. First, the simulation study does not consider shortest paths. In the simulation environment, with the increase of the mean number of users accessing the WMN, there is a higher wireless collision probability. This fact leads to the increase of the number of hops traveled by the messages required to perform both centralized and distributed schemes for user association, which leads to simulation gains slightly lower than the analytical ones. Second, due to the probabilistic nature of the analytical model, the analytical gains of the distributed approach are slightly lower than the ones obtained by simulations, when the WMN is accessed by a small mean number of users (small number of VNs and nodes per VN).

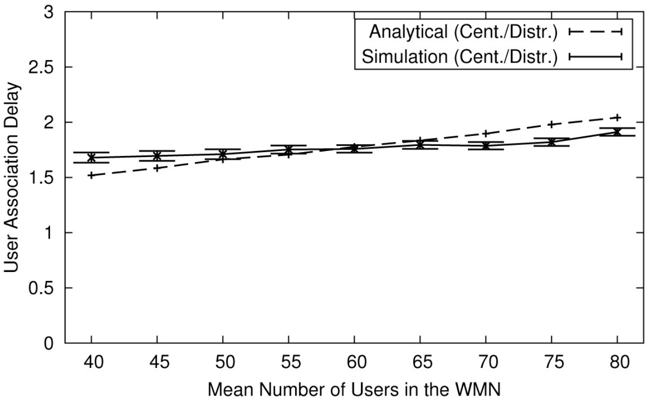


Figure 6.14: Analytical vs Simulation Models: User Association Delay of the Distributed Mechanism in comparison to the Centralized One, when using $|C|/|C'|$=4/2 and TTL_m=2 for a Grid-Based WMN with 10×10 Nodes, and varying the Mean Number of WMN Users.

6.5 Overall Architectural Evaluation

In this Section, we resort to the simulation framework presented in Section 3.4 to evaluate the data and control planes of the multi-VN architecture.

First, we evaluate the signaling control delay of the mechanism to associate an user to a fitting VN, signaling control overhead, VN reconfiguration costs and overall user perceived QoE. These results are presented for different WMN scenarios, where a different number of users can access the WMN. They are also compared against the ones obtained with a centralized and two other distributed approaches for VN control and management (in Section 6.5.1, we will detail these three approaches).

In the following, we measure the impact of varying the TTL_m and C' parameters on the behavior of the proposed distributed framework for VN control and management. The variation of these parameters will respectively influence the amount of control information disseminated in the WMN and the organization of the control ring structure; we will then analyze several performance aspects of the overall architectural functionality for each distinct combination of TTL_m and C'. These performance aspects span from the probability to trigger each VN control process in the WMN, signaling control delay of user association and ring maintenance, signaling control overhead, overall user perceived QoE, mean number and size of VNs in the WMN, VN reconfiguration costs and complexity of a global VN discovery performed within the ring structure.

Finally, we analyze the impact of varying the context- and resource-aware weights (w_1 and w_2) that are part of the WMN path selection metric defined in expression (6.2), on the overall QoE perceived by a VN user and on several QoS metrics associated to the user flows.

6.5.1 Evaluation Scenario: Details & Variables

In the following, we define all the parameters and variables of the multi-VN architecture implemented in NS-2, detailing the four distinct VN control and management approaches that are evaluated in this study (this information is summarized in Table 6.2).

We create a grid-based WMN topology with $|N| = 8 \times 8$ nodes, where each node has a

Table 6.2: Simulation Details: Evaluation of the Architectural Data & Control Planes.

Parameter	**Value**
$\|N\|$	64 (8×8 Grid WMN)
WMN Node Interface Range	$100m$ ($800m \times 800m$ of Simulated Area)
WMN Node Interface Capacity	$54Mb/s$ ($4Mb/s$ is for Control Purposes)
WMN Node Buffer Size	500
n_u	Randomly chosen from the set N
$/U$	$\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55\}$
User Connectivity Time	Exponential Distribution with Mean of $1/\mu = 1/20$
User Arrival Rate	Poisson Distribution with Mean Inter-Arrival Time of $1/(\overline{U} \times \mu)$
C	$\{c_1, c_2, c_3, c_4\}$
C'	$\{c_1\}$, $\{c_1, c_2\}$, $\{c_1, c_2, c_3\}$
M_c ($c \in C$)	$\{1, 2, 3, 4\}$
c_1=User Security Requirements	Security Header of VN Packets (EH): $\{0, 22, 34, 54\}Bytes$
	Extra E2E delay of VN flows (ED): $\{0, 5, 10, 15\}ms$
c_2=User Price Preferences	MAC priority level assigned to a VN (P): $\{0, 1, 2, 3\}$
c_3=User Energy Constraints	$\zeta = 5$ & $Y_{min} = \{0.9, 0.8, 0.7, 0.6\}$ (see expression (5.7))
c_4=User Requested Service	{Audio$_1$, Audio$_2$, Video and Data Transfer}
$r_{u_c} \in R_u$ ($c \in C$)	Randomly chosen from the set M_c
$t_{u_c} \in T_u$ ($c \in C'$)	Randomly chosen from the set $\{0, 1\}$
$t_{u_c} \in T_u$ ($c \in \neg C'$)	Randomly chosen from the set $\{1, 2\}$
Changes in $r_{u_c} \in R_u$ ($c \in C$)	Randomly chosen from the set $\{0, 1, ..., t_{u_c}\}$
TTL_m	$\{1, 2, 3\}$
(w_1, w_2)	{(0.0,1.0),(0.2,0.8),(0.4,0.6),(0.5,0.5),(0.6,0.4),(0.8,0.2),(1.0,0.0)}
Periodicity of Control Processes	$500ms$
Global WMN Manager	Central WMN Node
Central WMN Controller	Central WMN Node
Simulation Time/Runs	$1000s/20$

transmission radius of $100m$ ($800m \times 800m$ of simulated area). Each WMN node has one physical interface of $54Mb/s$ and buffer size of 500 packets.

According to the discussion of Section 3.4.3, in which we presented the details of the multi-VN architecture implemented in NS-2, the network virtualization is emulated by dynamically assigning a distinct number of "virtual" interfaces to each WMN node according to the number of VNs that it supports. Then, the capacity and buffer size of each physical WMN node are shared among its VN nodes (or "virtual" interfaces) according to the resource needs of such VN nodes. Moreover, the interface assigned to a VN uses its dedicated wireless channel. Finally, one "virtual" interface of $4Mb/s$ is assigned to all WMN nodes for control purposes, and the AODV routing protocol [36] is used for both data and control.

The arrival rate of users at the WMN is a Poisson process with mean inter-arrival time of $1/\lambda$. Each user $u \in U$ arrives at a random WMN node $n_u \in N$, and he/she is connected for a time following an exponential distribution with mean $1/\mu = 1/20$. From the queueing theory, the mean number of users in the WMN is then defined by $\overline{U} = \lambda/\mu = \lambda/20$; thus, we configure λ to perform tests for a mean number $\overline{U} = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55\}$ of users accessing the WMN. Finally, in the simulation environment, the user communication target is randomly chosen among the nodes from the VN, in which he/she is connected to.

We consider the set of context parameters $C = \{c_1, c_2, c_3, c_4\} = \{$*Security Requirements*, *Price Preferences*, *Energy Constraints*, *Service Type*$\}$ to drive the architecture. Moreover, we vary the number of context parameters that compose the set C', which contains the context parameters that do not frequently admit variations from the user side (low t_{u_c}). Therefore, we consider $C' = \{c_1\}$, $C' = \{c_1, c_2\}$ or $C' = \{c_1, c_2, c_3\}$. The set of levels of each context parameter $c \in C$ is $M_c = \{1, 2, 3, 4\}$

Each user $u \in U$ is characterized by particular context requirements (R_u and T_u): *(i)* $r_{u_c} \in R_u$ can take any value from the set M_c ($c \in C$); *(ii)* $t_{u_c} \in T_u$ ($c \in C'$) can take any value from the set $\{0, 1\}$, since these are the parameters that admit low variation ranges for their demands during the user connectivity time; *(iii)* $t_{u_c} \in T_u$ ($c \in \neg C'$) can take any value from the set $\{1, 2\}$, since these are the parameters that admit more distinct levels during the user connectivity time. Moreover, to model context dynamics during the connectivity time of the user $u \in U$, we consider that the number of changes associated to each context requirement $r_{u_c} {\in} R_u$ can take any value from the set $\{0, 1, ..., t_{u_c}\}$ ($c \in C$). According to this approach, the requirements that admit high variation ranges (the ones from the set C') are the ones characterized by a higher mean number of changes during the user connectivity time. Finally, the changes in the context requirements of user $u \in U$ are spaced by at least $3s$.

Based on the discussion of Section 5.2.1, we implement rules to model the user context requirements in levels, and to map such levels in proper VN features. These rules are described in the following four paragraphs.

We consider 4 security requirements that are mapped in 4 IPSec-compliant VN security levels characterized by distinct authentication and/or encryption mechanisms: {*No_Security*, *DES*, *3DES+SHA*, *3DES+SHA+IPv4_Tunnel*}. Following the strategy presented in Section 5.2.1.2, we emulate a VN security level by configuring the VN packets with an extra security header (EH, as defined in expression (5.5)) and the VN communications with an extra E2E delay (ED, as defined in expression (5.6)). The values assigned to the EH and ED parameters for each VN security level are described in Table 6.2 (please refer to [299]).

Based on the discussion of Section 5.2.1.4, we assign a distinct MAC priority level (P, as defined in expression (5.8)) to the WMN channel assigned to a VN, according to the price preferences of the users connected to such VN ($P{=}3$ is the highest MAC priority level).

To model the impact of a specific VN energy-efficiency level on the decrease of the QoE perceived by an user connected to such VN (Y), we consider that the steepness parameter of the Y decay is $\zeta{=}5$. For each one of the 4 VN energy-efficiency levels (ordered in an ascendant way), Y cannot be lower than a pre-defined value $Y_{min} = \{0.9, 0.8, 0.7, 0.6\}$ (please remember the expression (5.7) defined in Section 5.2.1.3).

We consider two audio services (audio$_1$ and audio$_2$), video and data transfer services as the 4 types of services requested by users, in order to differentiate the flows running at each VN. The first audio service is configured to be transmitted at $12.8Kb/s$ with a payload size of $64Bytes$, and the second one at $64Kb/s$ with a payload size of $160Bytes$. Video service uses the H.264 codec sampled at 30 frames per GoP, 10 of which are P-frames (the used video traces may be found in [303]). Data transfer service is modeled to return a maximum MOS for $450Kb/s$ and a minimum MOS for $10Kb/s$.

In the proposed framework, each WMN node can select any of its supported VNs to be updated, according to the metric defined in expression (5.10). Moreover, each WMN node periodically announces its stored information tuples to its 1-hop physical neighbors ($T_1{=}500ms$ is the periodicity of such process), and these tuples are then disseminated along a distinct maximum number of hops $TTL_m{=}\{1, 2, 3\}$. With the information of these tuples, each WMN node can also select any VN to be locally extended up to it, according to the metric defined in expression (6.2). To evaluate the impact of the weights of such metric, we perform distinct tests for the combinations: (w_1, w_2)={(0,1),(0.2,0.8),(0.4,0.6),(0.5,0.5),(0.6,0.4),(0.8,0.2),(1,0)}. In addition, we implement the distributed control ring, presented in Section 6.2.3, to perform global VN discoveries, being such ring composed by the controllers of the VNs available in the WMN. Each VN controller periodically starts a process to update the QoE-aware related information with the virtual nodes from such VN (T_2=500ms is the periodicity of such process), and to evaluate the need to update the VN controller location. At the end of such process, the actual VN controller contacts its 1-hop logical neighbors, in order to update the routes to maintain the ring connectivity. Further, the global WMN manager is the central WMN node, and it can decide to trigger the creation of a new VN in the WMN, if the WMN node that triggered a global VN discovery in the WMN did not receive any reply in the timeout of 500ms. To evaluate the possibility to create a new VN, the global WMN manager starts a process performed through a set $|L|$=4 of WMN paths, as described in Section 6.3.4. Finally, the timeout to trigger the remotion of VNs when there are no attached users is T_5=500ms.

Concerning the centralized approach to control and manage VNs, the central WMN node is selected as the central WMN controller ($O_c \in N$). O_c periodically contacts each WMN node to refresh its VN knowledge (500ms is the periodicity of such process) and, as a reply, each WMN node sends the context-aware information of the virtual nodes that it supports (please note that these replies do not include the WMN resource-aware information defined in expression (6.1)). In this centralized control approach, when the user $u \in U$ arrives at the node $n_u \in N$, node n_u contacts O_c, which can then trigger a: *(i)* VN update, if node n_u supports any fitting VN for user u (this process uses the metric defined in expression (5.10)); *(ii)* local VN extension, if any fitting VN for user u is available in any of the neighbors of node n_u located at a maximum distance of TTL_m hops; *(iii)* global VN extension, if any fitting VN for user u is available in another part of the WMN (these two VN extension processes use the metric defined in expression (6.2)); *(iv)* VN creation, if the WMN does not provide support for any fitting VN for user u (this process is described in Section 6.3.4).

Focusing on the two other distributed approaches for VN control and management, they mainly differ from the one proposed in Sections 6.2 and 6.3, since they consider different VN discovery mechanisms. More specifically, when the user $u \in U$ arrives at the node $n_u \in N$, and node u does not support any fitting VN for user u (and it cannot trigger a VN update), these approaches adopt distinct mechanisms to discover a point of attachment of any fitting VN for user u in the WMN. In the first approach, node n_u uses a similar strategy than the AODV routing protocol [36]. In this AODV-based VN discovery mechanism, node u broadcasts a VN discovery request in the WMN. The VN discovery request includes the user context; therefore, any WMN node, after receiving the request, is able to check if it supports any fitting VN for user u. If it is true, such WMN node sends a VN discovery reply to node n_u, in a unicast manner. In the second approach, node n_u uses the Bamboo overlay [285], where a set of Bamboo keys are representative of the VNs available in the WMN, and they are stored by one virtual node from such VNs. When a point of attachment of any fitting VN for user u receives a Bamboo-based VN discovery request, it sends a VN discovery reply to node n_u. The VN discovery replies of both the AODV- and Bamboo-based strategies include the context features of the discovered VN, as well as the levels of resources available (e.g., bandwidth) in the traversed path up to node n_u (which was already defined by RA_v in expression (6.1)). From the obtained replies, node n_u selects the VN to be assigned to user u, and the WMN path to perform a local or global VN extension up to node n_u, according to the metric defined in expression (6.2). In these two distributed control approaches, if node n_u did not receive any VN discovery reply in the timeout of 500ms, it contacts the global WMN manager (which is the central WMN node). The global WMN manager can then detect if any fitting VN for user u is available in the WMN, or it can trigger the creation of a new VN in the WMN (according to the process described in Section 6.3.4). In order to update the information stored by the global WMN manager, it periodically contacts each WMN node ($500ms$ is the periodicity of such process) and, as a reply, each WMN node sends the context-aware information of its supported VNs.

Finally, each simulation run lasts $1000s$, and each obtained value is the mean of 20 runs with a confidence degree of 90%.

6.5.2 Comparison of Approaches for VN Control & Management

According to the simulation setup defined above, this Section compares several performance aspects of the proposed distributed VN control and management approach, against the centralized and the AODV- and Bamboo-based distributed approaches. This Section first presents the obtained results, and then, it provides an overall discussion of these results.

6.5.2.1 Results

In Fig. 6.15, we show the mean signaling delay and the mean number of reconfigured WMN nodes associated to the mechanism to update, extend or create a VN to assign to an user that arrives at the WMN (a), or to adapt the VN assigned to an user due to any change in the user context requirements (b). These results are obtained with the four approaches for VN control and management implemented in NS-2, when using (w_1, w_2)=$(0.5, 0.5)$, TTL_m=2 and $|C'|$=2, and varying the mean number of users $\overline{U}$ that access the WMN. The signaling delays do not contemplate the processing delays associated to the WMN nodes, in which it is required to create new virtual nodes to extend or create VNs, since we are not able to measure them in NS-2.

For the same WMN scenario, Fig. 6.16, Fig. 6.17 and Fig. 6.18, respectively present the mean number of virtual nodes per VN, signaling overhead and E2E user perceived QoE associated to each approach for VN control and management. The signaling overhead reflects the total amount of traffic introduced by each approach for VN control and management in relation to the total duration of the simulation, and the QoE perceived by a VN user is obtained according to the formula described in expression (5.9).

6.5.2.2 Discussion

From the results obtained in Fig. 6.15, we observe the improvement of the proposed distributed VN control and management approach over the other three approaches, in terms of the time required to connect an user to a fitting VN. Moreover, the proposed approach requires the reconfiguration of a smaller number of WMN nodes to adapt the VN topologies on-demand, than the AODV- and Bamboo-based distributed approaches.

Focusing on the centralized approach for VN control and management, the user association delays are higher than the ones obtained with the proposed approach, since it is always required to contact the central WMN controller, even to perform the update or the local extension of any VN available in the user neighborhood. In the distributed approach, these two VN control processes can be quickly triggered by the WMN node where the user is connected to, by inspecting its stored VN information tuples. With respect to the number of WMN nodes reconfigured to perform the adaptation of a VN to assign to an user, our approach

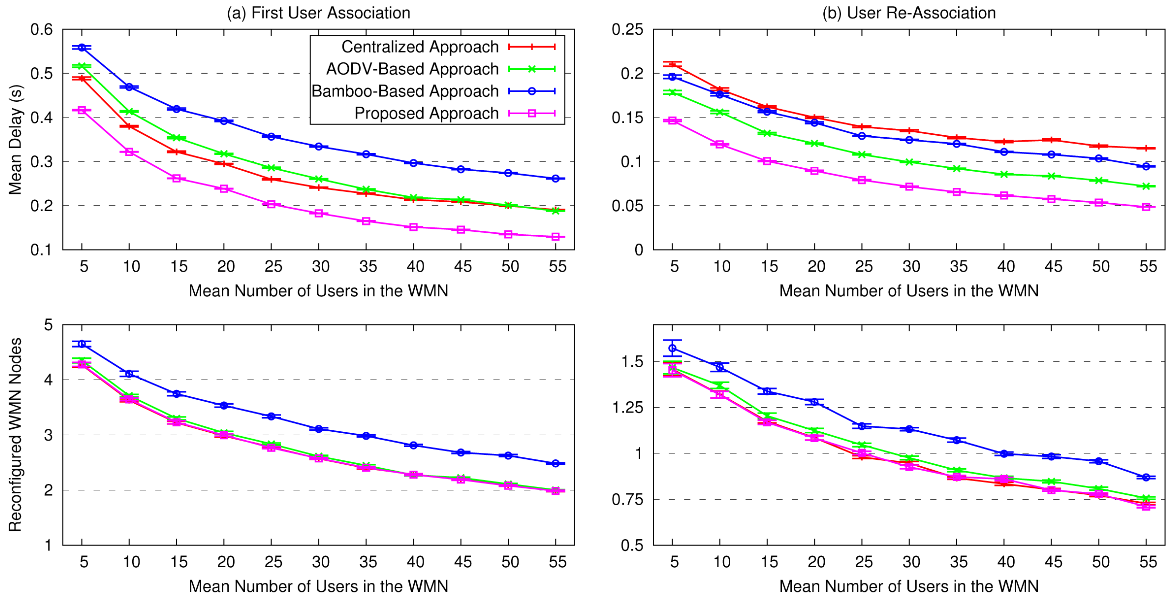


Figure 6.15: Mean Signaling Delay and Mean Number of Reconfigured WMN Nodes, in order to connect an User to a Fitting VN (a) or to adapt the VN assigned to the User (b) in each VN Control Approach, when using (w_1, w_2)=$(0.5, 0.5)$, $|C'|$=2 and TTL_m=2 for a Grid-Based WMN with 8×8 Nodes, and varying the Mean Number of Users in the WMN.

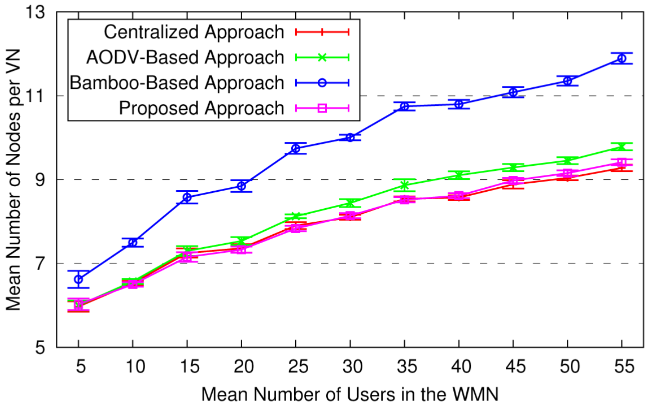


Figure 6.16: Mean Number of Virtual Nodes per VN in each VN Control Approach, when using (w_1, w_2)=$(0.5, 0.5)$, $|C'|$=2 and TTL_m=2 for a Grid-Based WMN with 8×8 Nodes, and varying the Mean Number of Users in the WMN.

presents very similar results than the ones obtained with the centralized approach. However, these results are slightly better for the centralized approach, since the central WMN controller has the complete knowledge of the WMN, so that it can trigger the VN adaptations through the shortest WMN paths (this fact is also strengthened in Fig. 6.16, since the centralized approach presents the lowest mean number of virtual nodes per VN).

In the AODV-based approach, the user association delays are higher than in the proposed

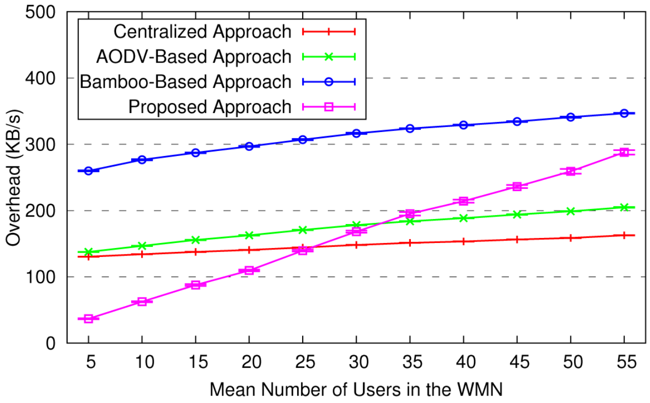


Figure 6.17: Mean Signaling Overhead associated to each VN Control Approach, when using (w_1, w_2)=$(0.5, 0.5)$, $|C'|$=2 and TTL_m=2 for a Grid-Based WMN with 8×8 Nodes, and varying the Mean Number of Users in the WMN.

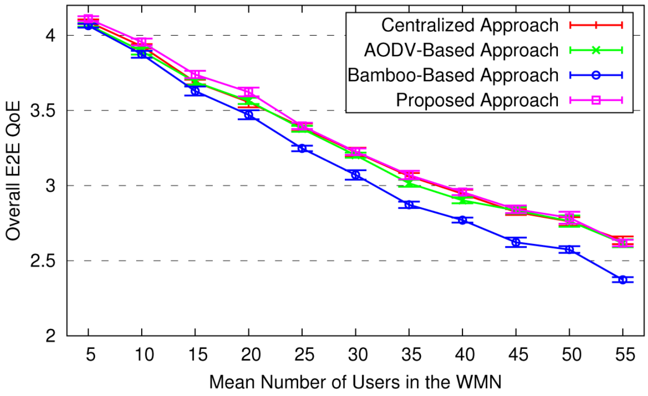


Figure 6.18: Mean E2E QoE perceived by a VN User in each VN Control Approach, when using (w_1, w_2)=$(0.5, 0.5)$, $|C'|$=2 and TTL_m=2 for a Grid-Based WMN with 8×8 Nodes, and varying the Mean Number of Users in the WMN.

approach, due to the broadcast nature of the AODV-based VN discovery mechanism, which can lead to a higher number of collisions, and so, higher delays. This fact can also require the reconfiguration of a higher number of WMN nodes to perform the extension of a VN up to the user location, since the AODV-based VN discovery reply cannot be received through the shortest path between the user location and the node containing the fitting VN (this fact is also shown in Fig. 6.16, where the AODV-based approach presents a higher mean number of nodes per VN than the proposed approach).

When using the Bamboo overlay for VN discovery, it is required to perform a high number of communications to find a fitting VN to assign to an user, and so, the Bamboo-based approach has the highest user association delays. Moreover, due to the random distribution of the Bamboo keys in the WMN, the point of attachment of a fitting VN for the user cannot be discovered near the user location, which can lead to the reconfiguration of a higher number of WMN nodes to extend the discovered VN up to the user location (this fact is also observed in Fig. 6.16, since the Bamboo-based approach is characterized by the highest mean number of virtual nodes per VN).

The user association delays and number of reconfigured WMN nodes decrease with the increase of the mean number of users accessing the WMN, $\overline{U}$. With the increase of $\overline{U}$, there are more VNs available in the WMN, and so, there is a higher probability to find a fitting VN in any node near the user location. This leads to the existence of more VN updates or local VN extensions, that are less complex than global VN extensions. In addition, with the increase of $\overline{U}$, the decreasing tendency of the user association costs (mainly the signaling delays) is lower, since a higher number of processes are simultaneously triggered in the WMN, leading to more collisions.

Still from Fig. 6.15, we conclude that the VN control and management approaches have lower user association delays and number of reconfigured WMN nodes in the adaptation of a VN to be assigned to an user that suddenly changes his/her context requirements (R_u), than in the adaptation of a VN to be assigned to a new user that arrives at the WMN. In the first case, due to the user context flexibility during the connectivity time, the user can remain connected in the same VN after changing his/her context requirements, without the need to discover and adapt another VN in the WMN. Moreover, in the centralized approach, the central WMN controller needs to be notified about any user context change. Therefore, the centralized approach has the highest signaling delays, when required to cope with any user context change. Further, the proposed approach has the lowest signaling delays, when required to deal with any user context change. This is because of the context-aware organization of the ring structure, in which we enable the possibility of the controllers of the VNs, that are most probable to deal with the most frequent user context changes, to be closely located in the ring structure. This leads to the decrease of the time spent in the global discovery of another VN to assign to the user.

Concerning the results presented in Fig. 6.17, we observe that the signaling overhead increases with the increase of the mean number of users accessing the WMN. With more users, it is required to perform more control processes to discover and adapt VNs to assign to users, and the mean number of VNs and virtual nodes per VN increase (as can be seen in Fig. 6.16). This increases the load of the messages required to update the VN knowledge in the architectural entities with control responsibilities.

Moreover, when there is a low mean number of users accessing the WMN ($\overline{U} < 30$), the signaling overhead of the proposed approach is lower than the one of the other three approaches. In these cases, the signaling overhead required to maintain the connectivity of the ring structure is small, since there is a low mean number of VNs available in the WMN (please remember that the ring interconnects the controllers of the VNs available in the WMN). However, with the increase of $\overline{U}$, there are more VNs available in the WMN, and so, the signaling overhead of the proposed approach is higher than the one of the centralized and the AODV-based approaches, due to the higher amount of control traffic required to maintain the ring structure.

From the results of Fig. 6.17 for $\overline{U} \geq 30$, we can see that the centralized approach requires the lowest signaling overhead, since it is only required to contact the central WMN controller to perform every VN control process. Moreover, the AODV-based approach has a higher overhead than the centralized approach, mainly due to the network flooding introduced by the broadcast nature of the AODV-based VN discovery process. However, this overhead is lower than the one of the proposed approach, since it is not required to update a control overlay in the AODV-based approach. The signaling overhead of the Bamboo-based approach is the highest, mainly due to the higher amount of maintenance traffic of the Bamboo overlay.

Finally, Fig. 6.18 shows that the mean perceived QoE by a VN user is very similar for every approach, except when opting for the Bamboo-based approach. This is explained by the fact that, when using the Bamboo overlay, the discovered VN to assign to an user has a very high probability to not maximize the result of the VN selection metric defined in expression (5.10) (when considering all the fitting VNs for such user that can be available in the WMN). In the centralized approach, the central WMN controller stores the information of all VNs and VN nodes available in the WMN. However, since it does not store the resource-aware information of WMN links (defined in expression (6.1)), the VN extensions can be performed through congested WMN paths according to the metric defined in expression (6.2). Therefore, when the WMN is not congested ($\overline{U}$<30), the centralized approach presents slightly worse QoE results than the proposed approach, since this last one makes use of resource-aware information to perform VN extensions through non-congested WMN paths (which improves the QoE results). Finally, when opting for the AODV-based approach, the broadcast nature of the AODV-based discovery mechanism can lead to the non-detection of the best fitting VNs to assign to an user (selected according to the metric defined in expression (5.10)). Therefore, we observe a small decrease on the mean perceived QoE by a VN user in this approach.

6.5.3 Evaluation of the Distributed VN Control Framework

In this Section, we analyze several performance aspects of the proposed distributed framework for VN control and management. More specifically, we analyze the impact of the TTL_m parameter that defines the maximum number of hops to disseminate a specific VN information tuple in the WMN (please refer to expression (6.3)). Moreover, we study the effect of the number of context parameters that compose the set C', which contains the context features that do not frequently admit variations from the user side (low t_{u_c}) (please refer to Section 6.2.3.1). In the following, we first present the results obtained in this study, and then, we provide an overall discussion of these results.

6.5.3.1 Results

In Fig. 6.19, we show the probability of each VN control process (VN update, local VN extension, global VN extension or VN creation) to be triggered in the WMN, when an user arrives at a WMN (a) or changes his/her context requirements (b). These results are obtained for a mean number of 20 users accessing the WMN, and using (w_1, w_2)=$(0.5, 0.5)$, the parameters TTL_m=1, TTL_m=2 or TTL_m=3, and the sets C'=$\{c_1\}$, C'=$\{c_1, c_2\}$ or C'=$\{c_1, c_2, c_3\}$.

For the same WMN scenario and variables, Fig. 6.20 shows the mean signaling delay and the mean number of reconfigured WMN nodes associated to the mechanism to update, extend or create a VN to assign to a new user that arrives at the WMN (a), or to react to any user context change (b).

In the following, Fig. 6.21, Fig. 6.22, Fig. 6.23 and Fig. 6.24 respectively present the mean number of VNs, number of virtual nodes per VN, signaling overhead and E2E user perceived QoE obtained for distinct combinations of the parameter TTL_m and the set C'.

Concerning the distributed ring structure proposed in Section 6.2.3, Fig. 6.25 shows the mean number of SCs that compose the ring (a), VNs that are part of each SC (b) and contacted SCs in a global VN discovery (c). These results are again obtained for a mean number of 20 users accessing the WMN, and using (w_1, w_2)=$(0.5, 0.5)$ for distinct combinations of

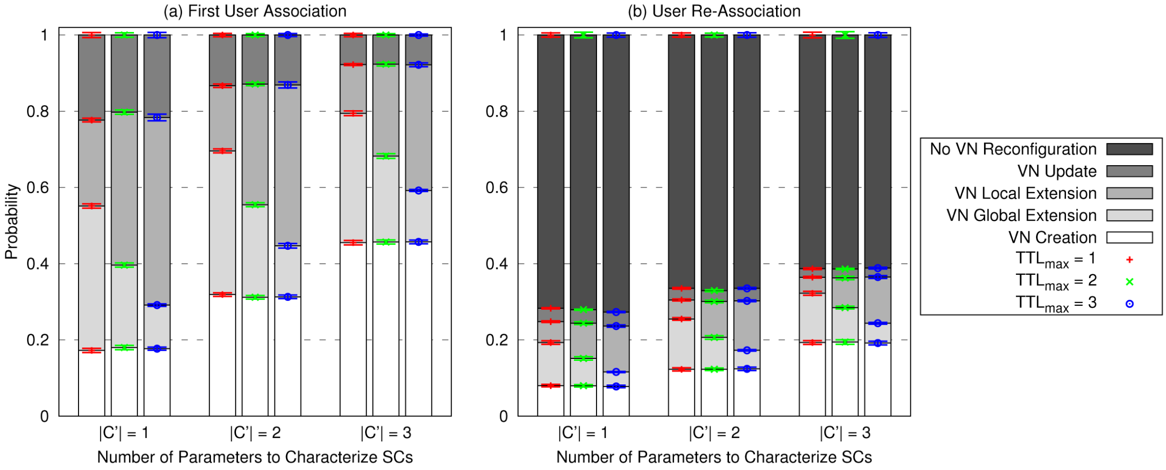


Figure 6.19: Probability of each VN Control Process to be triggered in the WMN, in order to connect an User to a Fitting VN (a) or to adapt the VN assigned to the User (b), when applying the Distributed VN Control Framework with (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

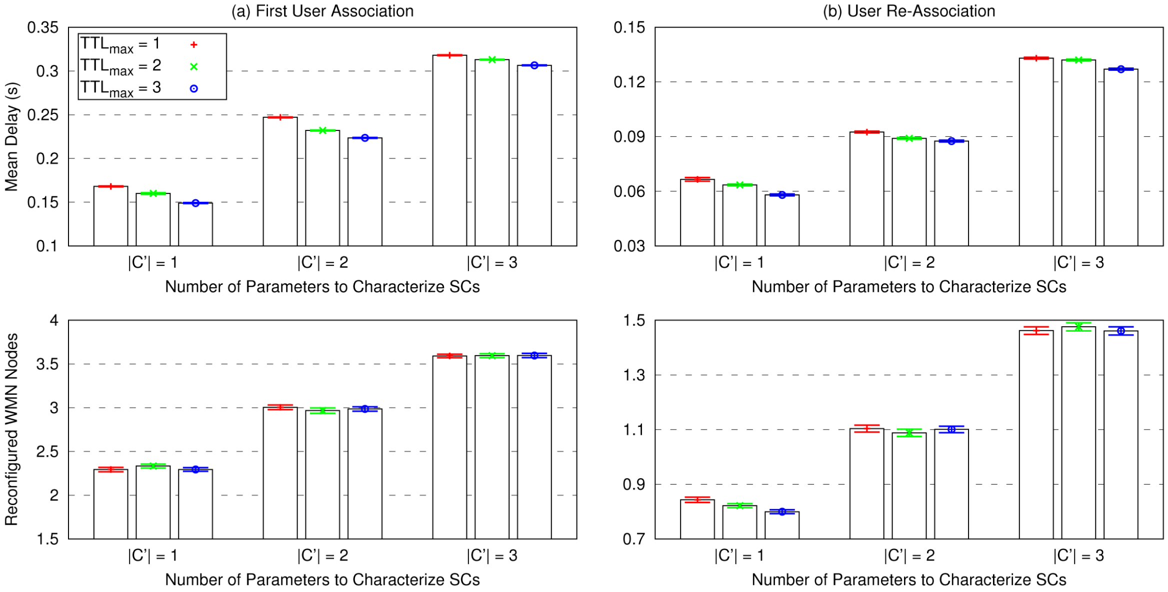


Figure 6.20: Mean Signaling Delay and Mean Number of Reconfigured WMN Nodes, in order to connect an User to a Fitting VN (a) or to adapt the VN assigned to the User (b), when applying the Distributed VN Control Framework with (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

TTL_m and of the number of context parameters that compose the set C'.

For the same WMN scenario and variables, Fig. 6.26 presents the mean time elapsed between two consecutive updates of the location of the controller of a VN. Finally, Fig. 6.27

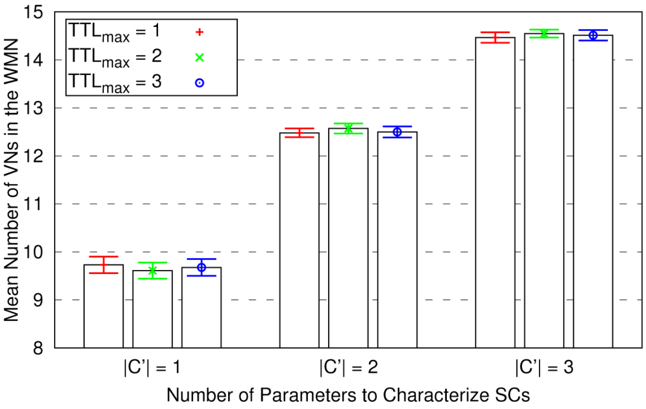


Figure 6.21: Mean Number of VNs in the WMN, when applying the Distributed VN Control Framework with (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

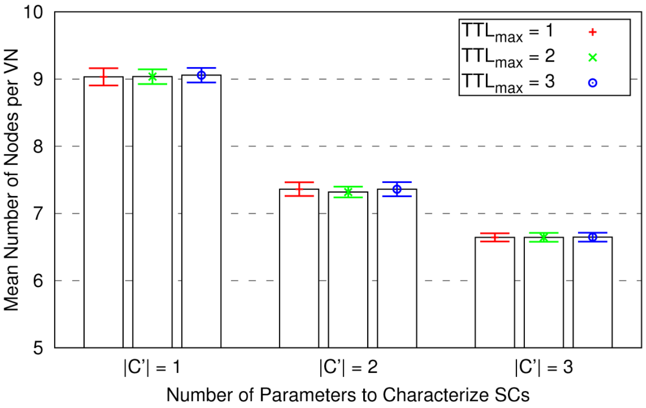


Figure 6.22: Mean Number of Virtual Nodes per VN, when applying the Distributed VN Control Framework with (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

shows the mean signaling delay of the control processes required to: *(i)* update the location of a specific VN controller inside such VN (detailed in Fig. 6.3); *(ii)* insert a new VN controller in the ring structure, when such VN is created in the WMN (detailed in Fig. 6.2); *(iii)* update the location of a VN controller with its 1-hop logical neighbors in the ring structure (detailed in Fig. 6.4); *(iv)* adapt the ring structure to the leaving of a VN controller, in case of the complete remotion of such VN (detailed in Fig. 6.5).

6.5.3.2 Discussion

From the results obtained in Fig. 6.19, we observe that the increase of the number of context features that compose the set C' ($|C'|$) leads to the increase of the probability to create a new VN in the WMN (and to the consequent decrease of the probability to update or extend any VN available in the WMN). This occurs because the increase of $|C'|$ is reflected

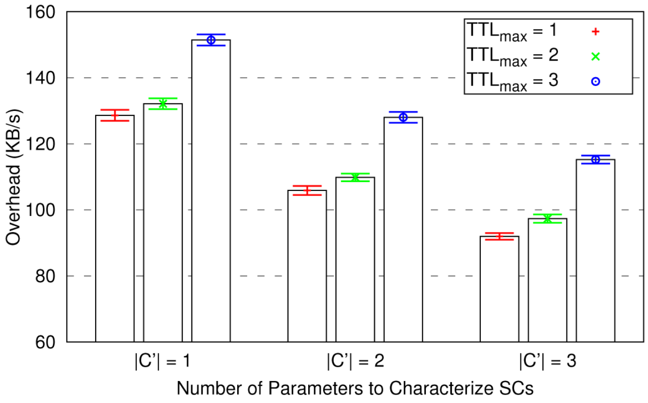


Figure 6.23: Mean Signaling Overhead associated to the Distributed VN Control Approach with (w_1, w_2)=(0.5, 0.5), when varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

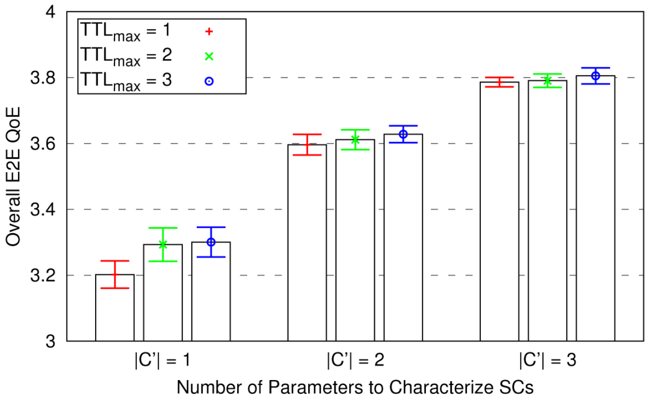


Figure 6.24: Mean E2E QoE perceived by a VN User, when applying the Distributed VN Control Framework with (w_1, w_2)=(0.5, 0.5), and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

in the decrease of the overall context-aware flexibility of users to be connected to VNs available in the WMN (the set C' contains the context features that are less flexible from the user side, or the ones that admit a small number of context levels provided by the VNs available in the WMN). Therefore, for a high $|C'|$, it is frequently required to create a new VN to assign to an user (instead of starting any VN update or VN extension), since the majority of the VNs available in the WMN do not fit the context requirements of such user (this fact is also observed in Fig. 6.21, where the mean number of VNs available in the WMN is the highest for $|C'|$=3). As can be seen in Fig. 6.20, the increase of $|C'|$ also leads to the increase of the signaling delays and number of reconfigured WMN nodes required to connect an user to a fitting VN. This is explained by the fact that these two metrics have associated higher values for a VN creation than for a VN update or local VN extension, and a VN creation has a higher probability to be performed for high values of $|C'|$.

Considering the same $|C'|$, we can see that the increase of the maximum number of hops

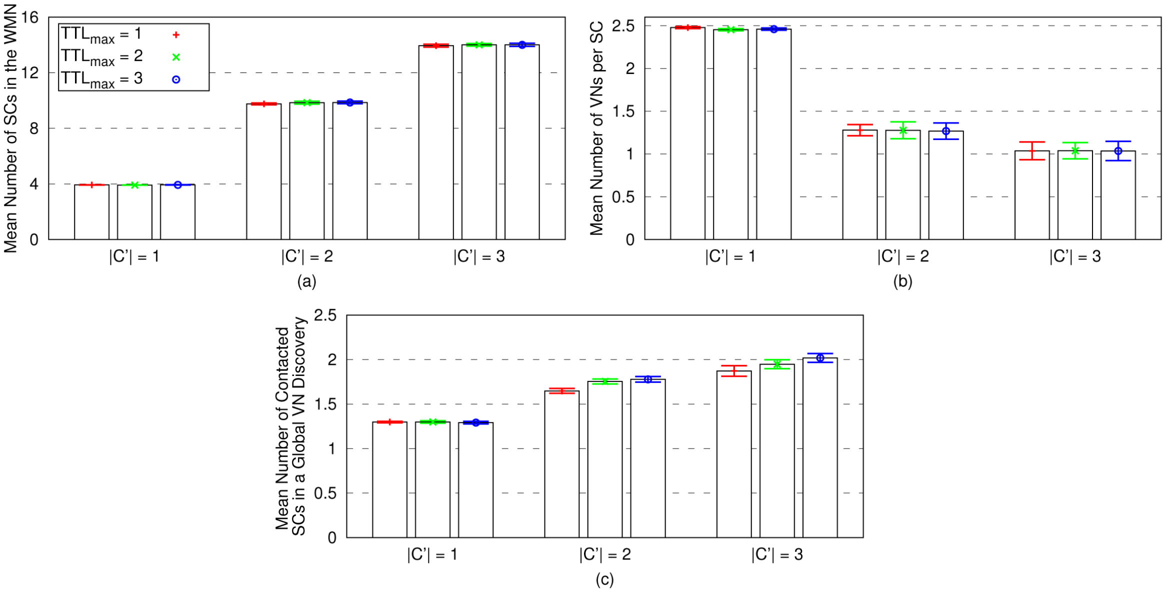


Figure 6.25: Mean Number of SCs (a), VNs per SC (b), and Contacted SCs in a Global VN Discovery (c), when applying the Distributed VN Control Framework with (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

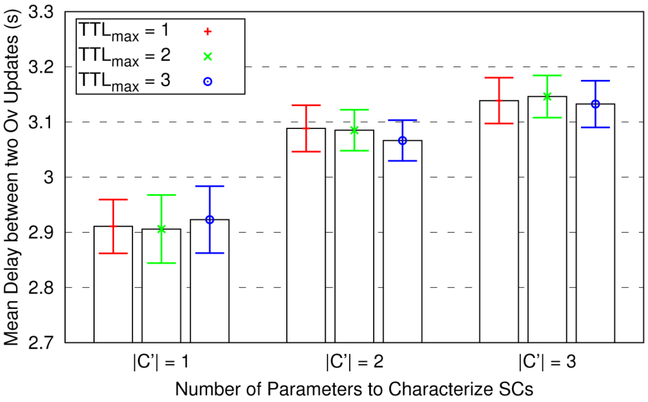


Figure 6.26: Mean Delay between the Update of the VN Controller Location, when using (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

to disseminate a specific VN information tuple in the WMN, TTL_m, leads to the increase of the probability to perform local VN extensions in the WMN (while decreasing the global VN extension probability). This occurs because, for a high TTL_m, the WMN nodes have the possibility to store a higher amount of information of the VNs available in their neighborhood. Moreover, for the same $|C'|$, the increase of TTL_m enables the decrease of the user association delays (please see Fig. 6.20), since the local VN extensions take less time than the global VN

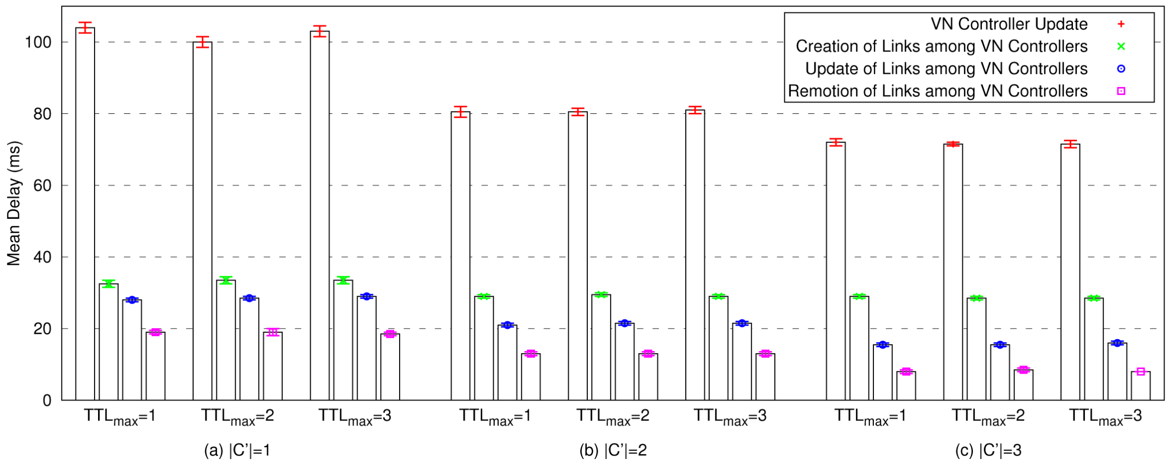


Figure 6.27: Mean Signaling Delay of the Control Processes required to Maintain the Ring Structure, when using (w_1, w_2)=$(0.5, 0.5)$, and varying TTL_m and $|C'|$ for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 20 Users.

extensions. However, the number of reconfigured WMN nodes in a VN adaptation is not highly affected by the variation of TTL_m for the same $|C'|$. Despite the use of TTL_m=3 can lead to the existence of more local VN extensions than the use of TTL_m=1 (please see Fig. 6.19), these local VN extensions can then involve the reconfiguration of three WMN nodes instead of only one WMN node. Further, in Fig. 6.21, we conclude that the variation of TTL_m for the same $|C'|$ does not affect the mean number of VNs available in the WMN, since it does not influence the VN creation probability (please see Fig. 6.19).

In Fig. 6.19, we can also see that there is a high probability of an user changing his/her context requirements to remain connected in the same VN, without the need to discover and adapt another VN in the WMN. This probability is again lower for $|C'|$=3 than for $|C'|$=1 since, in the first case, the overall context-aware flexibility of users is higher. Moreover, the benefits of not reconfiguring any VN are observed in Fig. 6.20. Here, for the same $|C'|$ and TTL_m, the signaling delays and number of reconfigured WMN nodes are lower when it is required to deal with user context changes than to deal with the arrival of new users.

Concerning the results presented in Fig. 6.22, we observe that the mean number of virtual nodes per VN decreases with the increase of $|C'|$. This occurs because the increase of $|C'|$ implies the decrease of the overall context-aware flexibility of users to be connected to VNs available in the WMN, increasing the probability to create new VNs, and, as a consequence, increasing the mean number of VNs available in the WMN (please see Fig. 6.21). Hence, for the same mean number of WMN users ($\overline{U}$=20, in this case), the number of users per VN is lower for $|C'|$=3 than for $|C'|$=1. Subsequently, the number of VN extensions that need to be triggered in a specific VN to allow the connection of more users is lower for $|C'|$=3 than for $|C'|$=1 (this leads to a lower mean VN size for $|C'|$=3 than for $|C'|$=1).

These distinct results for the mean VN size are also reflected in the signaling overhead presented in Fig. 6.23. With the increase of $|C'|$, the mean number of users and virtual nodes per VN decrease. Therefore, it is then required to periodically update a lower amount of knowledge among the virtual nodes from a specific VN and the respective VN controller, which decreases the signaling overhead. Moreover, the increase of TTL_m for the same $|C'|$, despite not increasing the mean VN size (as observed in Fig. 6.22), increases the signaling overhead, since a higher amount of VN information tuples are disseminated in the WMN.

Focusing on the mean perceived QoE by a VN user (please see Fig. 6.24), the increase of TTL_m and $|C'|$ shows a slight improvement of the overall QoE results. First, with the increase of TTL_m, each WMN node has access to a higher amount of VN information tuples, in order to perform local VN extensions that can lead to better QoE results, when applying the VN and WMN path selection metric defined in expression (6.2). Second, with the increase of $|C'|$, an user can be more frequently connected to the VN that exactly fits the user context requirements, which can lead to better QoE results.

We will now focus on several aspects related to the use and maintenance of the distributed ring structure proposed in Section 6.2.3. In Fig. 6.25, we observe that for the same mean number of WMN users ($\overline{U}$=20, in this case), the increase of $|C'|$ leads to the increase of the mean number of SCs that compose the ring structure, which decreases the mean number of VN controllers per SC. This is explained due to the followed strategy to build the ring structure, where each SC groups the controllers of VNs characterized by a specific combination of the levels that can be admitted by the context features of the set C'. Moreover, the increase of $|C'|$ requires the need to inspect more SCs of the ring structure in a global VN discovery, in order to find a fitting VN to assign to an user. This occurs because, a low number of VNs can fit the context requirements of a particular user for a high $|C'|$, due to the low overall context-aware flexibility of such user. Therefore, it is important to travel a high number of SCs in order to possibly find a fitting VN to assign to the user. Still from Fig. 6.25, we observe that for the same $|C'|$, the results do not depend on the variation of TTL_m, since TTL_m does not influence the mechanisms to build the distributed ring structure.

In Fig. 6.26, we observe that the increase of $|C'|$ leads to the increase of the mean time elapsed between two consecutive updates of the location of the controller of a VN (thus decreasing the signaling delays and reconfiguration costs to update the VN controller location, which include the transference of knowledge from the old to the new VN controller). In the one hand, since there is a lower number of VNs per SC for $|C'|$=3 than for $|C'|$=1 (please see Fig. 6.25), the amount of information that influences the metric to minimize the mean hop distance among consecutive VN controllers of the same SC (please see expression (6.6)) is lower for $|C'|$=3 than for $|C'|$=1. On the other hand, as previously observed, the increase of $|C'|$ leads to the decrease of the mean number of virtual nodes per VN, due to the decrease of the number of VN extensions that need to be triggered in a specific VN (please see Figures 6.19 and 6.22). This implies the decrease on the number of changes in the VN topology, and so, the reduction of the need to update the VN controller location, in order to reduce the delays of the control communications performed inside such VN (please see expression (6.5)). Again, for the same $|C'|$, the effects of the variation of TTL_m on the results of Fig. 6.26 are negligible, since TTL_m does not influence the selection of the location of VN controllers.

Finally, the signaling delays to build and maintain the ring structure (presented in Fig. 6.27) also depend on the number of context features that compose the set C' ($|C'|$). However, these delays do not depend on the TTL_m parameter, since it does not influence the selection of the VN controllers, which will then be interconnected to compose the ring structure. First, we already concluded that a high $|C'|$ results in a low mean VN size (please see Fig. 6.22). Therefore, the delay of the process to update the QoE-aware related information with the virtual nodes from such VN takes less time for $|C'|$=3 than for $|C'|$=1 (since, in the first case, the number of virtual nodes per VN is lower). Second, the signaling delays to insert, update or remove a VN controller in the ring structure are slightly lower for $|C'|$=3 than

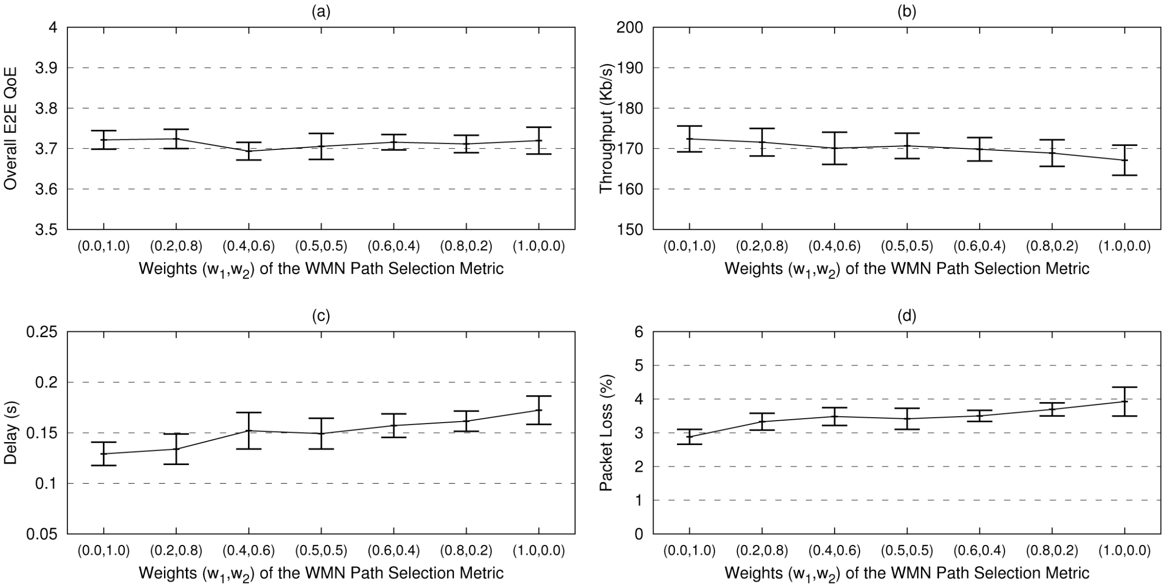


Figure 6.28: Mean E2E QoE perceived by a VN User (a), and Mean Throughput (b), Delay (c) and Packet Loss (d) of the User Flows, when applying the Distributed VN Control Framework with $|C'|$=2 and TTL_m=2, and varying the Weights (w_1, w_2) of the WMN Path Selection Metric for a Grid-Based WMN with 8×8 Nodes and a Mean Number of 15 Users.

for $|C'|$=1. This occurs because, for $|C'|$=3, the number of VNs per SC and virtual nodes per VN is lower than for $|C'|$=1 (as observed in Figures 6.22 and Fig. 6.25), which leads to a lower amount of information that influences the VN controller selection metric defined in expression (6.4). Therefore, the selection of the VN controllers can be performed, so that the mean hop distance among consecutive VN controllers of the same SC is lower for $|C'|$=3 than for $|C'|$=1. This fact leads to the decrease of the signaling delays to create, update and remove the links of the distributed ring structure.

6.5.4 Evaluation of the WMN Path Selection Mechanism

In the following, we analyze the impact of the WMN path selection metric, defined in expression (6.2), on the proposed distributed framework. To this purpose, when configuring the WMN path selection metric, we assign distinct weights for the: *(i)* context-aware difference among the context needs of an user and the context features of a fitting VN (w_1); *(ii)* level of resources available in the possible WMN paths to perform the extension or creation of a fitting VN up to the node where the user is connected to (w_2).

In Fig. 6.28, we present the mean user perceived QoE and several QoS metrics related to the user flows, when considering the set of combinations for the weights (w_1, w_2): $\{(0,1), (0.2, 0.8), (0.4, 0.6), (0.5, 0.5)$,$(0.6, 0.4), (0.8, 0.2), (1, 0)\}$. These results are obtained by using TTL_m=2 and $|C'|$=2, and considering a mean number of 15 users accessing the WMN. In this study, we decrease the mean number of WMN users from 20 to 15, with respect to the study of Section 6.5.3. This is performed, since we need to consider a less congested WMN scenario in order to better evaluate the impact of assigning a low or high weight to the resource-aware metric (in a more congested WMN scenario, the evaluation of this impact will be very difficult, since a high number of the WMN paths could be congested).

From the obtained results, we can see that, when it is assigned a lower value to the weight w_2, the mean throughput of VN communications slightly decrease (Fig. 6.28-b), and the mean delay and packet loss of the same communications slightly increase (Fig. 6.28-c/d). This is expected since the decrease of w_2 leads to the decrease of the importance given to the resource-aware metric (defined in expression (6.1)) in the WMN path selection metric (defined in expression (6.2)). Therefore, when considering a low w_2, a particular VN extension has a higher probability to be performed through a congested WMN path (or a path with a low level of free resources available), than when considering a high w_2. Since the selection of a congested WMN path can be reflected in high interference and collision probability inside the new VN links that need to be created to perform the VN extension, a low w_2 then implies worse results for the QoS metrics than a high w_2.

Nevertheless, despite the direct influence of the QoS metrics in the user perceived QoE, the overall QoE results are not affected by considering a low w_2 (as observed in Fig. 6.28-a). Moreover, when it is given a high importance to the context-aware metric (defined in expression (5.10)) in the WMN path selection metric (defined in expression (6.2)), it can lead to the selection of a VN that better fits the user context requirements.

6.6 Conclusions

In this Chapter, we defined and evaluated an overall distributed and context-aware framework to autonomously control and manage the VNs on-demand. Such framework included the signaling processes and messages to: *(i)* distribute several context-aware knowledge and control functionalities among the distinct architectural entities; *(ii)* allow the cooperation of these entities to dynamically and quickly perform the local or global discovery, selection, creation, adaptation and remotion of VNs, in order to react to the arrival or mobility of users in the WMN, and to the change of the context requirements of users.

In detail, the proposed framework incorporates the following mechanisms. First, WMN nodes can gather and disseminate the WMN and VN information available on it or on its physical neighborhood. This can enable the fast selection and adaptation of VNs to react to user context change or mobility, thus decreasing the delay and reconfiguration costs for VN adaptation. Second, the VN selection metric considers the context flexibility of an user, in order to allow the possibility of an user to be connected to VNs that do not exactly fit all the user context needs, but are characterized by context features that are very similar to the user needs (which decreases the need to create new VNs to serve single users and its associated complexity). This metric also aims to select a VN, from the candidate set of fitting VNs for the user, that enables the reduction of the number of WMN reconfigurations required to adapt the VN assigned to the user, in case of user context changes during the user connectivity time. Third, we consider a WMN path selection metric to perform VN extensions and creations through non-congested WMN paths, instead of only using a shortest path strategy, which can increase the WMN capacity. Fourth, the virtual nodes from a specific VN are envisioned to cooperate in the selection of the VN controller. The VN controller is the virtual node that coordinates the control functionalities performed in the VN, and configures the same context-aware mechanisms and protocols in all the virtual nodes from the VN. Fifth, there is a distributed and context-aware control ring structure, composed by the controllers of the VNs available in the WMN, to allow the fast global discovery of a point of attachment of any fitting VN to assign to an user. The different levels of user context flexibility are considered in the design and organization of the ring structure, enabling fast communications among controllers of VNs that are most probable to deal with user context requirements and their variations. Sixth, the location of the controller of each VN is periodically updated in order to both improve the performance of the control and management functionalities inside the VN, and to avoid topological mismatching problems between the physical WMN infrastructure and the ring structure. Finally, we define the processes and used criteria to dynamically create and remove VNs.

Concerning the evaluation of the proposed distributed framework, this Chapter first defined an analytical model to derive closed-form expressions that measure the signaling effort (or delay) of user association and the control overhead introduced by the distributed approach, as well as by centralized and decentralized approaches for VN control and management. From the obtained analytical results, we observed the lower user association delays of the distributed approach when compared to the centralized and decentralized ones. Moreover, we observed a small increase of the control overhead of the distributed approach over the centralized one (and in some cases, over the decentralized one). Further, we assessed the impact of considering distinct levels for the overall user context flexibility and for the amount of control information disseminated in the WMN, on the delay and overhead performance of the distributed approach, when compared to the centralized and decentralized ones. Finally, the closeness between the analytical and simulation results, for a specific WMN scenario, allowed us to validate the model.

In the following, we performed NS-2 simulations to evaluate the data and control planes of the multi-VN architecture, when compared to centralized and AODV- and Bamboo-based distributed approaches. From the obtained results, we observed the improvement of the proposed approach over the three other approaches, in terms of the time required to connect an user to a fitting VN. These gains are strongly observed (especially, with respect to the centralized approach) when it is required to adapt the VN assigned to an user, due to user context changes. Moreover, the proposed approach presents very similar results than the centralized one for the mean number of reconfigured WMN nodes to adapt the VN topologies on-demand and for the user perceived QoE. Concerning these two metrics, the proposed approach provides better results than the AODV- and Bamboo-based approaches, which perform global VN discoveries by flooding the WMN or using a Bamboo-based control overlay. However, for a high mean number of WMN users, the proposed approach introduces a higher control overhead than the centralized and AODV-based approaches. To better optimize the performance of the proposed approach, we also performed simulations that consider distinct levels for the: *(i)* overall user context flexibility; *(ii)* amount of information disseminated in the WMN; *(iii)* weights of the VN and WMN path selection metrics.

From the work of this Chapter, we can conclude that a distributed approach can improve the control and management of the proposed multi-VN architecture driven by user-centric context information. A distributed approach for VN control and management, beyond enabling the load-balancing of control functionalities along the network elements, also allows the fast architectural adaptation to user and WMN context dynamics, at the cost of a slightly higher control overhead and synchronization costs.

Chapter 7

Conclusion

This Chapter summarizes the impact and usefulness of the most important contributions of this Thesis. Moreover, it provides a look into important lines of research that could directly complement or evolve the context-aware multi-VN architecture presented in this Thesis, which can be a good starting point towards its integration in emerging and future network architectures.

7.1 Results & Achievements

Throughout the different Chapters of this Thesis, we presented several complementary policies, mechanisms and frameworks that enable a flexible and adaptive context-aware configuration and control of WMNs through network virtualization.

We started by providing a high level overview of a WMN architecture that considers the information related to user context as the main driver for its design, and network virtualization as the key enabling technique for its implementation. Bringing together these two concepts, we split a physical WMN infrastructure in a set of usable, scalable and adaptive VNs, being each VN specialized to meet a particular set of user context requirements.

However, this user-centric architecture raises several challenges in order to deal with the user context dynamics in highly mobile WMN environments, especially when the VN nodes and protocols need to be personalized and adapted on-demand to meet the ongoing context requirements of a particular user (or group of users). These challenges led to the definition and evaluation of several solutions that cover distinct aspects related to the dynamic control and management of virtual nodes and networks in WMN environments, with particular emphasis on context-aware and distributed solutions.

In the following, we highlight the several contributions of this Thesis that were described between Chapters 3 and 6. These solutions were proposed and evaluated to meet the research goals presented in Chapter 1, and the work of a particular Chapter was performed to enhance or complement the work of the previous Chapters.

7.1.1 Context-Aware Architecture for WMNs

Completely aligned with the research towards the future Internet, a substantial contribution of this Thesis resides in the definition of a WMN architecture, presented in Section 3.2. This architecture is driven by the particular expectations and preferences of mesh clients and the needs of their devices and services, which we denominate by user context. By taking advantage of the potentials of network virtualization, the main goal of the proposed architecture is to allow the splitting of a physical WMN infrastructure in a set of context-aware VNs. Each VN can be instantiated by a specific VN provider or network operator, and it is created, configured and adapted on-demand to meet a variable set of user context requirements. Users are grouped according to similarity on their context needs, and assigned to VNs matching these needs. Besides the clarification of the goals and features of the architecture, Section 3.2 also presented the key: *(i)* control and management challenges raised by the architecture; *(ii)* architectural entities, along with their functionalities and interactions.

Thus, this first contribution is a good starting point towards the definition, deployment and evaluation of more general user-centric architectures that can be part of future Internet environments. These architectures can enable a better convergence of a high variety of users, devices, services and operators in the same Internet-based network infrastructure.

7.1.2 Impact of Network Virtualization

In order to prevent the work performed in this Thesis from becoming exceedingly conceptual and loosing its potential impact, we defined and evaluated analytical models to better understand the concepts involved in the proposed architecture, contributing to the clarification of the effectiveness and usefulness of such concepts. We also implemented a NS-2 simulation platform to test several architectural mechanisms proposed in this Thesis. The key implementation details of such platform are presented in Section 3.4.

Following this strategy, Section 3.3 presented a model that covered several aspects of the proposed architecture. Such model was able to capture the mean E2E delay of a data communication performed within a specific context-aware VN, supported by a specific set of WMN nodes. In Section 3.5, we made use of such model to study the impact of network virtualization on our approach, being the results of such model validated against a NS-2 simulation setup, due to the closeness between the analytical and simulation results.

The results showed that network virtualization enables the flexible context-aware configuration of distinct VNs that cohabit in the same physical WMN infrastructure, and it also introduces enough isolation to meet distinct user communication requirements with reasonable delays. Moreover, network virtualization greatly reduces the interference among distinct competing communication flows, especially when the WMN starts to be overloaded, which results in lower delays and higher throughput for such flows when compared to a traditional WMN environment.

7.1.3 Context-Aware Network Control & Management

Given that we make use of an user-centric model of context information to drive the behavior and functionality of a multi-VN architecture for WMNs, it is important to define the mechanisms to deal with the user context information. These mechanisms can span from context modeling, mapping, discovery and selection schemes. Moreover, we need to measure how the user is perceiving the requested services, and possibly adapt several architectural processes to better meet the user expectations. Along this Thesis, we defined and evaluated mechanisms that cover several of these context-aware aspects.

Concerning the context modeling, Section 5.2 presented rules to model the user context requirements in a set of simple levels. These levels are able to express the demand of each user context requirement in a way that simplifies the processes by which such information is used in the network. Since the VNs are built and configured to personalize the communications of a group of users that share the same (or similar) context needs, Section 5.2 also gave examples of policies to automatically map these context needs into suitable VN structures, protocols or resources. Within this scope, the novelty of our approach resides in the fact that the network resources and operations are not selected according to general network characteristics, but they are selected to meet specific user context requirements. These requirements are modeled as multi-variate context data, and automatically mapped to corresponding VN features.

When an user arrives at a specific WMN node, it is important to automatically select and assign the user to a VN that fits his/her context requirements. Such VN can be available in the WMN node where the user is connected to, in any of the nodes of the user neighborhood, or in other part of the WMN, being then important to select the WMN nodes through which a VN adaptation can be performed. Therefore, Section 5.3 proposed a new user-centric metric to select the best fitting VN to assign to an user, and Section 6.2 addressed the selection of the best path to adapt a VN in the WMN. Concerning the VN selection, we proposed a way to handle certain ranges of variation for the demands of the user context requirements during the user connectivity time, in order to: *(i)* reduce the number of VN re-associations during the user connectivity time; *(ii)* increase the possibility of re-using VNs available in the WMN, while decreasing the need to create new VNs and its associated complexity. In Section 5.4, we evaluated, by means of analytical and simulation tools, the feasibility and performance benefits of this VN selection metric. Regarding the selection of the WMN path to create or adapt a VN, we considered the WMN resource availability to build VNs through short and non-congested paths, which impact was evaluated in Section 6.5.

Assuming that the user context requirements that admit high variation ranges are the ones that are more probable to change during the user connectivity time, the user-centric context perspective was also considered to improve the discovery mechanism of context-aware VNs in the WMN. To this purpose, we defined a scheme in Section 4.2.1, which was then enhanced in Section 6.2.3, to enable the dynamic cooperation among similar context-aware VNs. From an user-centric perspective, similar context-aware VNs are the ones that are most probable to deal with the user context requirements with high probability to change. In the evaluations performed in Section 6.5, we observed the added value of this context-aware inter-VN cooperation scheme, since it enabled the: *(i)* faster user association to a fitting VN than other control schemes, due to the context-driven redirection of the VN discovery mechanism in the WMN; *(ii)* automatic user re-association to a better fitting VN in case of context change, by inspecting the information about the location of similar VNs in the WMN.

With regard to the QoE perceived by an user that is connected to a specific context-aware VN, this Thesis also provided, in Section 5.2, a brief discussion on how to model the impact of distinct VN context-aware mechanisms on the performance of VN flows, and consequently on the perceived QoE of VN users. Concerning the QoE-awareness thematic, this Thesis focused on the QoE-aware routing mechanism within a specific context-aware VN. With this purpose in mind, Section 5.4 defined and evaluated a QoE-aware reinforcement learning strategy for VN routing. This approach first gathers the QoE feedbacks of ongoing VN users and then autonomously adjusts the routes among two distinct VN nodes, in order to maximize the overall QoE for VN users in an adaptive fashion, and in response to VN topology or resource availability dynamics.

7.1.4 Distributed VN Control & Management

In our approach, VNs need to be frequently reconfigured to react to user mobility or context change, new VNs have to be created to deal with the context requirements of new users, or VNs can be removed, if not used anymore, to free the wireless resources that were assigned to them.

In order to find or define solutions that properly deal with these challenges, it is important to leave behind the limitations of centralized approaches to control and manage nodes and networks. These approaches are not scalable and robust to cope with the high dynamics of an architecture driven by the context information that characterize users, especially when such information is used to differentiate and control multiple competitive VNs that share the same physical WMN infrastructure. Here, users can be constantly arriving, moving and leaving the WMN, and might not be always reachable from a central unit. Thus, the central unit might not have a real-time view of the WMN, which can jeopardize a fair control and management of VNs in the WMN. Therefore, this Thesis focused on the distribution of control knowledge and functionalities along the distinct elements that compose the multi-VN architecture, in order to increase the flexibility, scalability and resilience of the VN control and management.

In this context, Section 4.2 defined a basic distributed mechanism to locally and globally discover and extend the exactly fitting VN to assign to an user that arrives at a specific WMN node. In such mechanism, an user can be automatically assigned to a VN available in the WMN node where he/she is connected to, or it can be triggered the smooth reconfiguration of any fitting VN available in the user neighborhood, or in other node of the WMN. In Section 4.4, we evaluated the user association delay and control overhead obtained with the proposed distributed mechanism, by resorting to the analytical model defined in Section 4.3 and to the NS-2 simulation framework presented in Section 3.4. Based on this evaluation, we analyzed the potentials of distributing the control functionalities among the WMN nodes, in order to enable the fast discovery and reconfiguration of VNs in the WMN. Moreover, we observed the benefits of considering the key-based routing features of DHTs to aid the global context-driven discovery of VNs in the WMN. The work of Chapter 4 also allowed us to detect the issues that need to be addressed in a more complete solution for the control and management of the multi-VN architecture, which was the focus of Chapter 6.

Therefore, in Sections 6.2 and 6.3, we first detailed a framework to enable the distribution of control knowledge among distinct WMN nodes, VN nodes and VNs. We then defined the mechanisms to enable the distributed cooperation of these entities to autonomously discover, select, create, adapt and remove VNs on-demand to react to any type of user context change or mobility. In Section 6.4, we evaluated the advantages of the distributed control framework, mainly in terms of the delay required to associate an user to a fitting VN, when compared to a centralized and a decentralized control scheme. This study was performed by defining and evaluating an analytical model, which was properly validated against a specific NS-2 simulation setup. The comparison of our distributed scheme to control and manage VNs against other distributed approaches was performed in Section 6.5. Despite imposing a higher control overhead than the other approaches, mainly due to the required coordination among similar context-aware VNs, the main advantages of our scheme can be highlighted by the: *(i)* decrease of the delays required to associate an user to a fitting VN, which was more noticeable in case of the need to perform the re-association of the user to another VN due to user context change; *(ii)* reduction of the number of WMN nodes involved in a VN reconfiguration; *(iii)* slightly increase of the overall user perceived QoE and WMN capacity.

In summary, this Thesis proposed a multi-virtual architecture that is able to personalize a WMN environment in order to differently meet multiple expectations of users and requirements of their devices or services. Thus, we believe that this architecture can contribute to the enhancement of the user centricity landscape of future networking architectures, mechanisms and protocols. To provide effective means to increase the flexibility and scalability of the proposed architecture, and very much aligned with the need to adapt the virtual environment on-demand to react to the user and WMN dynamics, we focused on the context-aware and distributed behavior of several control and management mechanisms that are related to the overall architectural functionality. Thereby, this Thesis can be considered an important step to empower novel context-aware and distributed control mechanisms, in order to be integrated in future networking environments. Finally, in the context of the evaluations performed along this Thesis, we think that the proposed analytical models, which were properly validated against specific NS-2 simulation scenarios, are an added value to be extended to other scenarios and approaches.

7.2 Guidelines for Future Work

Within the scope of the conducted research, there are numerous possible avenues for further work that can directly complement the architecture proposed in this Thesis, along with its control and management mechanisms. Certainly, there are several aspects that have room for improvement in the presented research, and these aspects can create opportunities for a more advanced study of several concepts addressed in this Thesis.

To comply with this vision, the following Sections provide several directions to investigate other aspects that can improve the definition, or the obtained results, of some of the concepts explored in this Thesis. Such aspects are related to: *(i)* the use of more intelligent context modeling techniques; *(ii)* the adoption of dynamic schemes for the mapping, scheduling and switching of WMN nodes, interfaces and channels among the VNs; *(iii)* the deployment of a real testbed to evaluate the effectiveness and usefulness of the proposed architecture and its intrinsic mechanisms.

7.2.1 Context Modeling

When dealing with context information, one of the key challenges is the consistent and coherent modeling and structuring of context information. These mechanisms need to be extensible and flexible enough to accommodate existing and future aspects of context information, which can be derived from distinct scenarios and applications.

In the proposed architecture, these context-aware functionalities are extremely important. Here, the VN structures, protocols and resources need to be dynamically selected, assigned, configured, adapted and removed on-demand, in order to quickly react to user context/QoE change and WMN resource availability.

In this Thesis, we made use of a simple scheme to model user context requirements in levels, which are then mapped to specific VN features by using simple context-aware rules or policies. Moreover, we considered a basic mechanism to model the range of variation for the demand of each user context requirement during the user connectivity time, assuming that the user context requirements that admit high variation ranges are the ones that are more probable to change. Finally, we provided a very preliminary insight on how to model the impact of distinct VN context-aware mechanisms on the perceived QoE of VN users.

From this point of view, one of the primary directions to improve the work presented here, is to study other ways to model the user context requirements, through the investigation and evaluation of the performance trade-offs and gains/losses of, e.g., key-value pairs, tagged encoding, object-oriented models or logic-based models. This can be important to aid the measurement and interpretation of similar context information at execution time, and consequently, to ease the mechanism by which users, that share similar context patterns, are grouped in the same VNs. Moreover, it is important to model the user context requirements as independent parameters, and also to correlate the demands and variation ranges of such requirements through probabilistic relationships. Additionally, whenever the user context requirements show some regularities over time, these regularities can be learned or predicted based on historical and current information. This predicted information can then be used to estimate the future context requirements and their variation ranges, so that proactive VN control and management can take place. Finally, it is important to define sophisticated closed-form expressions, or more general rules or frameworks, to model the impact of the user context information on the user perceived QoE.

7.2.2 Channel Mapping, Scheduling & Switching

The use of network virtualization, as a flexible and programable tool to built the proposed context-aware WMN architecture, requires the definition of a mechanism for the dynamic mapping, scheduling and switching of the interfaces and channels to be assigned to a specific VN at a specific WMN node. This topic is extremely important, and certainly requires a Thesis of its own, in order to enable isolated and non-interfering context-aware WMN communications, while optimizing the usage of the WMN nodes, interfaces and channels.

Along this Thesis, we introduced several considerations and simplifications with regard to this topic, especially in the NS-2 simulation environment. For instance, we assumed that each VN worked in a pre-defined and static wireless channel, that was available at every WMN node supporting such VN. Nevertheless, this premise cannot be directly applied to a real WMN environment, in which the number of interfaces and technologies supported by each WMN node can be different, as well as the number of working channels per interface.

Therefore, there is now space for improvements in this thematic. It is important to define a more intelligent and context-aware mechanism for the dynamic and statistic mapping, scheduling and switching of interfaces and channels among the VNs.

In the definition of such mechanism, it is first required to deploy a platform that dynamically and precisely gathers the resource availability and interference level at each WMN node, as well as the number of available interfaces and channels. In the following, such information can be used as input of an algorithm that statistically decides the interface and channel to be assigned to each VN node. This algorithm can be driven by several context-aware metrics derived from the context characteristics of each VN. For instance, as described in Section 5.2, the VNs, where users that pay more (or less) to be served are connected to, can be assigned to higher (or lower) quality channels. Moreover, the VN channel assignment algorithm can be influenced by the QoS requirements of the VN services.

Another idea, that is yet to be explored, is to study an algorithm that not only assigns an interface and channel to each VN node, but also enables the possibility to control the transmission power and antenna mode/directionality, all at the same time. Following this strategy, the scheduling and allocation of WMN resources among the multitude of context-aware VNs can certainly be characterized by a higher degree of flexibility.

Finally, the mechanism to dynamically control the interface, channel, transmission power and antenna characteristics of each VN node, can be designed in a centralized or distributed way. This implies the use of distinct communication protocols to synchronize and coordinate the whole WMN, thus requiring the evaluation of the benefits and drawbacks of each approach.

7.2.3 Real World Deployment

In order to evaluate several performance aspects of the concepts proposed in this Thesis, and to not let our efforts to become simply conceptual, we also defined several analytical models, which were properly validated against the results of a NS-2 simulation platform.

Nevertheless, the lack of a proof-of-concept prototype can be a limiting factor for the consolidation and wide applicability of the work presented in this Thesis. Indeed, all of the proposed solutions were not implemented, deployed and tested in a real networking environment, and the analytical models and NS-2 simulation scenarios relied on several premises that do not stand when real networking equipment is used. Therefore, it is important to define a real testbed to enable a complete assessment of the effectiveness and usefulness of the architecture, mechanisms and models proposed in this Thesis, in an integrated way.

With the implementation of this testbed, especially if resorting to highly reconfigurable, controllable and monitorable networking equipment, the real benefits and drawbacks of using network virtualization can be evaluated in terms of, e.g., level of wireless interference, QoS metrics and user perceived QoE. Moreover, with a complete definition of a test platform, we can assess the signaling overhead and delay of the distributed VN control framework, when applied to distinct WMN scenarios, services and applications with a selected sample of end-users. Such results need then to be compared against other distributed, decentralized or centralized implementations.

Finally, the chance to evaluate the Thesis' concepts with deployed prototypes and experimental validation, can become these concepts mature and ready to be incorporated in real and usable products, possibly bringing benefits for future networking environments. This is currently being addressed in parallel in the research group.

7.3 Evolving Paradigms

Along this Thesis, we followed a concrete line of research that involved the definition and evaluation of an architecture, along with a set of frameworks, mechanisms and models, to meet the research goals presented in Chapter 1. In the last two Sections, we presented the contributions that resulted from the followed line of research, as well as several directions that can directly enhance or complement such contributions.

Nevertheless, the work of this Thesis should not only contribute to a compiled set of results and contributions, but should also stimulate the integration of novel paradigms to evolve such work into other domains. It is then important to apply the knowledge gained throughout this Thesis to define other lines of research, that could extend the range of concepts explored under the vision of the proposed architecture for WMNs.

Therefore, this Section provides several paths and insights, that others may follow, in order to link the work of this Thesis with other paradigms that span from the: *(i)* energy-efficient resource management; *(ii)* multi-homing and multi-path concepts in virtualized environments; *(iii)* secure multi-VN inter-operability.

7.3.1 Energy-Efficient Resource Management

One of the paradigms that have recently attracted a huge attention by the research community is the energy-efficient networking, which is becoming a fundamental issue, especially in wireless environments. In this Thesis, we presented an user-centric architecture that can be used to personalize WMNs on-demand, in order to, for instance, meet the ongoing energy constraints of user devices. This is achieved through the proper selection of the mechanisms and protocols that need to be in place in a specific VN, in order to minimize the overall VN energy consumption. Moreover, we defined, in Section 5.2, a very preliminary empirical model to describe the impact of the overall VN energy savings on the performance of VN flows, and consequently, on the decrease of the QoE of VN users.

However, this Thesis did not devote any type of attention to the energy-efficient resource management. This thematic can be important to minimize the overall energy expenditures in the creation and maintenance of the proposed architecture, and it can clearly be considered a natural evolution of the work presented in this Thesis. Further, despite the recent advances in the energy-efficient networking, there is still room for improvement in the design, materialization and test of more general algorithms, models and frameworks for the energy-efficient resource management.

This way, as a first step towards addressing this thematic in the scope of the multi-VN architecture for WMNs, an analytical model can be defined to describe the relation among the number and properties of WMN nodes (e.g., bandwidth or power consumption), the number of VNs available in the WMN, the size of each VN (number of virtual nodes) and the desired properties of each VN (e.g., QoS or security level).

This model can then be applied to determine the number of active powered-on WMN resources that should be assigned to each VN, in order to not only provide enough resources to ensure the fulfillment of the VN requirements in a given context, but also to minimize the overall WMN energy consumption. From this perspective, the envisioned model for an energy-efficient resource management shifts the traditional resource management schemes for communication infrastructures: while the common approaches for resource management are based on the fixed input and maximum output principle, where a set of fixed physical resources are exploited to maximize the network benefit (e.g., in terms of QoS, security or performance), the proposed line of research appeals for a model that relies on the fixed output and minimum input principle, where a fixed benefit (e.g., a determined QoS or security level) is made available by using a minimal set of physical resources.

In the definition of this model, all the energy-relevant properties of the physical and virtual resources have to be identified, and functions are required to describe the relationships and interactions among the different properties, as well as their influence on the overall energy consumption. Moreover, network polling and monitoring concepts have to be found that are able to provide the data needed by the models, which can include information about the level of utilization of physical and virtual resources, and about the current power consumption and temperature of these resources.

Once the number of active powered-on WMN resources is determined, the optimal placement of the VNs in the WMN has to be found, in order to increase the utilization and capacity of the WMN, while minimizing its overall energy consumption. This VN mapping problem can foster the need to define and implement mechanisms to allow the live migration of VN nodes, services and resources from a specific WMN node to another, since several WMN nodes can change into dormant modes due to energy-efficiency purposes. It can also be required to turn-off or hibernate several VN nodes, in order to maximize the number of coexistent VNs without degrading the performance of any of them. If necessary, the WMN or VN nodes in low power modes can be woken up to ensure the fulfillment of a certain VN requirement. Finally, the WMN communication protocols have to be optimized in a way that they actively support an energy-efficient resource management of the multi-VN architecture.

7.3.2 Multi-Homing & Multi-Path Optimization

Due to the recent digital and technological networking advances, mobile devices are becoming equipped with multi-technology interfaces, being able to support multi-homing. Such technique enables these devices to be connected to different networks and paths in parallel, in order to provide a better support for the services currently requested.

In the context of the multi-interface support brought by the innovative devices that mesh clients may use, network virtualization can allow the multi-homing optimization over different VNs, in which such clients are connected to, and multi-path optimization over distinct VN paths and infrastructure providers.

From this perspective, the parallel and simultaneous use of multiple VNs, paths and connections can decrease the probability of a mesh client to get disconnected, thus increasing the availability of WMN communications. In addition, an effective selection of the transmission paths can avoid high delays affecting QoE, and also increase the throughput required for sustaining a certain level of QoE. This can lead to a higher reliability, load-balancing and fault tolerance of the WMN in case of congestion.

Nevertheless, there is a lack of robust simulation and mathematical models capable of describing the fundamental performance, overhead and reliability gains and drawbacks of using concurrent multi-path transmissions, even among multi-homed networks, where data communications are delivered over several network paths instead of selecting only one.

Moreover, the interaction, correlation and interference among distinct network paths is expected to be complex, and thus, the path scheduling, multiplexing and switching mechanisms have to consider a very detailed description of the paths, possibly based on statistical characteristics. In this scope, the selection of the transmission paths from one or more context-aware VNs can follow a large variety of strategies, which can be defined, implemented and tested based on the quality, quantity and level of the paths provided by each VN.

Finally, it is important to limit the overall network oscillations and path switching delays, and improve the data buffering and re-ordering mechanisms in the WMN.

7.3.3 Secure Multi-VN Inter-Operability

Along this Thesis, we proposed a distributed control structure (or overlay), driven by the cooperation among specific nodes of different context-aware VNs, in order to accelerate the global discovery of points of attachment of VNs available in the WMN. Moreover, in the previous Section, we encouraged the application of multi-homing and multi-path transmission concepts to allow the traffic management optimization over distinct VNs, paths and infrastructure providers. These two mechanisms require the dynamic and secure interaction and communication among elements of different VNs, since these VNs can have their own structure and control administration. Nevertheless, this VN inter-operability thematic was not addressed in-depth throughout this Thesis (for instance, the distributed control structure was not developed with security in mind). Therefore, the definition of a more general and secure scheme to improve the inter-operability and spontaneous collaboration among VNs, can be considered another line of research that arises from the concepts explored in this Thesis.

First, in the context of the distributed control structure proposed in this Thesis, it is important to define a protocol that allows the secure cooperation among the VN controllers. This secure VN cooperation involves an in-depth analysis of the relevant current standards used in virtualized environments, which can enable the detection of common denominators with our architecture. Then, it is necessary the interfacing to allow VN controllers to communicate with the protocols defined by these standards, in order to: *(i)* negotiate the parameters required to transfer individual VN security features (e.g., keys and authorization issues) between two VN controllers; *(ii)* determine which control information can be exchanged between two VN controllers under what circumstances.

Each VN can be specialized to meet specific context characteristics and operation modes, and therefore, these VNs can be both diverse in terms of assigned users and running security, mobility, addressing and routing protocols. Due to the different requirements of these protocols, the interaction among the protocols of distinct VNs can then be virtually impossible or possible only to a very limited extend. This way, in case of applying multi-path transmission concepts over two VNs characterized by completely different context features, it is important to analyze the interactions that need to be performed between these VNs, and derive translation schemes for the messages exchanged between elements of these VNs. For instance, to allow the inter-operability between two VNs with different security requirements or different

E2E protocols (such as TCP and UDP), it is required a strong analysis on how and if these VNs can inter-operate, and in the positive case, how context, protocols and messages can be translated and mapped from one VN to another.

Finally, due to the user-centric and context-driven functionality of the architecture, a privacy and identity management should protect critically sensitive context-aware information and the disclosure of this information related to the user. Moreover, when mobility dynamics are in place (both mobility of users and mobility of virtual resources), and this context-aware information is moved among the nodes of the same or different VNs, these privacy issues are even exacerbated.

7.4 Final Remarks

In this Thesis, we have looked at different aspects of user-centric networking, context-awareness and distributed network control and management, and how they can work together in a multi-virtual WMN environment.

To achieve this, we proposed an architecture to split a WMN in a set of VNs, being each VN specialized to meet the particular requirements of a group of users, devices and applications. We also defined a framework to control and manage the VNs in a distributed way, in order to quickly react to the context dynamics of users and WMN nodes. Therefore, we think this Thesis is an important step to significantly empower two growing trends in networking research: *(i)* support for individualized and personalized solutions and services for mobile users; *(ii)* ease the administration and maintenance of conventional (virtual) networks.

Moreover, the network virtualization paradigm, proposed in this Thesis, can certainly increase the flexibility of newly deployed user-oriented network structures, since it allows network providers to modify their network topologies, to meet specific user context requirements, in a real-time manner. It can also stimulate the integration of WMNs as a high quality network part of future Internet architectures and virtual platforms of network operators.

Finally, throughout this Thesis, we defined several mechanisms for fundamental challenges raised by the proposed architecture. From the conclusions supported by the evaluation of such mechanisms, we believe they can be especially important in the definition of more advanced solutions to allow the personalized and simultaneous accommodation of highly diverse users in the same network infrastructure, and the distributed control and management of nodes and networks.

Bibliography

[1] B.N. Schilit and M.M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32, Sep.-Oct. 1994.

[2] Anind K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, Jan. 2001.

[3] Jianli Pan, S. Paul, and R. Jain. A survey of the research on future internet architectures. *IEEE Communications Magazine*, 49(7):26–36, Jul. 2011.

[4] N.M.M.K. Chowdhury and R. Boutaba. Network virtualization: state of the art and research challenges. *IEEE Communications Magazine*, 47(7):20–26, Jul. 2009.

[5] S. Benedetto and L. M. Correia. A Vision into Medium-Long Term Research in Wireless Communications. *Advances in Electronics and Telecommunications*, 1(1):13–25, Apr. 2010.

[6] E. Gustafsson and A. Jonsson. Always best connected. *IEEE Wireless Communications*, 10(1):49–55, Feb. 2003.

[7] I.F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9):23–30, Sep. 2005.

[8] R. Matos and S. Sargento. Context-aware Connectivity and Mobility in Wireless Mesh Networks. In *ICST Conference on Mobile Networks And Management (MONAMI)*, Oct. 2009.

[9] K. Hummel, A. Hess, S. Sargento, R. Matos, K. Tutschku, and H. Meer. Context-based Wireless Mesh Networks: A Case for Network Virtualization. In *Workshop on Future Internet Architectures: New Trends in Service Architectures, Co-located with ICT Mobile Summit*, Jun. 2009.

[10] R. Matos, S. Sargento, K. Hummel, A. Hess, K. Tutschku, and H. Meer. Context-Based Wireless Mesh Networks: A Case for Network Virtualization. *Special Issue on "Future Internet Services and Architectures: Trends and Visions", Springer Telecommunication Systems*, 52(3), Mar. 2013.

[11] R. Matos and S. Sargento. Data Communications over Context-Based WMNs: Delay Performance Evaluation. In *IFIP/IEEE International Workshop on Bandwidth on Demand and Federation Economics (BoD), Co-located with IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Apr. 2010.

[12] R. Matos and S. Sargento. Context-Based Connectivity over Multi-Virtual Wireless Mesh Networks: Analytical Study. In *IEEE Symposium on Computers and Communications (ISCC)*, Jun. 2010.

[13] S. Sargento, R. Matos, K. Hummel, A. Hess, S. Toumpis, Y. Tselekounis, G. Stamoulis, H. Meer, Y. Al-Hazmi, and M. Ali. *Wireless Multi-Access Environments and QoS Provisioning*, chapter User-Driven Multi-Access Communications Through Virtualization. IGI GLOBAL, 2011.

[14] R. Matos, C. Marques, and S. Sargento. Context-Based Connectivity and Characterization of Wireless Mesh Networks: Simulation Study. In *IEEE Workshop on Mobile Computing and Emerging Communication Networks (MCECN), Co-located with IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2010.

[15] R. Matos, C. Marques, and S. Sargento. Context-Based Connectivity in Wireless Mesh Networks: Analytical vs Simulation Studies. In *IEEE International Conference on Computer as a Tool (EUROCON)*, Apr. 2011.

[16] R. Matos, C. Marques, S. Sargento, K. Hummel, and H. Meyer. Analytical Modeling of Context-Based Multi-Virtual Wireless Mesh Networks. *Special Issue on "Models And Algorithms For Wireless Mesh Networks", Elsevier Ad Hoc Networks*, 2011. ISSN: 1570-8705.

[17] Funkfeuer Vienna. Information available at http://www.funkfeuer.at/.

[18] R. Matos, C. Marques, and S. Sargento. On-Demand Selection, Control and Management of Context-Based Wireless Mesh Networks. In *IEEE International Conference on Communications (ICC)*, Jun. 2012.

[19] R. Matos, N. Coutinho, C. Marques, S. Sargento, J. Chakareskiy, and A. Kassler. Quality of Experience-Based Routing in Multi-Service Wireless Mesh Networks. In *IEEE Workshop on Realizing Advanced Video Optimized Wireless Networks (ViOPT), Co-located with IEEE International Conference on Communications (ICC)*, Jun. 2012.

[20] R. Matos and S. Sargento. Distributed Control and Management of Context-Based Wireless Mesh Networks. In *ICST Conference on Mobile Networks And Management (MONAMI)*, Sep. 2011.

[21] R. Matos, C. Marques, and S. Sargento. Distributed Approach to Control and Manage Context-Based Multi-Virtual Networks. *ACM/Springer Mobile Networks and Applications (MONET)*, 17(4):447–462, 2012.

[22] R. Matos, C. Marques, and S. Sargento. Distributed Architecture to Build and Control Context-Based Wireless Networks Through Virtualization. *Under Preparation.*

[23] EURO-NF. FP7 European (Network of the Future) Network of Excellence. Information available at http://euronf.enst.fr/en_accueil.html.

[24] CAMYN. Context-Aware Mobility in Wireless Mesh Networks. Information available at http://cs.univie.ac.at/research/projects/projekt/infproj/789/.

[25] MOMO. Multi-Overlays for Multi-hOming: A Wireless Mesh Network Use Case. Information available at http://cs.univie.ac.at/research/projects/projekt/infproj/788/.

[26] Ubiquimesh. Cross-Layer Optimization in Multiple Mesh Ubiquitous Networks. Information available at http://ubiquimesh.av.it.pt/.

[27] UMM. User-centric Mobility Management. Information available at http://siti.ulusofona.pt/index.php/research/projects/149-umm.

[28] MC-WMNs. Multiple Context-based Wireless Mesh Networks. Information available at http://www.fct.pt/apoios/projectos/consulta/areas.phtml.en?idElemConcurso=4212.

[29] IEEE 802.11 Wireless Local Area Networks. Information available at http://www.ieee802.org/11/.

[30] IEEE 802.11s TG. Information available at http://www.ieee802.org/11/Reports/tgs_update.htm.

[31] IEEE 802.15 Working Group for WPAN. Information available at http://www.ieee802.org/15/.

[32] IEEE 802.16's Relay Task Group. Information available at http://www.ieee802.org/16/relay/.

[33] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. K. Costa, O. C. M. B. Duarte, D. G. Passos, C. V. N. De Albuquerque, D. C. M. Saade, M. G. Rubinstein, and M. G. Rubinstein. Routing Metrics and Protocols for Wireless Mesh Networks. *IEEE Network*, 22(1):6–12, Jan.-Feb. 2008.

[34] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). In *RFC 3626*, Oct. 2003.

[35] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24(4):234–244, Oct. 1994.

[36] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. In *RFC 3561*, Jul. 2003.

[37] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. In *RFC 4728*, Feb. 2007.

[38] Daniel Aguayo, John Bicket, and Robert Morris. SrcRR: A High Throughput Routing Protocol for 802.11 Mesh Networks. In *ACM Whitepaper*, 2003.

[39] K. N. Ramachandran, M. M. Buddhikot, G. Chandranmenon, S. Miller, Belding E. M. Royer, and K. C. Almeroth. On the Design and Implementation of Infrastructure Mesh Networks. *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[40] L. Couto, J.P. Barraca, S. Sargento, and R.L. Aguiar. Fast mobility in proactive routing protocols. In *IST Mobile & Wireless Communications Summit*, 2008.

[41] M. Bahr. Update on the Hybrid Wireless Mesh Protocol of IEEE 802.11s. In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2007.

[42] Jangeun Jun and Mihail L. Sichitiu. Mrp: Wireless mesh networks routing protocol. *Elsevier Computer Communications*, 31(7):1413–1435, May 2008.

[43] Sanjit Biswas and Robert Morris. Exor: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, Aug. 2005.

[44] E. Rozner, J. Seshadri, Y. Mehta, and Lili Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 8(12):1622–1635, Dec. 2009.

[45] Nagesh S. Nandiraju, Deepti S. Nandiraju, and Dharma P. Agrawal. Multipath routing in wireless mesh networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2006.

[46] Seungjoon Lee, Bobby Bhattacharjee, and Suman Banerjee. Efficient geographic routing in multihop wireless networks. In *ACM International Symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2005.

[47] Y. Yang, J. Wang, and R. Kravets. Designing routing metrics for mesh networks. In *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[48] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2003.

[49] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2004.

[50] Yaling Yang, Jun Wang, and Robin Kravets. Interference-aware Load Balancing for Multihop Wireless Networks. Technical report, University of Illinois at Urbana-Champaign, 2005.

[51] A.P. Subramanian, M.M. Buddhikot, and S. Miller. Interference aware routing in multi-radio wireless mesh networks. In *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2006.

[52] Devu Manikantan Shila and Tricha Anjali. Load aware traffic engineering for mesh networks. *Elsevier Computer Communications*, 31(7):1460–1469, May 2008.

[53] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *ACM SIGCOMM Computer Communication Review*, 2004.

[54] Jonathan Guerin, Marius Portmann, and Asad Pirzada. Routing metrics for multi-radio wireless mesh networks. In *Australasian Telecommunication Networks and Applications Conference (ATNAC)*, 2007.

[55] P. Kyasanur, Jungmin So, C. Chereddi, and N.H. Vaidya. Multichannel mesh networks: challenges and protocols. *IEEE Wireless Communications*, 13(2):30–36, Apr. 2006.

[56] Chandrakanth Chereddi, Pradeep Kyasanur, and Nitin H. Vaidya. Design and implementation of a multi-channel multi-interface network. In *ACM International workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN)*, 2006.

[57] Atul Adya, Paramvir Bahl, Jitendra Padhye, Alec Wolman, and Lidong Zhou. A multi-radio unification protocol for ieee 802.11 wireless networks. In *IEEE International Conference on Broadband Networks (BROADNETS)*, 2004.

[58] Mahesh K. Marina, Samir R. Das, and Anand Prabhu Subramanian. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Elsevier Computer Networks*, 54(2):241–256, Feb. 2010.

[59] Jungmin So and Nitin Vaidya. Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *ACM International Symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2004.

[60] Paramvir Bahl, Ranveer Chandra, and John Dunagan. Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad hoc wireless networks. In *ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2004.

[61] R. Maheshwari, H. Gupta, and S.R. Das. Multichannel mac protocols for wireless networks. In *IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2006.

[62] Pradeep Kyasanur. Routing and interface assignment in multi-channel multi-interface wireless networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2005.

[63] Ian F. Akyildiz and Xudong Wang. Cross-layer design in wireless mesh networks. *IEEE Transactions on Vehicular Technology*, 57(2):1061–1076, Mar. 2008.

[64] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 3(1):74–85, Jan. 2004.

[65] Chun-Chia Chen and Duan-Shin Lee. A Joint Design of Distributed QoS Scheduling and Power Control for Wireless Networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[66] Wei Wang, Xin Liu, and D. Krishnaswamy. Robust routing and scheduling in wireless mesh networks under dynamic traffic conditions. *IEEE Transactions on Mobile Computing*, 8(12):1705–1717, Dec. 2009.

[67] Jihui Zhang, Haitao Wu, Qian Zhang, and Bo Li. Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks. In *IEEE International Conference on Broadband Networks (BROADNETS)*, 2005.

[68] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):50–65, Apr. 2004.

[69] G. Jakllari, S. Eidenbenz, N. Hengartner, S.V. Krishnamurthy, and M. Faloutsos. Link positions matter: A noncommutative routing metric for wireless mesh network. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[70] V. Navda, A. Kashyap, and S.R. Das. Design and evaluation of imesh: an infrastructure-mode wireless mesh network. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2005.

[71] Yair Amir, Claudiu Danilov, Michael Hilsdale, Raluca Musăloiu-Elefteri, and Nilo Rivera. Fast handoff for seamless wireless mesh networks. In *ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2006.

[72] Yair Amir, Claudiu Danilov, Raluca Musaloiu-Elefteri, and Nilo Rivera. An Inter-domain Routing Protocol for Multi-homed Wireless Mesh Networks. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2007.

[73] M. Ito, T. Shikama, and A. Watanabe. Proposal for a Wireless Mesh Network that Realizes Seamless Handover and Its Simulation Results. In *IEEE International Symposium on Communications and Information Technologies (ISCIT)*, 2007.

[74] Hung Quoc Vo and Choong Seon Hong. A novel scheme for seamless hand-off in Wireless Mesh Networks. In *IEEE International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2008.

[75] Bachar Wehbi, Wissarn Mallouli, and Ana Cavalli. Light Client Management Protocol for Wireless Mesh Networks. In *ACM International Conference on Mobile Data Management (MDM)*, 2006.

[76] John C. Lin and Sampath Rangarajan Rangarajan. LIHP: A Low Latency Layer-3 Handoff Scheme for 802.11 Wireless Networks. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2006.

[77] Hui Wang, Quan Huang, Yong Xia, Yichuan Wu, and Yingang Yuan. A Network-Based Local Mobility Management Scheme for Wireless Mesh Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[78] A. Capone, M. Cesana, S. Napoli, and A. Pollastro. MobiMESH: A Complete Solution for Wireless Mesh Networking. In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2007.

[79] R. Wishart, A. Pirzada, M. Portmann, and J. Indulska. A Light-Weight Client Mobility Approach for Infrastructure Mesh Networks. In *IEEE International Conference on Networks (ICON)*, 2007.

[80] Maoshen Ren, Chao Liu, Huizhou Zhao, Tong Zhao, and Wei Yan. MEMO: An Applied Wireless Mesh Network with Client Support and Mobility Management. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2007.

[81] Yan He and D. Perkins. BASH: A backhaul-aided seamless handoff scheme for Wireless Mesh Networks. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2008.

[82] Zhenxia Zhang and Azzedine Boukerche. A Novel Mobility Management Scheme for IEEE 802.11-Based Wireless Mesh Networks. In *IEEE International Conference on Parallel Processing - Workshops (ICPPW)*, 2008.

[83] S. Speicher and C.H. Cap. Fast Layer 3 Handoffs in AODV-based IEEE 802.11 Wireless Mesh Networks. In *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, 2006.

[84] S. Speicher. OLSR-FastSync: Fast Post-Handoff Route Discovery in Wireless Mesh Networks. In *IEEE Vehicular Technology Conference (VTC)-Fall*, 2006.

[85] Hung-Yu Wei, S. Kim, S. Ganguly, and R. Izmailov. Seamless Handoff Support in Wireless Mesh Networks. In *IEEE Workshop on Operator-Assisted (Wireless Mesh) Community Networks*, 2006.

[86] W. Lei and B. Landfeldt. On the problem of placing Mobility Anchor Points in Wireless Mesh Networks. In *IEEE Workshop on Operator-Assisted (Wireless Mesh) Community Networks*, 2006.

[87] Luigi Iannone and Serge Fdida. MeshDV: A Distance Vector mobility-tolerant routing protocol for Wireless Mesh Networks. In *ACM International workshop on Multi-hop ad hoc networks: from theory to reality (REALMAN)*, 2005.

[88] R. Baumann, O. Bondareva, S. Heimlicher, and M. May. A Protocol for Macro Mobility and Multihoming Notification in Wireless Mesh Networks. In *IEEE International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, 2007.

[89] Bin Xie, Yingbing Yu, Anup Kumar, and Dharma P. Agrawal. Load-balancing and Inter-domain Mobility for Wireless Mesh Networks. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2006.

[90] Franck Rousseau et al. An architecture for seamless mobility in spontaneous wireless mesh networks. In *ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture (Mobiarch)*, 2007.

[91] Edward Bortnikov, Israel Cidon, and Idit Keidar. Scalable real-time gateway assignment in mobile mesh networks. In *ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2007.

[92] Mehdi Sabeur, Ghazi Alsukkar, Badii Jouaber, Djamal Zeghlache, and Hossam Afifi. A new Routing & Mobility Management Solution for Wireless Mesh Network. In *IEEE International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS)*, 2007.

[93] Yanfei Fan, Jingqiu Zhang, and Xuemin Shen. Mobility-Aware Multi-Path Forwarding Scheme for Wireless Mesh Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2008.

[94] R. Huang, C. Zhang, and Y. Fang. A Mobility Management Scheme for Wireless Mesh. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2007.

[95] Di-Wei Huang, Phone Lin, and Chai-Hien Gan. Design and Performance Study for a Mobility Management Mechanism (WMM) Using Location Cache for Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, 7(5):546–556, May 2008.

[96] Bo Rong, Yi Qian, Kejie Lu, and Michel Kadoch. Call Admission Control for Mobile Agent Based Handoff in Wireless Mesh Networks. In *IEEE International Conference on Communications (ICC)*, 2008.

[97] Guojun Shui and Shuqun Shen. Smooth Handoff with Relay Router in Wireless Mesh Networks. In *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2008.

[98] Jiang Xie and Xudong Wang. A Survey of Mobility Management in Hybrid Wireless Mesh Networks. *IEEE Network*, 22(6):34–40, Nov.-Dec. 2008.

[99] MIT Roofnet. Information available at http://pdos.csail.mit.edu/roofnet/doku.php.

[100] Purdue Mesh. Information available at https://engineering.purdue.edu/MESH.

[101] MeshNet. Information available at http://moment.cs.ucsb.edu/meshnet/.

[102] ORBIT Lab. Information available at http://www.orbit-lab.org/.

[103] Emulab. Information available at http://www.emulab.net/.

[104] Sherif M. Elrakabawy, Simon Frohn, and Christoph Lindemann. A scalable dual-radio wireless testbed for emulating mesh networks. *ACM Wireless Networks*, 16(8):2191–2207, Nov. 2010.

[105] Yang Su and Thomas Gross. Validation of a miniaturized wireless network testbed. In *ACM International workshop on Wireless network testbeds, experimental evaluation and characterization (WiNTECH)*, 2008.

[106] Erik Nordström, Per Gunningberg, Christian Rohner, and Oskar Wibling. Evaluating wireless multi-hop networks using a combination of simulation, emulation, and real world experiments. In *ACM International Workshop on System Evaluation for Mobile Platforms: Metrics, Methods, and Tools (MobiEval)*, 2007.

[107] Alexander Zimmermann, Mesut Günes, Martin Wenig, Jan Ritzerfeld, and Ulrich Meis. Architecture of the hybrid mcg-mesh testbed. In *ACM International workshop on Wireless network testbeds, experimental evaluation and characterization (WiNTECH)*, 2006.

[108] Alexander Zimmermann, Mesut Gunes, Martin Wenig, Ulrich Meis, and Jan Ritzerfeld. How to study wireless mesh networks: A hybrid testbed approach. In *IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2007.

[109] CUWIN. Community wireless networks. Information available at http://cuwireless.net.

[110] Community Wireless. Information available at http://www.communitywireless.org/.

[111] Elias C. Efstathiou, Pantelis A. Frangoudis, and George C. Polyzos. Stimulating participation in wireless community networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.

[112] Joseph Camp, Joshua Robinson, Christopher Steger, and Edward Knightly. Measurement driven deployment of a two-tier urban mesh access network. In *ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2006.

[113] Manolis Delakis, Kostas Mathioudakis, Nikolaos Petroulakis, and Vasilios A. Siris. Experiences and investigations with heraklion mesh: an experimental metropolitan multi-radio mesh network. In *ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, 2008.

[114] Meraki Mesh. Information available at http://sf.meraki.com/map.

[115] SolarMESH. Information available at http://owl.mcmaster.ca/ todd/SolarMESH.

[116] Calmesh. Information available at http://calmesh.calit2.net/index.html.

[117] R. B. Dilmaghani, B. S. Manoj, B. Jafarian, and R. R. Rao. Performance evaluation of rescuemesh: a metro-scale hybrid wireless network. In *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[118] Yusuke Takahashi, Yasunori Owada, Hiraku Okada, and Kenichi Mase. A wireless mesh network testbed in rural mountain areas. In *ACM International workshop on Wireless network testbeds, experimental evaluation and characterization (WiNTECH)*, 2007.

[119] Kun-chan Lan, Zhe Wang, Mahbub Hassan, Tim Moors, Rodney Berriman, Lavy Libman, Maximilian Ott, Bjorn Landfeldt, and Zainab Zaidi. Experiences in deploying a wireless mesh network testbed for traffic control. *ACM SIGCOMM Computer Communication Review*, 37(5):17–28, Oct. 2007.

[120] Giacomo Bernardi, Peter Buneman, and Mahesh K. Marina. Tegola tiered mesh network testbed in rural scotland. In *ACM Wireless Networks and Systems for Developing Regions Workshop (WiNS-DR)*, 2008.

[121] Yi Wang, Eric Keller, Brian Biskeborn, Jacobus van der Merwe, and Jennifer Rexford. Virtual routers on the move: live router migration as a network-management primitive. *ACM SIGCOMM Computer Communication Review*, 38(4):231–242, Aug. 2008.

[122] Strix Systems. Information available at http://www.strixsystems.com.

[123] Cisco Systems. Information available at http://www.cisco.com/web/go/ outdoorwirelessnetworks.

[124] MeshDynamics. Information available at http://www.meshdynamics.com.

[125] BelAir Networks. Information available at http://www.belairnetworks.com.

[126] N. Nandiraju, D. Nandiraju, L. Santhanam, Bing He, Junfang Wang, and D.P. Agrawal. Wireless mesh networks: Current challenges and future directions of web-in-the-sky. *IEEE Wireless Communications*, 14(4):79–89, Aug. 2007.

[127] James E. Smith and Ravi Nair. The architecture of virtual machines. *IEEE Computer*, 38(5):32–38, May 2005.

[128] Mendel Rosenblum and Tal Garfinkel. Virtual machine monitors: Current technology and future trends. *IEEE Computer*, 38(5):39–47, May 2005.

[129] Joseph D. Touch, Yu shun Wang, Lars Eggert, and Gregory G. Finn. A virtual internet architecture. In *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, 2003.

[130] Thomas Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner. Overcoming the internet impasse through virtualization. *IEEE Computer*, 38(4):34–41, Apr. 2005.

[131] Nick Feamster, Lixin Gao, and Jennifer Rexford. How to lease the internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64, Jan. 2007.

[132] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, Mar. 2008.

[133] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, N. McKeown, and G. Parulkar. Flowvisor: A network virtualization layer. Technical report, Deutsche Telekom Inc. R&D Lab, Stanford University, Nicira Networks, 2008.

[134] Jad Naous, Glen Gibb, Sara Bolouki, and Nick McKeown. Netfpga: reusable router architecture for experimental research. In *ACM SIGCOMM Workshop on Programmable Routers for Extensible Services of TOmorrow (PRESTO)*, 2008.

[135] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router. In *ACM Symposium on Operating Systems Principles (SOSP)*, 1999.

[136] Scott Karlin and Larry Peterson. Vera: an extensible router architecture. *Elsevier Computer Networks*, 38(3):277–293, Feb. 2002.

[137] Mark Handley, Orion Hodson, and Eddie Kohler. Xorp: an open platform for network research. *ACM SIGCOMM Computer Communication Review*, 33(1):53–57, Jan. 2003.

[138] Norbert Egi, Adam Greenhalgh, Mark Handley, Mickael Hoerdt, Felipe Huici, Laurent Mathy, and Panagiotis Papadimitriou. A platform for high performance and flexible virtual routers on commodity hardware. *ACM SIGCOMM Computer Communication Review*, 40(1):127–128, Jan. 2010.

[139] Scope Alliance. I/O Virtualization: A NEP Perspective. Technical report, Feb. 2009.

[140] Scot Rixner. Network virtualization: Breaking the performance barrier. *ACM Queue*, 6(1):36–44, Jan.-Feb. 2008.

[141] Paul A. Karger and David R. Safford. I/o for virtual machine monitors: Security and performance issues. *IEEE Security and Privacy*, 6(5):16–23, Sep. 2008.

[142] PlanetLab. An open platform for developing, deploying, and accessing planetary-scale services. Information available at http://www.planet-lab.org/.

[143] VINI. A virtual network infrastructure. Information available at http://www.vini-veritas.net/.

[144] GENI. Global environment for network innovations. Information available at http://www.geni.net/.

[145] 4WARD. Information available at http://www.4ward-project.eu/.

[146] Minlan Yu, Yung Yi, Jennifer Rexford, and Mung Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *ACM SIGCOMM Computer Communication Review*, 38(2):17–29, Mar. 2008.

[147] Jiayue He, Rui Zhang-Shen, Ying Li, Cheng-Yen Lee, Jennifer Rexford, and Mung Chiang. Davinci: dynamically adaptive virtual networks for a customized internet. In *ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2008.

[148] Wireless Working Group-GENI. Technical document on wireless virtualization. Technical report, Sep. 2006.

[149] Domenico Giustiniano, Eduard Goma, Alberto Lopez, and Pablo Rodriguez. Wiswitcher: an efficient client for managing multiple aps. In *ACM SIGCOMM Workshop on Programmable Routers for Extensible Services of TOmorrow (PRESTO)*, 2009.

[150] MadWifi. Information available at http://madwifi-project.org/.

[151] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri. Splitap: Leveraging wireless network virtualization for flexible sharing of wlans. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[152] Lei Xia, Sanjay Kumar, Xue Yang, Praveen Gopalakrishnan, York Liu, Sebastian Schoenberg, and Xingang Guo. Virtual wifi: bring virtualization from wired to wireless. In *ACM SIGPLAN/SIGOPS International conference on Virtual execution environments (VEE)*, 2011.

[153] R. Chandra, P. Bahl, and P. Bahl. Multinet: Connecting to multiple ieee 802.11 networks using a single wireless card. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2004.

[154] Chandrakanth Chereddi, Pradeep Kyasanur, and Nitin H. Vaidya. Net-x: a multichannel multi-interface wireless mesh implementation. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):84–95, Jul. 2007.

[155] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramach, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2005.

[156] Gregory Smith, Anmol Chaturvedi, Arunesh Mishra, and Suman Banerjee. Wireless virtualization on commodity 802.11 hardware. In *ACM International workshop on Wireless network testbeds, experimental evaluation and characterization (WiNTECH)*, 2007.

[157] Joachim Sachs and Stephan Baucke. Virtual radio: a framework for configurable radio networks. In *ICST Wireless Internet Conference (WICON)*, 2008.

[158] S.L. Shrestha, J. Lee, and S. Chong. Virtualization and slicing of Wireless Mesh Network. In *Conference on Future Internet (CFI)*, 2008.

[159] Gang Ding, John Vicente, Sanjay Rungta, Dilip Krishnaswamy, and Kai Miao. Overlays on wireless mesh networks: Implementation and cross-layer searching. In *IFIP/IEEE international conference on Management of Multimedia and Mobile Networks and Services (MMNS)*, 2006.

[160] Cuitao Zhu, Di Wu, Wenqing Cheng, and Zongkai Yang. Efficient Overlay Multicast Strategy for Wireless Mesh Networks. In *IEEE Vehicular Technology Conference (VTC)-Fall*, 2008.

[161] Abu (Sayeem) Reaz, Vishwanath Ramamurthi, Dipak Ghosal, John Benko, Wei Li, Sudhir Dixit, and Biswanath Mukherjee. Enhancing Multi-Hop Wireless Mesh Networks with a Ring Overlay. In *IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2008.

[162] Yahya Al-Hazmi and Hermann De Meer. Virtualization of 802.11 interfaces for wireless mesh networks. In *IEEE Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2011.

[163] Aun Haider, Richard Potter, and Akihiro Nakao. Challenges in Resource Allocation in Network Virtualization. In *ITC Specialist Seminar on Network Virtualization*, 2009.

[164] Min Zhang, Chunming Wu, Ming Jiang, and Qiang Yang. Mapping Multicast Service-Oriented Virtual Networks with Delay and Delay Variation Constraints. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[165] Andreas Berl, Nicholas Race, Johnathan Ishmael, and Hermann de Meer. Network virtualization in energy-efficient office environments. *Elsevier Computer Networks*, 54(16):2856–2868, Nov. 2010.

[166] Yanfeng Zhang, Lixin Gao, and Cuirong Wang. Multinet: multiple virtual networks for a reliable live streaming service. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2009.

[167] Roland Bless. Network virtualization from a signaling perspective. In *IEEE International Conference on Communications (ICC)*, 2009.

[168] P. Papadimitriou, O. Maennel, A. Greenhalgh, A. Feldmann, and L. Mathy. Implementing Network Virtualization for a Future Internet. In *ITC Specialist Seminar on Network Virtualization*, 2009.

[169] C-CAST. Context casting. Information available at http://www.ict-ccast.eu/.

[170] SENSEI. Integrating the physical with the digital world of the network of the future. Information available at http://www.sensei-project.eu/.

[171] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *ACM International Conference on Pervasive Computing (Pervasive)*, 2002.

[172] Cristiana Bolchini, Carlo A. Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. A data-oriented survey of context models. *ACM SIGMOD Record*, 36(4):19–26, Dec. 2007.

[173] Ping Yi, Tianhao Tong, Ning Liu, Yue Wu, and Jianqing Ma. Security in wireless mesh networks: Challenges and solutions. In *IEEE International Conference on Information Technology: New Generations (ITNG)*, 2009.

[174] Georg Lukas and Christian Fackroth. Wmnsec: security for wireless mesh networks. In *ACM/IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2009.

[175] K. Khan and M. Akbar. Authentication in multi-hop wireless mesh networks. In *International Conference on Information Technology (ICIT)*, 2006.

[176] Yanchao Zhang and Yuguang Fang. ARSA: An Attack-Resilient Security Architecture for Multihop Wireless Mesh Networks. *IEEE Journal on Selected Areas in Communications*, 24(10):1916–1928, Oct. 2006.

[177] Yanchao Zhang and Yuguang Fang. A secure authentication and billing architecture for wireless mesh networks. *ACM/Springer Wireless Networks*, 13(5):663–678, Oct. 2007.

[178] Xin Zhao, Yuqin Lv, Tet Hin Yeap, and Bin Hou. A novel authentication and key agreement scheme for wireless mesh networks. In *IEEE Networked Computing and Advanced Information Management (NCM)*, 2009.

[179] Celia Li, Zhuang Wang, and Cungang Yang. Seaodv: A security enhanced aodv routing protocol for wireless mesh networks. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2010.

[180] Shafiullah Khan, Kok-Keong Loo, Noor Mast, and Tahir Naeem. Srpm: Secure routing protocol for ieee 802.11 infrastructure based wireless mesh networks. *ACM Journal of Network and Systems Management*, 18(2):190–209, Jun. 2010.

[181] Y. Al-Hazmi, H. de Meer, K.A. Hummel, H. Meyer, M. Meo, and D. Remondo. Energy-efficient wireless mesh infrastructures. *IEEE Network*, 25(2):32–38, Mar.-Apr. 2011.

[182] Dusit Niyato, Ekram Hossain, and Afshin Fallahi. Solar-powered ofdm wireless mesh networks with sleep management and connection admission control. In *ACM/IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2006.

[183] Marco Ajmone Marsan, Luca Chiaraviglio, Delia Ciullo, and Michela Meo. A simple analytical model for the energy-efficient activation of access points in dense wlans. In *ACM SIGCOMM International Conference on Energy-Efficient Computing and Networking (e-Energy)*, 2010.

[184] Amit P. Jardosh, Konstantina Papagiannaki, Elizabeth M. Belding, Kevin C. Almeroth, Gianluca Iannaccone, and Bapi Vinnakota. Green wlans: On-demand wlan infrastructures. *ACM/Springer Mobile Networks and Applications (MONET)*, 14(6):798–814, Dec. 2009.

[185] Josip Lorincz, Antonio Capone, and Massimo Bogarelli. Energy savings in wireless access networks through optimized network management. In *IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, 2010.

[186] A. Gladisch, R. Daher, P. Lehsten, and D. Tavangarian. Context-aware energy management for energy-self-sufficient network nodes in wireless mesh networks. In *IEEE International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2011.

[187] M.M. Mhlanga, T.O. Olwal, M.B. Mutanga, and M.O. Adigun. Energy optimization based path selection algorithm for ieee 802.11s wireless mesh networks. In *IEEE AFRICON*, 2011.

[188] Antonio Capone, Filippo Malandra, and Brunilde Sansò. Energy savings in wireless mesh networks in a time-variable context. *ACM/Springer Mobile Networks and Applications (MONET)*, 17(2):298–311, Apr. 2012.

[189] Minseok Oh. An adaptive routing algorithm for wireless mesh networks. In *IEEE International Conference on Advanced Communication Technology (ICACT)*, 2008.

[190] Juho Määttä and Timo Bräysy. A novel approach to fair routing in wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking*, 2009(5):1–13, Jan. 2009.

[191] Rami Langar, Nizar Bouabdallah, and Raouf Boutaba. Mobility-aware clustering algorithms with interference constraints in wireless mesh networks. *Elsevier Computer Networks*, 53(1):25–44, Jan. 2009.

[192] Yanfei Fan, Jingqiu Zhang, and Xuemin Shen. Mobility-aware multi-path forwarding scheme for wireless mesh networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2008.

[193] L. Couto, J.P. Barraca, S. Sargento, and R.L. Aguiar. Fastm in wmn: A fast mobility support extension for wireless mesh networks. In *IEEE International Conference on Advances in Mesh Networks (MESH)*, 2009.

[194] Farooq Bari and Victor C.M. Leung. Automated network selection in a heterogeneous wireless network environment. *IEEE Network*, 21(1):34–40, Jan.-Feb. 2007.

[195] Chen Yiping and Yang Yuhang. A new 4g architecture providing multimode terminals always best connected services. *IEEE Wireless Communications*, 14(2):36–41, Apr. 2007.

[196] Qingyang Song and A. Jamalipour. Network selection in an integrated wireless lan and umts environment using mathematical modeling and computing techniques. *IEEE Wireless Communications*, 12(3):42–48, Jun. 2005.

[197] Bo Xing and Nalini Venkatasubramanian. Multi-constraint dynamic access selection in always best connected networks. In *IEEE International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS)*, 2005.

[198] Xuejun Cai, Ling Chen, R. Sofia, and Yanqi Wu. Dynamic and user-centric network selection in heterogeneous networks. In *IEEE International Performance Computing and Communications Conference (IPCCC)*, 2007.

[199] Quoc-Thinh Nguyen-Vuong, Nazim Agoulmine, and Yacine Ghamri-Doudane. A user-centric and context-aware solution to interface management and access network selection in heterogeneous wireless environments. *Elsevier Computer Networks*, 52(18):3358–3372, Dec. 2008.

[200] Jian Tang, Guoliang Xue, and Weiyi Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *ACM International Symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2005.

[201] R. Hincapie, Jian Tang, Guoliang Xue, and R. Bustamante. Qos routing in wireless mesh networks with cognitive radios. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2008.

[202] Baogang Li, Yuanan Liu, Kaiming Liu, and Yuan Liu. Qos-aware routing with rate adaptation in wireless mesh networks. In *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2011.

[203] Thomas Staub and Torsten Braun. Atom: Adaptive transport over multipaths in wireless mesh networks. In *ERCIM Working Group eMobility*, 2008.

[204] Tehuang Liu and Wanjiun Liao. Interference-aware qos routing for multi-rate multi-radio multi-channel ieee 802.11 wireless mesh networks. *IEEE Transactions on Wireless Communications*, 8(1):166–175, Jan. 2009.

[205] Vinod Kone, Sudipto Das, Ben Y. Zhao, and Haitao Zheng. Quorum: quality of service routing in wireless mesh networks. In *ACM International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)*, 2007.

[206] I. Aydin, C. Jaikaeo, and Chien-Chung Shen. Quorum-based match-making services for wireless mesh networks. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2005.

[207] Tao Zhang, Kun Yang, and Hsiao-Hwa Chen. Topology control for service-oriented wireless mesh networks. *IEEE Wireless Communications*, 16(4):64–71, Aug. 2009.

[208] A. Banchs, N. Bayer, David Chieng, A. de la Oliva, B. Gloss, M. Kretschme, S. Murphyk, M. Natkaniec, and F. Zdarsky. Carmen: Delivering carrier grade services over wireless mesh networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2008.

[209] A. Azcorra, T. Banniza, D. Chieng, J. Fitzpatrick, D. Von-Hugo, M. Natkaniec, S. Robitzsch, and F. Zdarsky. Supporting carrier grade services over wireless mesh networks: The approach of the European FP-7 STREP CARMEN. *IEEE Communications Magazine*, 47(4):14–16, Apr. 2009.

[210] Giovanni Resta and Paolo Santi. Wiqosm: An integrated qos-aware mobility and user behavior model for wireless data networks. *IEEE Transactions on Mobile Computing*, 7(2):187–198, Feb. 2008.

[211] Jianyong Chen, Huawang Zeng, Cunying Hu, and Zhen Ji. Optimization between security and delay of quality-of-service. *Elsevier Journal of Network and Computer Applications*, 34(2):603–608, Mar. 2011.

[212] Jong-Tae Park, Jae-Wook Nah, Song Wang, and Seong-Man Chun. Context-aware mobility management with energy efficiency for multimedia streaming service in wireless lan. In *IEEE Consumer Communications and Networking Conference (CCNC)*, 2009.

[213] Phongsak Prasithsangaree and Prashant Krishnamurthy. On a framework for energy-efficient security protocols in wireless networks. *Elsevier Computer Communications*, 27(17):1716–1729, Nov. 2004.

[214] Peizhao Hu, Marius Portmann, Ricky Robinson, and Jadwiga Indulska. Context-aware routing in wireless mesh networks. In *ACM International Conference on Context-awareness for Self-managing Systems (CASEMANS)*, 2008.

[215] International Telecommunication Union. Definition of Quality of Experience (QoE). Technical Report TD 109rev2 (PLEN/12), FOCUS GROUP ON IPTV, Jan. 2007.

[216] T. Heger, T. Monath, and M. Kind. A multi-actor analysis of the qoe environment. In *IEEE International Conference of Telecommunication, Media and Internet Techno-Economics (CTTE)*, 2010.

[217] David Soldani, Man Li, and Renaud Cuny. *QoS and QoE Management in UMTS Cellular Systems*. Wiley, 2006.

[218] Markus Fiedler, Kalevi Kilkki, and Peter Reichl. From quality of service to quality of experience. In *Dagstuhl Seminar*, 2009.

[219] International Telecommunication Union. Methods for Subjective Determination of Transmission Quality. Recommendation P.800, Series P: Telephone Transmission Quality, Aug. 1996.

[220] P. Brooks and B. Hestnes. User measures of quality of experience: why being objective and quantitative is important. *IEEE Network*, 24(2):8–13, Mar.-Apr. 2010.

[221] M. Fiedler, T. Hossfeld, and Phuoc Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2):36–41, Mar.-Apr. 2010.

[222] Wanmin Wu, Ahsan Arefin, Raoul Rivas, Klara Nahrstedt, Renata Sheppard, and Zhenyu Yang. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In *ACM Multimedia (MM)*, 2009.

[223] Junaid Shaikh, Markus Fiedler, and Denis Collange. Quality of experience from user and network perspectives. *Springer Annals of Telecommunications*, 65(1-2):47–57, Feb. 2010.

[224] Manzoor Ahmed Khan and Umar Toseef. User utility function as quality of experience (qoe). In *IEEE International Conference on Networks (ICN)*, 2011.

[225] Rafael Lopes Gomes, Waldir Moreira Junior, Eduardo Cerqueira, and Antônio Jorge Abelém. Using fuzzy link cost and dynamic choice of link quality metrics to achieve qos and qoe in wireless mesh networks. *Elsevier Journal of Network and Computer Applications*, 34(2):506–516, Mar. 2011.

[226] Kuan-Ta Chen, Chi-Jui Chang, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. Quadrant of euphoria: a crowdsourcing platform for qoe assessment. *IEEE Network*, 24(2):28–35, Mar.-Apr. 2010.

[227] M. Volk, J. Sterle, U. Sedlar, and A. Kos. An approach to modeling and control of qoe in next generation networks [next generation telco it architectures]. *IEEE Communications Magazine*, 48(8):126–135, Aug. 2010.

[228] Jingjing Zhang and N. Ansari. On assuring end-to-end qoe in next generation networks: challenges and a possible solution. *IEEE Communications Magazine*, 49(7):185–191, Jul. 2011.

[229] Eileen Dillon, Gemma Power, Manuel Ortiz Ramos, Mara A. Callejo Rodrguez, Jos Rodrguez Argente, Markus Fiedler, and Dario S. Tonesi. Perimeter: A quality of experience framework. In *International Workshop on the Future Internet of Things and Services*, 2009.

[230] Shoaib Khan, Svetoslav Duhovnikov, Eckehard Steinbach, and Wolfgang Kellerer. Mos-based multiuser multiapplication cross-layer optimization for mobile multimedia communication. *Hindawi Journal on Advances in Multimedia*, 2007(1):6–6, Jan. 2007.

[231] Pablo Ameigeiras, Juan J. Ramos-Munoz, Jorge Navarro-Ortiz, Preben Mogensen, and Juan M. Lopez-Soler. Qoe oriented cross-layer design of a resource allocation algorithm in beyond 3g systems. *Elsevier Computer Communications*, 33(5):571–582, Mar. 2010.

[232] R. Serral-Gracia and, E. Marin-Tordera, M. Yannuzzi, X. Masip-Bruin, and S. Sanchez. Quality of experience aware multimedia provisioning with cross-layer network management. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[233] Steven Latré, Pieter Simoens, Bart De Vleeschauwer, Wim Van de Meerssche, Filip De Turck, Bart Dhoedt, Piet Demeester, Steven Van den Berghe, and Edith Gilon de Lumley. An autonomic architecture for optimizing qoe in multimedia access networks. *Elsevier Computer Networks*, 53(10):1587–1602, Jul. 2009.

[234] Rastin Pries, David Hock, Nico Bayer, Matthias Siebert, Dirk Staehle, Veselin Rakocevic, Bangnan Xu, and Phuoc Tran-Gia. Dynamic Bandwidth Control in Wireless Mesh Networks: A Quality of Experience based Approach. In *ITC Specialist Seminar on Quality of Experience*, 2008.

[235] Pejman Goudarzi and Mohammad R. Nezami Ranjbar. Bandwidth allocation for video transmission with differentiated quality of experience over wireless networks. *ACM Computers and Electrical Engineering*, 37(1):75–90, Jan. 2011.

[236] Srisakul Thakolsri, Wolfgang Kellerer, and Eckehard Steinbach. Qoe-based rate adaptation scheme selection for resource-constrained wireless video transmission. In *ACM Multimedia (MM)*, 2010.

[237] A. Khan, L. Sun, E. Jammeh, and E. Ifeachor. Quality of experience-driven adaptation scheme for video applications over wireless networks. *IEEE IET Communications*, 4(11):1337–1347, Jul. 2010.

[238] Kandaraj Piamrat, Adlen Ksentini, Jean-Marie Bonnin, and César Viho. Q-dram: Qoe-based dynamic rate adaptation mechanism for multicast in wireless networks. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2009.

[239] A.B. Reis, J. Chakareski, A. Kassler, and S. Sargento. Distortion optimized multi-service scheduling for next-generation wireless mesh networks. In *IEEE International Conference on Computer Communications (INFOCOM) Workshops*, 2010.

[240] B. Ciubotaru and G.-M. Muntean. Quality of multimedia streaming-oriented handover management solution for mobile applications. In *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2009.

[241] Kandaraj Piamrat, César Viho, Adlen Ksentini, and Jean-Marie Bonnin. QoE-aware network selection in wireless heterogeneous networks. Technical report, May 2010.

[242] Mónica Aguilar Igartua, Luis J. de la Cruz Llopis, Víctor Carrascal Frías, and Emilio Sanvicente Gargallo. A game-theoretic multipath routing for video-streaming services over mobile ad hoc networks. *Elsevier Computer Networks*, 55(13):2985–3000, Sep. 2011.

[243] M.A. Khan, H. Tembine, and S. Marx. Learning in user-centric iptv services selection in heterogeneous wireless networks. In *IEEE International Conference on Computer Communications (INFOCOM) Workshops*, 2011.

[244] M.A. Khan, H. Tembine, and A.V. Vasilakos. Game dynamics and cost of learning in heterogeneous 4g networks. *IEEE Journal on Selected Areas in Communications*, 30(1):198–213, Jan. 2012.

[245] R. Kawamura and R. Stadler. Active distributed management for ip networks. *IEEE Communications Magazine*, 38(4):114–120, Apr. 2000.

[246] A.G. Prieto, D. Gillblad, R. Steinert, and A. Miron. Toward decentralized probabilistic management. *IEEE Communications Magazine*, 49(7):80–86, Jul. 2011.

[247] Ludwig Seitz, Gran Selander, Erik Rissanen, Cao Ling, and Babak Sadighi. Decentralized access control management for network configuration. *Springer Journal of Network and Systems Management*, 16(3):303–316, Sep. 2008.

[248] A.G. Prieto, D. Dudkowski, C. Meirosu, C. Mingardi, G. Nunzi, M. Brunner, and R. Stadler. Decentralized in-network management for the future internet. In *IEEE International Conference on Communications (ICC) Workshops*, 2009.

[249] M. Brunner, D. Dudkowski, C. Mingardi, and G. Nunzi. Probabilistic decentralized network management. In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2009.

[250] M.M. Zavlanos, A. Tahbaz-Salehi, A. Jadbabaie, and G.J. Pappas. Distributed topology control of dynamic networks. In *IEEE American Control Conference*, 2008.

[251] E. Logota, A. Neto, and S. Sargento. A new strategy for efficient decentralized network control. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[252] M. Hauswirth and R. Schmidt. An overlay network for resource discovery in grids. In *IEEE International Workshop on Database and Expert Systems Applications*, 2005.

[253] Amine M. Houyou, Alexander Stenzer, and Hermann Meer. Springer-verlag wireless systems and mobility in next generation internet. chapter Performance Evaluation of Overlay-Based Range Queries in Mobile Systems. 2008.

[254] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys Tutorials*, 7(2):72–93, Second Quarter 2005.

[255] Dilip Joseph, Jayanth Kannan, Ayumu Kubota, Karthik Lakshminarayanan, Ion Stoica, and Klaus Wehrle. Ocala: an architecture for supporting legacy applications over overlays. In *ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2006.

[256] R. Bless, C. Hiibsch, S. Mies, and O.P. Waldhorst. The underlay abstraction in the spontaneous virtual networks (spovnet) architecture. In *IEEE Next Generation Internet Networks (NGI)*, 2008.

[257] J. Rezgui, A. Hafid, and M. Gendreau. Distributed admission control in wireless mesh networks: Models, algorithms, and evaluation. *IEEE Transactions on Vehicular Technology*, 59(3):1459–1473, Mar. 2010.

[258] S. Schuetz, K. Zimmermann, G. Nunzi, S. Schmid, and M. Brunner. Autonomic and decentralized management of wireless access networks. *IEEE Transactions on Network and Service Management*, 4(2):96–106, Sep. 2007.

[259] C. Canali, M.E. Renda, P. Santi, and S. Burresi. Enabling efficient peer-to-peer resource sharing in wireless mesh networks. *IEEE Transactions on Mobile Computing*, 9(3):333–347, Mar. 2010.

[260] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, Feb. 2003.

[261] Hyunok Lee and D.C. Cox. A fully-distributed control time slot assignment protocol for large wireless mesh networks. In *IEEE Military Communications Conference*, 2008.

[262] Ho Ting Cheng, Hai Jiang, and Weihua Zhuang. Distributed medium access control for wireless mesh networks: Research articles. *Wiley Wireless Communications and Mobile Computing*, 6(6):845–864, Sep. 2006.

[263] Shweta Jain, Samir R. Das, and Himanshu Gupta. Distributed protocols for scheduling and rate control to achieve max-min fairness in wireless mesh networks. In *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2007.

[264] Hsien-Po Shiang and M. van der Schaar. Multi-user video streaming over multi-hop wireless networks: A cross-layer priority queuing scheme. In *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, 2006.

[265] F. O. Aron, T. O. Olwal, A. Kurien, and M. O. Odhiambo. A distributed topology control algorithm to conserve energy in heterogeneous wireless mesh networks. In *IEEE International Conference on Information Communication Technologies (ICICTE)*, 2008.

[266] M. D. Di Benedetto, A. D'Innocenzo, and E. Serra. Dynamical power optimization by decentralized routing control in multi-hop wireless control networks. *IFAC World Congress*, 2011.

[267] B. Mathieu, Meng Song, A. Galis, L. Cheng, K. Jean, R. Ocampo, M. Brunner, M. Stiemerling, and M. Cassini. Self-management of context-aware overlay ambient networks. In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2007.

[268] AN. Ambient networks. Information available at http://www.ambient-networks.org/.

[269] I. Al-Oqily and A. Karmouch. Policy-based context-aware overlay networks. In *IEEE International Global Information Infrastructure Symposium (GIIS)*, 2007.

[270] AUTOI. Autonomic internet. Information available at http://www.ist-autoi.eu/.

[271] Lawrence Cheng, Alex Galis, Bertrand Mathieu, Kerry Jean, Roel Ocampo, Lefteris Mamatas, Javier Rubio-Loyola, Joan Serrat, Andreas Berl, Hermann de Meer, Steven Davy, Zeinab Movahedi, and Laurent Lefevre. Self-organising management overlays for future internet services. In *Modelling Autonomic Communications Environments*, volume 5276, pages 74–89. 2008.

[272] Seung-Ik Lee, Jong-Hwa Yi, and Shin-Gak Kang. A context-aware service composition scheme for service overlay network. In *IEEE nternational Conference on ICT Convergence (ICTC)*, 2011.

[273] T. Gu, H.K. Pung, and D. Zhang. A peer-to-peer overlay for context information search. In *IEEE International Conference on Computer Communications and Networks (ICCCN)*, 2005.

[274] H.T. Shen, Y. Shu, and B. Yu. Efficient semantic-based content search in p2p network. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):813–826, Jul. 2004.

[275] B. Sapkota, S. Nazir, T. Vitvar, I. Toma, L. Vasiliu, and M. Hauswirth. Semantic overlay for scalable service discovery. In *IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2007.

[276] Christos Doulkeridis, Kjetil Norvag, and Michalis Vazirgiannis. Desent: Decentralized and distributed semantic overlay generation in p2p networks. *IEEE Journal on Selected Areas in Communications*, 25(1):25–34, Jan. 2007.

[277] A. Loser, S. Staab, and C. Tempich. Semantic social overlay networks. *IEEE Journal on Selected Areas in Communications*, 25(1):5–14, Jan. 2007.

[278] H. Kubo, R. Shinkuma, and T. Takahashi. Topology control using multi-dimensional context parameters for mobile p2p networks. In *IEEE Vehicular Technology Conference (VTC)-Spring*, 2010.

[279] N. Bisnik and A. Abouzeid. Delay and Throughput in Random Access Wireless Mesh Networks. In *IEEE International Conference on Communications (ICC)*, 2006.

[280] Gunter Bolch, Stefan Greiner, Hermann De Meer, and Kishor S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation With Computer Science Applications, Chapter 10.* John Wiley and Sons, 1998.

[281] The Network Simulator NS-2. Information available at http://www.isi.edu/nsnam/ns/.

[282] BRITE (Boston University Representative Internet Topology Generator). Information available at http://www.cs.bu.edu/brite/.

[283] NS-2 Multi-Interface Support. Information available at http://personales.unican.es/aguerocr/.

[284] An IEEE 802.11e EDCA and CFB Simulation Model for NS-2. Information available at http://www.tkn.tu-berlin.de/research/802.11e_ns2/.

[285] The Bamboo Distributed Hash Table. Information available at http://bamboo-dht.org/.

[286] J. Loughney, M. Nakhjiri, C. Perkins, and R. Koodli. Context Transfer Protocol (CXTP). In *RFC 4067*, Jul. 2005.

[287] Chord Implementation. Information available at http://open-chord.sourceforge.net/.

[288] R. Hancock, G. Karagiannis, J. Loughney, and S. Van den Bosch. Next Steps in Signaling (NSIS): Framework. In *RFC 4080*, Jun. 2005.

[289] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter Base Protocol. In *RFC 3588*, Sep. 2003.

[290] X. Fu, H. Schulzrinne, A. Bader, D. Hogrefe, C. Kappler, G. Karagiannis, H. Tschofenig, and S. Van den Bosch. Nsis: A new extensible ip signaling protocol suite. *IEEE Communications Magazine*, 43(10):133–141, Oct. 2005.

[291] H. Schuzrinne and R. Hancock. GIMPS: General Internet Messaging Protocol for Signaling. In *Internet draft, work in progress*, Jul. 2005.

[292] J. Manner, G. Karagiannis, and A. McDonald. NSLP for Quality-of-Service Signaling. In *RFC 5974*, Oct. 2010.

[293] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, Mar. 2000.

[294] ITU-T. The E-model, a computational model for use in transmission planning, Feb. 2003. Information available at http://www.itu.int/itudoc/itu-t/aap/sg12aap/history/g107/g107.html.

[295] Video Quality Experts Group. Final Report on the Validation of Objective Models of Video Quality Assessment, Aug. 2003. Information available at http://www.its.bldrdoc.gov/vqeg/.

[296] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, Jan.-Feb. 1997.

[297] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, Sep. 2001.

[298] IEEE 802.11e 2005. Information available at http://awin.cs.ccu.edu.tw/paper_file/802.11e-2005.pdf.

[299] Roberto Barbieri, Danilo Bruschi, and Emilia Rosti. Voice over ipsec: Analysis and solutions. In *IEEE Annual Computer Security Applications Conference (ACSAC)*, 2002.

[300] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, England, 1989.

[301] Shailesh Kumar and Risto Miikkulainen. Dual reinforcement q-routing: An on-line adaptive routing algorithm. In *Artificial Neural Networks in Engineering (ANNIE)*, 1997.

[302] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 1998.

[303] Video Traces for Network Performance Evaluation, Oct. 2008. Information available at http://trace.eas.asu.edu/tracemain.html.

[304] L. E. Miller. Connectivity Properties of Mesh and Ring/Mesh Networks. *Information Available at http://w3.antd.nist.gov/wctg/netanal/MeshCon.pdf*, Apr. 2001.