Traffic Management in Multipath Wireless Networks

Oscar Delgado

Department of Electrical & Computer Engineering
McGill University, Montreal, Canada

October 2016

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Wireless mobile technologies have experienced a rapid growth during the last decades. Today, mobile communications provide access and connectivity to billions of people around the world. With a steadily increasing demand for high-bandwidth multimedia applications, such as live and on-demand video streaming, virtual reality and yet-to-be-imagined applications, the need for the development of new wireless technologies becomes imperative.

Although not standardized yet the Fifth Generation (5G) wireless technology promises to offer reliable, low-latency and high-density networks capable of supporting High-Definition (HD) video. Some of the supporting technologies that could enable dynamic management of network resources and could be the key components in the design of 5G wireless networks are network virtualization and Software-Defined Networking (SDN).

Video and in general multimedia traffic management over SDN networks is an important emerging research area where innovative solutions will be required to meet the challenging 5G performance requirements. An important paradigm shift that has to be considered is that 5G will be based on the user-centric concept instead of operator-centric or service-centric concept as in previous wireless generations. Hence, it would be possible for multiple incoming flows from different radio access technologies to be combined at the mobile device.

In this thesis, we first develop a flow management framework, called Joint Slice Manager (JSM), over wireless networks. While the problem of video traffic management has received significant attention in recent years, to the best of our knowledge, this is the first work that designs an SDN-based solution over wireless networks considering that mobile devices can be connected simultaneously to multiple radio nodes of the same technology. One of the benefits of building JSM over the SDN framework is that it enables us to extend JSM over other wireless technologies like 5G. JSM does not require any modifications to the mobile device (client) or the server, facilitating a quicker implementation. Our simulation results show significant improvements not only in video quality, but also in the radio nodes' utilization.

In order to deal with the network capacity constraints derived from the expected demand of video traffic mentioned above, we propose new load balancing and Device-to-Device (D2D) relaying strategies that make better use of the network resources and increases the throughput respectively.

We develop three multipath load balancing algorithms that deal with the problem of

how to split traffic such that the traffic load remains at the desired level. The performance evaluation focuses on uplink wireless networks transmitting User Data Protocol (UDP) video traffic. Our simulation results show significant improvements over state-of-the-art algorithms not only in performance indicators like the splitting error and power consumption, but also in Peak Signal-to-Noise Ratio (PSNR).

We also develop a method to improve mobile device throughput by devising the use of mobile devices with multiple network interfaces that act as full duplex D2D relays. We model the problem as an optimization problem and propose a family of heuristic methods for selecting the relays. Our simulation results indicate that using mobile devices as relay not only helps to improve the total throughput, but also to improve fairness (throughput at the cell edge) while significantly reducing the transmission power consumption.

Résumé

Les technologies mobiles sans fil ont connu une croissance rapide au cours des dernières décennies. Aujourd'hui, les communications mobiles permettent l'accès et la connectivité entre des milliards de personnes à travers le monde. Avec une demande sans cesse croissante pour les applications multimédia à haut débit, telles que le streaming vidéo (en direct et à la demande), la réalité virtuelle et les applications futures, il devient alors impératif de développer des nouvelles technologies sans fil.

Même si elle n'est pas encore standardisée, la cinquième génération (5G) des technologies sans fil promet d'offrir des réseaux fiables, à faible latence et à haute densité, qui sont capables de supporter de la vidéo à haute définition (HD). Parmi les technologies qui pourraient permettre une gestion dynamique des ressources du réseau et qui pourraient être l'élments clés dans la conception des réseaux sans fil 5G entrent en scène la virtualisation et le réseautage logiciel (software-defined networking (SDN)).

La vidéo et la gestion du trafic multimédia sur les réseaux SDN en général, est un domaine de recherche émergeant où des solutions innovantes seront nécessaires pour répondre aux exigences strictes de performance des réseaux 5G. La 5G devrait prendre en considération un changement de paradigme important basé sur l'utilisateur à la place de l'opérateur ou des services tel qu'il est le cas des générations sans fil précédentes. Par conséquent, il serait possible pour les multiples flux entrants provenant de différentes technologies d'accès radio d'être combinés au niveau de l'appareil mobile.

Dans cette thèse, nous développons d'abord une plateforme de gestion de flux appelée Joint Slice Manager (JSM) dans les réseaux sans fil. Bien que le problème de la gestion du trafic vidéo ait connu beaucoup d'attention ces dernières années, au mieux de notre connaissance, ce travail est le premier qui conçoive une solution de type SDN pour les réseaux sans fil étant donné que les appareils mobiles peuvent être connectés simultanément à plusieurs stations de base avec la même technologie. L'un des avantages de l'implémentation de JSM en utilisant SDN est qu'elle permet d'étendre JSM á d'autres technologies sans fil comme la 5G. JSM ne nécessite aucune modification de l'appareil mobile (client) ou du serveur, ce qui rend son implémentation plus facile et plus rapide. Nos résultats de simulations montrent d'importantes améliorations au niveau de la qualité de la vidéo, mais aussi au niveau de l'utilisation de la station de base.

Afin de faire face aux contraintes de la capacité du réseau provenant de la forte demande

du trafic vidéo mentionnée ci-dessus, nous proposons de nouvelles stratégies d'équilibrage de charge et de relais appareil-à-appareil (Device-to-device (D2D)), ce qui permet une meilleure utilisation des ressources du réseau et une l'augmentation du débit respectivement.

Nous développons trois algorithmes d'équilibrage de charge multi-trajects qui traitent du probléme de la répartition du trafic, de telle sorte que la charge du trafic reste au niveau désiré. L'évaluation du rendement se concentre sur la liaison en amont des réseaux sans fil transmettant un trafic vidéo de type UDP (User Data Protocol). Nos résultats de simulations montrent des améliorations significatives par rapport aux algorithmes dits classiques, non seulement au niveau des indicateurs de performance, comme l'erreur de fractionnement et la consommation d'énergie, mais aussi au niveau du rapport signal-sur-bruit du pic (Peak Signal-to-Noise Ratio (PSNR)).

Nous développons également une méthode pour améliorer le débit de l'appareil mobile en utilisant plusieurs interfaces réseau qui agissent comme relais D2D bidirectionnels simultanés (full duplex). Nous modélisons le problème comme étant un problème d'optimisation et nous proposons plusieurs méthodes heuristiques pour sélectionner les relais. Nos résultats de simulations indiquent que l'utilisation d'appareils mobiles comme relais permet non seulement d'améliorer le débit total, mais permet également d'améliorer l'équité (débit au bord de la cellule), tout en réduisant de manière significative la consommation d'énergie de la transmission.

Acknowledgments

Pursuing a doctoral degree is a long and mostly lonely journey that in many ways shapes your life. No matter how hard you try reaching this point would not have been possible without God, and the help, support and inspiration of amazing colleagues and friends I met along the way.

Firstly, I would like to express my sincere gratitude to my thesis supervisor Prof. Fabrice Labeau for opening the door to a dynamic research lab and for his continuous support during my PhD study, as well as for his patience, motivation and expertise. His positive guidance and availability to answer my questions helped me during the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my PhD study.

A special thanks goes out to Dr. Brigitte Jaumard, without whose understanding, motivation and kindness I would not have reached this far. Dr. Jaumard is the one professor who truly believed in me. I doubt that I will ever be able to convey my appreciation fully, but I owe her my eternal gratitude.

I would like to thank my colleagues in the lab, both past and present, for the stimulating conversations and companionship over the years. Your support, both scientifically and personally was fundamental in supporting me during stressful and difficult moments. A huge thank-you goes to all of my wonderful friends who have always believed in me. I really do not think I could have survived without all of your support. I am especially grateful to my colleagues Santosh, Marwen and Fabien, and to my friends Julio, Claudia and Parag who always stood by me during my restless years at McGill.

Undoubtedly, this journey would not have been possible without the support of my family. The acknowledgements presented here are only a blurry reflection of how much I love you all. To my mother Magdalena, I admire your ability to hold steadfast to your beliefs and for being the exemplar of integrity, honesty and patience, thanks for instilling in me the importance of these principles in life. To my father Humberto, I thank you for your optimism, hard work and strong will, and because you teach me to fight for what I want and to never give up. To my sisters Yenny and Cathia, for cheering me up and for offering to help in all the ways you knew possible. I will always cherish the magic and mischief we shared which keeps the child in me alive. Together you have helped to build the foundation on which I stand today.

Preface

All of the work presented henceforth was conducted in the Telecommunications & Signal Processing Laboratory in the Department of Electrical and Computer Engineering at McGill University under the PhD supervision of Professor Fabrice Labeau.

A version of the material in Chapter 3 has been filed for a U.S. patent (PCT international application number 14/516,394 filed on 10/16/2014). I was responsible for all major areas of concept formation, implementation, testing and data analysis, as well as the majority of manuscript composition. The implementation of the NS-3 network simulator was performed by Marwen Bouanen and I, with assistance from many other collaborators. Marwen Bouanen (McGill University) was involved in the network simulator design, testing, concept formation and contributed to manuscript composition. Fabrice Labeau was the supervisory author on this project and was involved throughout the project in concept formation and manuscript edits. Alex Stephenne and Ngoc Dung Dao (Huawei Technologies Canada Co., Ltd.) were involved throughout the project in concept formation.

A version of Chapter 4 and 5 has been published or submitted for publication. I was the lead investigator, responsible for all major areas of concept formation, simulation implementation, experimental setup, data collection and analysis, as well as the majority of manuscript composition. Fabrice Labeau was the supervisory author and was involved throughout the project in concept formation and manuscript edits.

A detailed description of the contributions of this thesis is presented in the Introductory Chapter in Section 1.2, along with the list of related publications.

Contents

List of Figures

List of Tables

List of Acronyms

ASN-GW	Access Service Network Gateway
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CDF	Cumulative Density Function
D2D	Device-to-Device
eNodeB	Radio node in LTE
FDM	Frequency-Division Multiplexing
fps	Frames per second
FLARE	Flowlet Aware Routing Engine
GOP	Group of pictures
HTTP	Hypertext Transfer Protocol
HSS	Home Subscriber Server
i.i.d.	independent and identically distributed
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPTV	Internet Protocol Television
ITU	International Telecommunication Union
LTE	Long-Term Evolution
MAC	Media Access Control
MPEG	Moving Picture Experts Group
MSE	Mean-Square Error
MME	Mobility Management Entity
NFC	Near Field Communication
OTT	Over-The-Top content

PCRF	Policy Control and Charging Rules Function
PDF	Probability Density Function
P-GW	Packet data network Gateway
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RAT	Radio Access Technology
SDN	Software Defined Network
SDM	Space-Division Multiplexing
S-GW	Serving Gateway
SINR	Signal to Interference Noise Ratio
SLA	Service Level Agreement
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
TDM	Time-Division Multiplexing
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
VoIP	Voice over IP
VBR	Variable Bit Rate
Wi-Fi	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
3G	Third Generation cellular systems
3GPP	3rd Generation Partnership Project
4G	Fourth Generation cellular systems
5G	Fifth Generation cellular systems

Chapter 1

Introduction

1.1 Motivation

Wireless mobile networks have changed the way we communicate over the last few decades. From analog to digital communications, wireless technologies continue to evolve leading to increased interconnectivity between mobile devices and by extension individuals. Recently, the concept of the Fifth Generation cellular network (5G) has promised to satisfy the growing needs of mobile wireless communication. The most common features associated with 5G are increased data rate, larger number of users, improved quality of service and extended coverage of mobile networks. Not to mention the need for low latency and high reliability that are important for many services such as self-driving cars or telemedicine. Although, the 5G network has not been standardized yet, the International Telecommunication Union (ITU) plans that the final specifications of the 5G network will be issued in 2020 [1].

The exponential growth of mobile applications and rapid adoption of mobile connectivity by end users combined with the new services offered by service providers has fueled the need for faster connectivity, optimized bandwidth management and higher security. Considering mobile data projections, we can see that, while, the global mobile data traffic will grow at a compound annual grow rate of 53% from 2015 to 2020, three-fourths of the world's mobile data traffic will be video by 2020 [2]. Moreover, users always require the emergence of new services and applications.

To satisfy the above mentioned needs many new technologies are emerging. One of such new technologies is Software-defined Networking (SDN). The main characteristic of SDN is its ability to decouple the control plane from the data plane. Moving the control plane to

a centralized SDN controller has the advantage that the controller has global knowledge of the network, making it easier to reconfigure the network, which results in enhanced network flexibility [3].

Given that most of the traffic will be video traffic, we have to consider how to pre-provision resources such that video flows do not over or underutilize network resources, as well as how to deal with channel capacity fluctuations, mainly due to interference, user mobility, etc. Similarly, since video rates and lengths are variable, the question of how the network can be more tolerant to such fluctuations is still open. In this thesis, we specifically use SDN as an enabling technology that can be utilized to transport and deliver video traffic.

Some of the most important differences between 5G and previous wireless mobile networks are heterogeneous network architecture and device-to-device (D2D) communication. The heterogeneous network architecture, aims to integrate into one network several wireless networks with possibly different radio access technologies (RATs). Opening the opportunity for mobile devices to be connected to multiple RATs simultaneously, has the potential to offer better coverage and greater flexibility in satisfying different quality of service (QoS) requirements which leads to seamless handover between RATs [4].

Integration of multiple RATs into one network is probably the hardest challenge for 5G, as today's technologies are incompatible. As a first step to understanding the challenges of mobile devices connected to multiple RATs we have limited this thesis to the case in which mobile devices are connected to multiple radio nodes of the same technology.

The advantages of multiple paths are no longer restricted to avoiding single points of failure, but also aim to increase network capacity and guarantee high quality of service. Right now, as any smart-phone owner might know, a phone or tablet can be connected to multiple wireless networks, but it is not possible to use those multiple connections at the same time for the same service. For example, in the case of video streaming, the video may cut out because the network being used drops, even though there is another network available.

In this example and countless other real world applications, an efficient use of network resources is needed. Consequently the ability to distribute traffic across multiple network resources (load balancing) is one of the solutions that can be applied to maximize network capacity. Load balancing can bring benefits in the form of high throughput, increased reliability, and fair use of resources. However, there are some challenges that need to be addressed in order to efficiently reap all of the theoretical benefits of load balancing in

practical applications. The primary challenge is to find an optimal way to switch packets between multiple paths [5], e.g., to minimize packet reordering or to make a smart wireless interface selection such that energy is not wasted (the battery life of a mobile device is related to how much energy is required to transmit a packet).

An important metric to measure the load balancing is the splitting error. Splitting error measures the deviation of the desired load from the actual load in each path. The lower the splitting error is, the better load balancing we have. One of the major problems when switching packets between multiple paths is the occurrence of packet reordering. Packet reordering is caused by differences in transmission delays among multiple paths. If a packet is delivered along a faster path, it may arrive at the destination before the previously transmitted packet, which is delivered along a slower path [6].

As mentioned earlier, D2D communication is one the new features of 5G. D2D communication allows mobile devices to communicate directly between each other without passing data through the radio node (except possibly control signals). Consequently relaying using D2D could be enabled, which could result in offloading traffic from the radio nodes and saving power consumption [7].

1.2 Contributions

The first contribution of this thesis is the design of a Joint Slice Manager (JSM) framework for video traffic management in wireless mobile networks. This is presented in Chapter 3. The second contribution, as described in Chapter 4, is the development of a series of low complexity algorithms that addresses the issue of how to split traffic such that the traffic load remain balanced, while trying to improve the basic QoS requirements. The third contribution of this thesis work is the analysis of the possible benefits of relaying when using D2D communications, which is presented in Chapter 5. We summarize below the major contributions of the work described in this thesis:

1.2.1 Joint Slice Manager Framework

- In the control plane, the flow manager is configured to classify the packet flow, serviced by more than one radio node, to a specific slice according to its nature (short or long), the available capacity of each slice, and the feedback information provided by the radio

nodes. It also has the ability to reclassify the packet flow according to the traffic load, and the feedback information provided by the radio nodes.

- The data plane is operatively coupled to the control plane, the data plane is configured to store the packets from the flows according to the classification information provided by the control plane.

- Our network contribution includes the design of the first traffic management framework for wireless networks with the capability to give service to mobile devices connected simultaneously to more than one radio node of the same technology.

1.2.2 Uplink Load Balancing Algorithms

- Our first load balancing algorithm QBALAN shows that highly accurate load balancing can be achieved by subdividing the load balancing task into long-term and short-term load balancing algorithms without adding complexity. This strategy effectively reduces the splitting error by reducing the average number of times a flow switch paths, while not significantly affecting the packet reordering.

- The second load balancing algorithm DALBA accurately splits traffic while reducing the average packet loss and the average end-to-end delay. It is robust to different traffic loads and it exhibits superior PSNR performance in high traffic load conditions. Additionally, DALBA outperforms previous algorithms by reducing the total end-to-end delay and packet reordering.

- The third load balancing algorithm GEL manages to reduce the mobile devices' power consumption while maintaining acceptable QoS levels. It also demonstrates that there is a trade-off between power consumption and splitting error that should be considered when evaluating a load balancing strategy.

1.2.3 Relaying Using Device-to-device Communications

- We propose an infrastructure based solution (controlled by the network operator) that enables relaying capabilities on mobile devices with multiple network interfaces, due to this characteristic we propose that mobile devices act as full duplex relays

(in the context of this thesis full duplex refers to the ability to use one interface to transmit and one interface to receive data at the same time).

- We define an objective optimization problem that maximizes the total network capacity, considering that connections can have at most one relay (two hops).
- We propose a sub-optimal family of greedy algorithms, that exhibit good performance in terms of total capacity, average transmission power, and user throughput fairness (increased throughput at the cell edges).

1.2.4 List of Related Publications

This thesis has led to the following publications

- Marwen Bouanen, Oscar Delgado, Fabrice Labeau, Alex Stephenne and Ngoc Dung Dao, "System and Method for Transmission Management in Software Defined Networks", *A United States Patent application has been submitted*, PCT international application number 14/516,394 filed on 10/16/2014.
- Oscar Delgado, and Fabrice Labeau, "Uplink load balancing over multipath heterogeneous wireless networks", *IEEE Vehicular Technology Conference (VTC)*, pp. 686-691, May 2015.
- Oscar Delgado and Fabrice Labeau, "Delay aware load balancing over multipath wireless networks", submitted to *IEEE Transactions on Vehicular Technology (TVT)*, 2015.
- Oscar Delgado and Fabrice Labeau, "Uplink energy-efficient load balancing over multipath wireless networks", *IEEE Wireless Communication Letters (WCL)*, pp. 424-427, August 2016.
- Oscar Delgado and Fabrice Labeau, "D2D relay selection and fairness on 5G wireless networks", accepted for publication in *IEEE Globecom workshops*, December 2016.

1.3 Organization of the Thesis

This chapter has presented a brief introduction to this thesis work. The remainder of this thesis is organized as follows. Chapter 2 provides the necessary background for the

remainder of the thesis. The purpose is to provide a review on existing software defined network solutions that can be used to handle video traffic in wireless cellular networks, and to highlight the fact that video content is one of the key players when developing the next generation of wireless communication systems. The background emphasizes the use of mobile devices with multiple interfaces that are simultaneously connected to multiple radio nodes, as an alternative to improve capacity and quality of service. We also review the problem of how to split traffic such that the load remains balanced, and study new solutions that can help us increase channel capacity.

A novel video traffic management framework (Joint Slice Manager) developed under the software defined network paradigm is presented in Chapter 3. The purpose of our proposed solution is to improve video quality in wireless cellular networks.

Chapter 4 presents the work done in investigating uplink load balancing for addressing the problem of how to split traffic without reducing the quality of service.

Chapter 5 presents device-to-device communication as an approach to increase channel capacity and to save power consumption. Finally, Chapter 6 concludes the thesis and presents possible topics of future work.

Chapter 2

Background: SDN and Wireless Communications

2.1 Introduction

The constant development of new technologies has created a wireless environment in which multiple standards are coexisting. In such heterogeneous environment, the interoperability and the resource allocation among different technologies are some of the potential issues to solve. To address both current and future needs in a wireless environment, the abstraction provided through wireless virtualization is one possible solution to both simplify and unify wireless networks [8]. Combined with SDN [9], virtualization allows to enhance resource utilization, and to provide value-added service differentiation.

SDN is an approach that allows network operators to manage network services through abstraction of lower level functionality. This is done by decoupling the control plane from the data plane, i.e., the system that makes decisions about where traffic is sent from the underlying systems that forwards traffic to the selected destination [10].

For example, having multiple network interfaces offers the possibility to have parallel connections through multiple paths to the destination. In this context, an efficient use of network resources is needed; load balancing and mobile relaying models are some of the solutions that can be applied to maximize network capacity.

In this chapter, we present the existing wireless technologies that can be used to transport and deliver video traffic, we describe the wireless SDN solutions that can be applied,

and we explain their main advantages and disadvantages. We also study load balancing strategies that help to maximize the overall network capacity. Finally, we present existing mobile relaying models and their characteristics.

2.2 LTE Architecture

LTE stands for Long Term Evolution and is a wireless communication standard for mobile phones and data terminals maintained by the 3rd Generation Partnership Project (3GPP). Currently LTE is marketed as the fourth generation (4G) wireless service (sometimes called 4G LTE), the 3GPP consortium has adopted the latest version of the standard (LTE Advanced standard).

The LTE cellular network architecture can be described as: User Equipments (UEs) served by radio nodes (eNodeB), which are typically connected to a local anchor (Serving Gateway S-GW) that routes and forwards user data packets; another gateway (Packet Gateway P-GW) connects the S-GW to the Internet, and typically also performs policy enforcement, and packet screening; the Mobility Management Entity (MME) controls the high-level operation of the mobile device and it is linked to the Home Subscriber Server (HSS), which is the database containing all the subscriber's information. It also has a Policy Control and Charging Rules Function (PCRF) that is responsible for policy decision and flow-based charging functionalities, see Fig. 2.1.

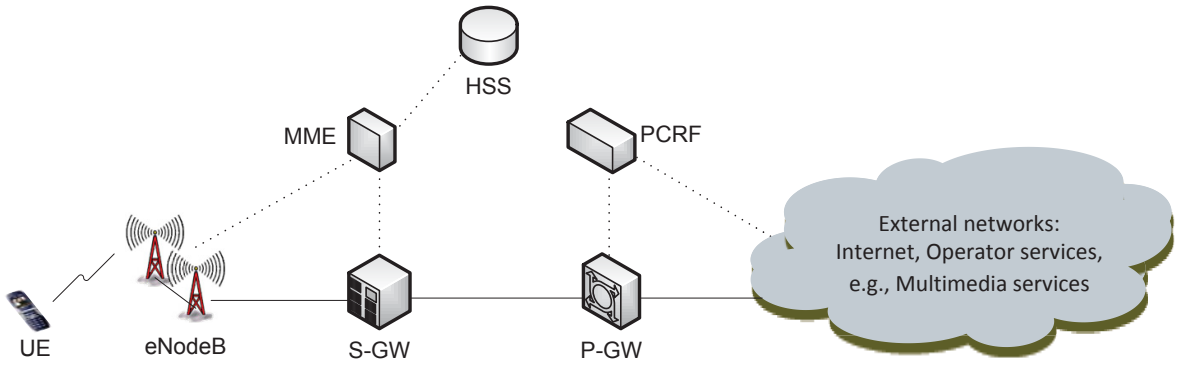


Fig. 2.1 LTE architecture: basic network elements.

A comprehensive survey on LTE can be found in [11]. In general, one of the known concerns of the LTE architecture is the scalability related to the centralization of multiple functionalities in the serving or packet gateways. Furthermore, the centralization of all the data plane functionalities at the interface between LTE and internet domains can also be a limitation for in-cellular-network traffic [12, 13].

2.3 Software-defined Networking

SDN is an architecture that centralizes the intelligence of the network, allowing network operators to manage applications and network services through abstraction. This abstraction is done by decoupling the control plane (where decisions are taken) from the data plane (where the actual traffic is forwarded) [3].

The SDN architecture can support virtualization, allowing to isolate resources across user groups, and/or services (see Fig. 2.2); enabling several network operators to coexist on the same hardware (virtual network operators). In that sense SDN can be a potential technology to enable actual and future services.

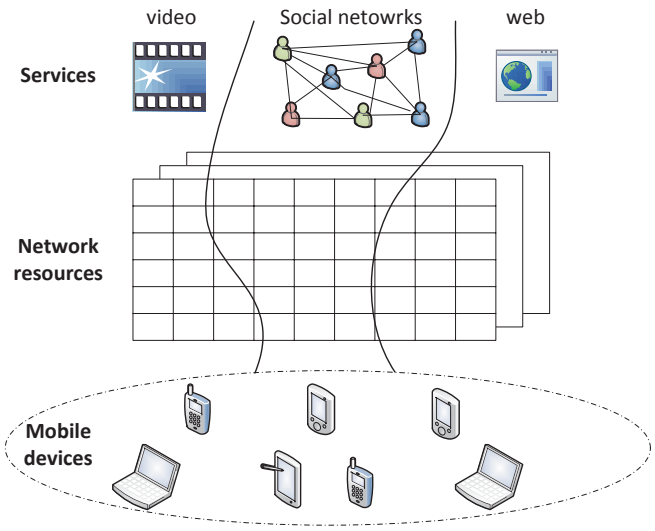


Fig. 2.2 Virtualization scenario: resources isolated across services

It is important to remark here that network virtualization is not the same as SDN. Although by definition SDN provides a level of abstraction and decouples functionalities

from the hardware, that can enable virtualization, SDN by itself does not necessarily imply sharing or isolating resources.

2.3.1 Wireless SDN

A brief overview on the current development of wireless software-defined networks can be found in [14]. Most of the prototype implementations on cellular-type Wireless SDN has been conducted in WiMAX [15–20], mainly because of its rapid implementation due to the availability of affordable PicoChip femtocell WiMAX base stations that can be used for experimental testing.

Some other related work in wireless SDN can be found on WLAN [21–24] and WPAN [25–27], but they have slightly different issues. Compared to cellular networks, WLAN networks are easier to deploy. However the control functions are relatively less sophisticated, having only support for priority-based QoS, but not for scheduling, making a resource reservation difficult to achieve. WPAN networks use some form of traffic aggregation in the nodes, and treat their traffic differently, which does not directly relate to a cellular scenario. We can also find some research work on wireless SDN that does not focus on mobile core networks, but instead targets heterogeneous mobile networks, i.e., to coordinate multiple radio access technologies [28].

2.3.1.1 Initial Wireless SDN Efforts

Some of the first proposals in wireless SDN were motivated by the need to run several experiments simultaneously in the same test bed, each within its own network slice [29,30], thus the need to enable network virtualization. Due to the shared nature of the wireless access medium, one of the main challenges was the difficulty to isolate network resources. To address this problem several basic virtualization schemes were proposed. The main idea was to share access to the wireless medium in the time domain (TDM), the frequency domain (FDM), the spatial domain (SDM, different slices occupying disjoint physical regions of the wireless network, apart enough so that they do not introduce interference to one another), or even through frequency hopping.

Among the practical difficulties that have prevented the deployment of these techniques within wireless network test beds, the switching capabilities of the deployed equipment is one of the most important. This is because it sets the limit to the speed at which the slices

can be scheduled, e.g., from the wireless hardware point of view, it limits the switching speed for frequency hopping, or in the CPUs operating the network, the operating system's kernel imposes a speed limitation. From an experimental perspective, another challenge is the difficulty of controlling and reproducing propagation characteristics inherent to wireless channels, so that repeating exactly the same experiments is also a concern.

The first efforts describe what can be viewed as an idealized perspective of wireless SDN, which targets a perfect isolation of resources in an uncontrolled wireless environment. More recent wireless SDN proposals described below do not aspire for a perfect network isolation, but for a less strict setting, thus defining slicing as a constrained resource reservation and allocation problem. It is important to mention that, due to the motivation of their development, these early proposals described above are not well suited for practical wireless scenarios which are naturally affected by interference; mainly because all traffic slices should coexist to some extent to efficiently use all the available wireless resources, contrary to the experimental wireless network test beds, where interference between traffics from different slices must be avoided to guarantee isolation.

2.3.1.2 Wireless Network Virtualization

In general, wireless network virtualization refers to the ability to share and abstract wireless access resources among multiple mobile devices or groups of mobile devices with a certain degree of isolation. In this context virtualization of LTE [13, 31, 32], WiMAX [16, 33] and Wireless Local Area Network (Wi-Fi) [21, 23] has been studied to some degree. Although few researchers have proposed a unified framework for virtualization of these three technologies [28], almost all of them take OpenFlow [34] as the main starting point.

Since the SDN concept was first developed, the idea of decoupling the data plane and the control plane has been studied in LTE. Two of the works that provide LTE network level solutions are found in [13] and [35]. In [35], the eNodeB has been sliced into virtual eNodeBs by using the FlowVisor policy described in [36], where the strategy is to create the same number of controllers as virtual eNodeBs, In the same way, [13] extends the work of [36] to CellVisor, which provides a wider range of resource slicing, including topology, bandwidth, forwarding tables, etc.

One of the few examples, that explicitly attempts to use the SDN concept and abstracts from the technology used by the radio access nodes, is the OpenRoads test bed at

Stanford [19, 20]. The OpenRoads test bed is an open platform for innovation under the proposed multi-radio access architecture; thus no specific set of algorithms are directly proposed. Nevertheless, it must be mentioned that some experimental results are available for handover scenarios with time overlap, as well as for the so-called n-casting (packet transfer through multiple redundant routes of different technologies) [37].

2.3.2 SDN Challenges

In a production network, the SDN goal is not only to enhance resource utilization but also to provide value-added service differentiation. Thus SDN can be defined as a resource reservation and allocation problem, constrained on some performance metrics, such as QoS or quality of experience (QoE). The objective of using these metrics is to reflect either the network management goals or the Service Level Agreements (SLA) fixed by the network operator and the customers.

Given the LTE architecture depicted in Fig. 2.1, we can argue that the first challenge is to determine which portion of the network have to be virtualized in order to efficiently provide SDN services. Intuitively, in the cellular architecture, the radio node is the place where to operate slicing, since typically the radio node is in charge of allocating radio resources to flows and mobile devices, and handles uplink scheduling. It is then logical to consider that the radio node itself could handle virtualization and assign physical resources to each slice so as to meet the performance constraints [16, 18].

There are, however solid arguments for not fully adopting radio nodes as the main place where to deploy SDN [12, 13, 33]. Having too many functions situated at the edge of the wireless network could lead to scalability issues. Also the high level information, such as handovers, traffic load of other radio nodes, billing, etc, is not readily available at the radio node. Additionally, we should mention that due to the numerous radio node suppliers, it will be much easier to enter into the market if it is done independently of the radio node (integration issues).

Numerous researchers are currently proposing to deploy SDN virtualization one step away from the radio node, within the first or the second gateway (serving or packet gateway in LTE) [13,33,35,38]. This strategy allows scheduling and admission control to be handled by the gateway, while interfacing with the radio node. However, two important challenges emerge in these types of implementations:

1. The radio node MAC scheduling must be controlled by the gateway, such that, if there is wireless channel congestion, it does not interfere with the gateway's own scheduling mechanism. Given the fact that usually the radio node and the gateway are two separate network elements, it is complicated to achieve this without modifying the radio node.

2. Additional information exchange between the radio node and the gateway is required to ensure that the gateway can reliably control the radio node. The required information can be described as: (i) information from the radio node to the gateway in relation to the actual wireless conditions and radio resource utilization, and (ii) information from the gateway to the radio node regarding uplink mobile device scheduling decisions. Note that the specific details, like the frequency of update and level of precision of these exchanged information is to be tailored for a given implementation; in some special cases, already existing features could be used, but it is still highly probable that such features would have to be tailored, e.g., in WiMAX the maximum sustainable rate request from the gateway to the radio node determines the maximum rate that should be allocated to a given service class; if fine-granularity virtualization is used, this feature would have to be applied on a per-flow instead of a per-class basis.

An important aspect to consider when applying virtualization at the gateway level, and which can be difficult to address, is how to handle uplink scheduling, primarily because it implies to remove completely the ability of the radio node to take uplink scheduling decisions.

Finally, it should be mentioned that new challenges arise when multiple radio nodes serve multiple mobile devices, especially in the case when several radio technologies are used. Evidently in this case arbitration between radio technologies for the uplink and the downlink has to be made. Although multi-queue multi-server systems have been studied, and weighted fair queuing-type strategies have been developed [33,39], the addition of other practical constraints, like for example preferred technologies used for the transmission of certain flows, challenge existing proposed solutions [40].

One additional issue that appears when having multiple radio nodes of multiple radio technologies is how to handle connection-oriented flows, in the case when several interfaces are used to transmit portions of the same flow (such as TCP flow). In this scenario a wireless

SDN architecture appears as a promising way to deal with this issue. Since SDN keeps track of the control plane information, it will be easier to gather flow state information, independently of the actual path or radio technology.

2.3.3 Wireless SDN for Video

As mentioned in the introduction the emerging trend of adopting high definition video streams is growing rapidly, requiring higher bandwidth. Additionally, wireless mobile networks are the preferred costumer choice. To deal with the bandwidth growth, and the shift to wireless, we explore the proposed solutions in the SDN context.

Although the amount of available literature on wireless SDN for video is not huge, there are a few important examples of video traffic enhancement applications that have been reported. At the basic level, the proposed algorithms [24, 41] subdivide traffic into real-time and non-real-time, and each type of traffic is then associated with a slice, but this kind of slice isolation could be directly achieved in the LTE or WiMAX frameworks by using the existing traffic classes.

If the goal is to achieve finer traffic shaping for video, it would be necessary to be able to (i) separate video traffic from other real-time traffic, e.g., voice over IP (VoIP), (ii) have a much finer granularity when determining flow properties, such as the relative importance of packets of the flow, or the video properties (duration, size, etc.).

An example of this approach is adopted by the Network Virtualization Substrate (NVS) [16], a proposal for WiMAX virtualization, which essentially consists of a scheduler based on resource or bandwidth provisioning for different slices in the network. The authors of [16] demonstrate that, tagging video packets adequately (e.g., belonging to I, P or B frames when using H.264 video codec), it will be possible to drop some packets within a video slice so as to reduce congestion and improve the average video performance. The video tagging is assumed to be done by the content provider by writing a priority number in the Type of Service field of the IP header. Depending on the network, this may or may not be a good implementation idea, i.e., it depends on whether the network makes use of the Type of Service field for other purposes or not, but another alternative is to do the tagging by using the TCP options field of the TCP header.

As mentioned above, the video packet tagging is completely necessary if one wants to avoid deep packet inspection. Despite the fact that this tagging task might not be

so easy to implement in practice for many applications, the scenario considered in this thesis could assume such cooperation between network operators and content providers. In a value-added service scenario offered by a wireless network operator through SDN and virtualization, it is reasonable to assume that some sort of SLA would be set with specific content providers, e.g., over-the-top content (OTT), so that their video traffic could benefit from the prioritization schemes being offered by the network operator.

Another interesting work found in [17] focuses on UDP video traffic, and provides a deeper study of a potential admission control and scheduling system for video flows in a virtualized WiMAX access network. Their proposed solution (MESA) uses the video tagging discussed above to selectively drop packets; however its most important contribution consists in separating video flows into slices, depending on the length of the video, the creation of a special slice that contains flows during congestion, and flow management techniques to manage the migration of flows between slices. We will describe in more detail this proposal in the next section.

2.3.4 Some Specific Systems

2.3.4.1 NVS

The Network Virtualization Substrate (NVS) [16], investigates a more general resource scheduling policy, specifically defined to virtualize cellular networks. NVS defines at the slice level a set of network resource provisioning algorithms, i.e., inter-slice resource provisioning, and it can allocate resources to slices based on bandwidth or network resources.

The authors of [16] take a dynamic programming approach to solve a constrained optimization scheduling problem for the downlink, and derive a set of inter-slice scheduling policies. An important architectural choice, made by the authors of [16], is not to define a specific intra-slice scheduling system, allowing in this way the managers of each slice to implement any particular scheduling policy that they decide.

In order to demonstrate the high degree of isolation that can be obtained by NVS, in one of their simulation examples, the authors of [16] use a video slice with a simplified version of MESA that coexist with other slices.

2.3.4.2 MESA

As mentioned earlier, MESA [17] is a video delivery framework for WiMAX networks, that is based on the concept of clustering video traffic into slices, assigning membership to the slices so that the scheduling of the different slices can be made based (among other metrics) on long-term user QoE metrics.

The main idea of MESA is to differentiate the long-running and the short-running video flows; the argument in favor of this subdivision is that, when long-running video flows are present in the network, they might consume all the available network resources, and when short-running video flows try to be admitted into the system they are most likely to be rejected, creating a fairness issue between long-running and short-running videos. The long and short slicing enables MESA to actively adapt admission control and scheduling so as to increase fairness across each type of video traffic.

Additionally to the short and long slices, a third slice called the victim slice is created to better handle video flows in case of congestion. In the victim slice the system may decide to drop or not some of the packets to avoid congestion (based on packet priority). This requires that video packets are tagged with an importance indicator. Another important characteristic of MESA is that the scheduling across slices uses a simple proportional fair scheduler, and the ratio of transmission slots assigned to each slice are kept constant over time.

Simple metrics are used to approximate the user QoE at the downlink scheduler (dissatisfaction metric). The QoE metric represents not only network congestion, but also hypothesized user dissatisfaction, based on the video packet loss. Nevertheless, no specific information from the user is needed to calculate the QoE metric. The purpose of the QoE metric is to help in the flow migration process between the short, the long and the victim slices. For example, if a congestion condition is detected, flows with the best value of the QoE metric are migrated to the victim slice; it is also possible to migrate flows from the short slice to the long slice. MESA has been evaluated in [17] using a mobile WiMAX network test bed showing good video quality performance.

2.3.4.3 CellSlice

Another important work was presented in [33]. Motivated by the idea to modify the radio nodes as little as possible, the authors of [33] proposed CellSlice, a generalization of NVS

that moves the solution out of the radio node and into the Access Service Network Gateway (ASN-GW) of WiMAX. One of the most important contributions of CellSlice is the design of a complete uplink and downlink slicing mechanism that does not reside in the radio nodes. It is important to notice here that, because all the uplink and downlink scheduling decisions are taken in the ASN-GW, some exchange of information between the radio node and the ASN-GW is mandatory.

CellSlice assumes not only that the radio node gives periodic feedback to the ASN-GW about its resource utilization and Modulation and Coding Scheme (MCS), but also that the radio node guarantees a maximum sustained rate for each flow, as instructed by the ASN-GW. These interactions would of course be rendered easier in the SDN framework, as it could be assumed that the control plane has knowledge about all of this information. On the downlink, CellSlice bypasses the radio node's scheduler by matching the outgoing traffic to the radio node traffic consumption rate (starving the radio node's buffers to prevent the radio node's scheduling algorithms to kick in); on the uplink side, the ASN-GW control is indirectly enforced through the per-flow mandated maximum sustained rate messages to the radio node.

2.3.4.4 OpenRoads

Another important work was done at Stanford [40] related to the development of the OpenRoads project. OpenRoads [20] aims to use virtualization of wireless networks to allow connections of any device with multiple radio interfaces to multiple networks; OpenRoads is a test bed development, but some of the examples developed are relevant to our work.

The research described in [40] focuses on the scheduling problem at the client side, as a multi-flow, multi-interface scheduling problem: each mobile device in this work has multiple network interfaces (e.g., Wi-Fi, WiMAX and 3G), over which it can transmit information either simultaneously or in sequence. An interesting assumption is the ability to define the preferred interface to be used by some flows. This will mean for example that video flows might prefer the Wi-Fi connection as its transmission rate is generally better.

An important consideration comes from the fact that some synchronization is necessary between network interfaces for each flow, i.e., when a scheduling decision needs to be taken in a given network interface, the average rate, for a given flow, served in each interface needs to be available. The key finding in [40] is that this synchronization can be achieved by acti-

vating a one-bit state information for each flow kept at each network interface. The authors in [40] proposed a generalization of Weighted Fair Queuing and showed that it is practically implementable through a modified deficit round robin technique called multiple-interface deficit round robin. The above considerations for multi-flow, multi-interface scheduling apply directly to the client-side, without control from a radio node.

Note that, in case of a reservation-based uplink system, additional coordination between mobile devices and radio nodes is needed, and in the case of mixed architectures, where some interfaces and RATs would be reservation-based and some others would be contention-based, even more complex situations arise.

It is important to mention that one of the major obstacles to using the same architecture for downlink multi-interface scheduling is the availability of global information (network interface status and in which interfaces flows will be scheduled). Nevertheless, if we consider the SDN architecture, the control plane will concentrate all the information, allowing us to abstract from this problem, at least theoretically. In [40], an HTTP proxy-based solution was proposed to partially solve this problem, where the author proposes to use virtual Ethernet interfaces connected to a special gateway that handles multiple interfaces, this approach helps to decouple the packets in a given flow from the IP addresses on each interface, allowing in this way the set of interfaces to change dynamically as connectivity changes.

2.4 Traffic Load Balancing

This section presents a review on the traffic load balancing strategies using multiple network interfaces for simultaneous data transmission. In general, if there exist different wireless networks, multiple paths can be established by using the same or a totally different wireless technology. We focus on load balancing over multiple paths, thus we do not directly address routing to establish multiple paths, i.e., paths are assumed to be established by routing techniques.

Having multiple network interfaces, that enable us to have multiple parallel connections simultaneously, makes it necessary to efficiently use all the available network resources. In that sense load balancing helps us not only to avoid single points of failure, but also to guarantee QoS at high data rates and to facilitate network provisioning.

Fig. 2.3 illustrates the basic load balancing principle. The load balancing component

splits the traffic into smaller traffic units, each of which independently takes a path. Because of the difference in which each proposed solution splits traffic, each of them exhibits different advantages and shortcomings.

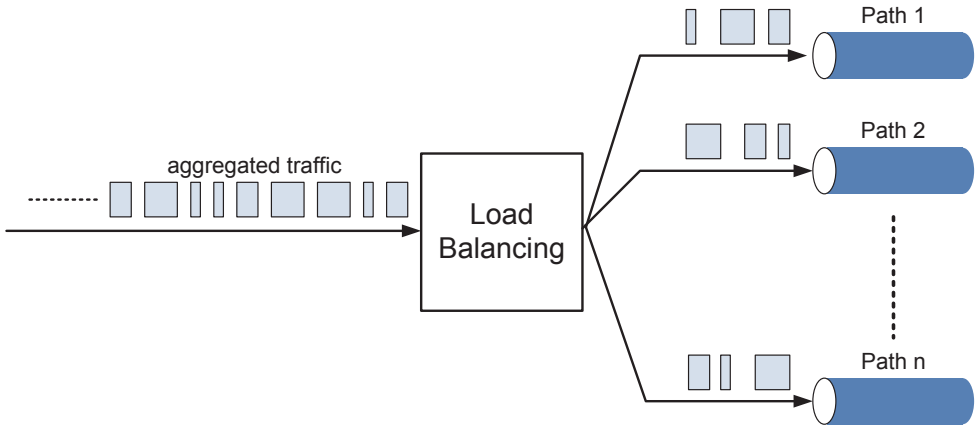


Fig. 2.3 Basic load balancing principle

2.4.1 Load Balancing Challenges

Load balancing performance might affect Quality of Service perceived by network users. Some of the challenges can be summarized as follows.

2.4.1.1 Packet Reordering

Packet reordering occurs when the order of packets at the destination is different from the order of the same packets at the source. Fig. 2.4 shows an example of a situation with packet reordering; the reordered packets are highlighted. Packet reordering has a negative impact on both TCP and UDP-based applications. For TCP-based applications, duplicate acknowledgements are generated when reordered packets are received, causing to retransmit packets and to decrease the transmission rate. If packet reordering is persistent, TCP performance quickly degrades. For UDP-based applications, reordered packets that arrive after the play out deadline are considered lost.

Packet reordering is not a rare anomaly. According to the study conducted in [42] packet reordering is on par with, if not a larger issue than, packet loss. For UDP traffic over the internet, packet reordering occurs less than 3% of the time [43]. In addition,

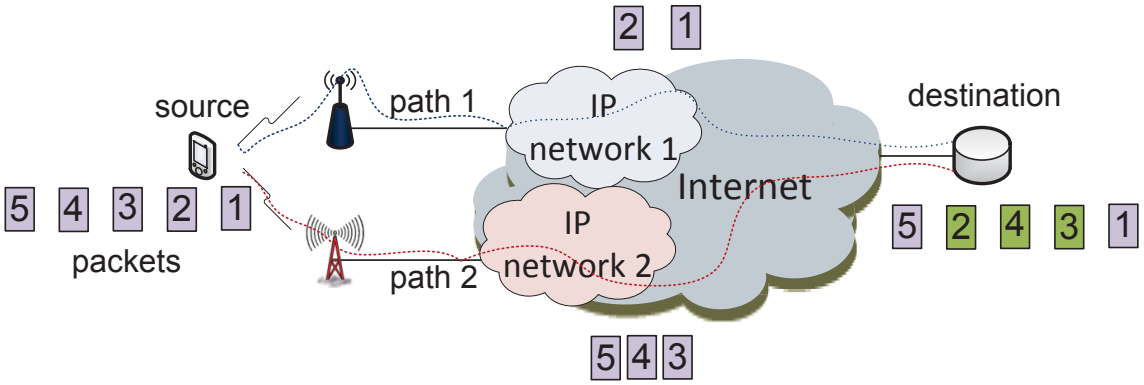


Fig. 2.4 Packet reordering example, packets 2, 3 and 4 arrive to the destination out of order

packets of smaller sizes are reordered more frequently than larger packets. This has negative implications on UDP-based applications such as VoIP which has small packet sizes. If packet reordering is persistent during a VoIP session, the quality of the voice significantly degrades.

It is worth noticing that the impact of packet reordering is directly related to the application in use. Some applications might be more tolerant to packet reordering than others, e.g., UDP video streaming when no packets are late for play out, packet reordering is still acceptable.

Most of the current research discussing packet reordering is focused on single-path flows through wired, high speed networks. Nevertheless, one of the few papers that considers an heterogeneous multipath scenario shows that packet reordering can approach 50% [6]. It is interesting to note that packet reordering is not consistent across the paths that exhibited it. Instead, there is a close relationship between packet reordering and traffic load [42], which also exhibits a similar behavior [44]. As a consequence, the advent of high bandwidth applications, such as high quality video, will increase packet reordering.

We should also mention that in wireless networks, the amount of packet reordering is heavily dependent on many parameters that are beyond the user's control [6] such as unpredictable interference, varying signal quality, other users contending on the same channel, overloaded radio nodes, etc. The tolerance to packet reordering depends on various factors [45], including packet size, transmission rate, capacity of receive buffers, application purpose, etc. The most vulnerable applications are those that generate small packets followed by large packets in a single stream. This is because even though the average transfer

rate is low, packet reordering may occur within packet bursts, where the packets are closer to each other.

Even though the existence of packet reordering in IP networks is established, its extent is still not completely understood. An early effort to quantify the effects of packet reordering on IPTV using a MPEG-2 video codec revealed that the video quality becomes unacceptable for more than 0.12% of reordered packets [46]. It is also noted that the viewer's sensitivity to packet reordering depends on the video genre, with sport being the most vulnerable.

The packet reordering problem may be mitigated by using buffers. However, for battery-operated devices, such as cell phones and tablets, large buffers are very resource-intensive. Keeping in mind the goal of not to increase the workload of such devices, this thesis studies the problem of how to split traffic over multiple links in heterogeneous wireless networks.

2.4.1.2 Energy Consumption

It is well known that one of the challenges of mobile devices is their limited battery power. A mobile device consumes a significant amount of power not only during operation, but also during idle periods (until its battery gets depleted). Additionally, if we consider the case when the mobile device is equipped with multiple network interfaces, its battery power consumption can increase even more.

Consuming more energy leads to reduce the mobile device operational lifetime, risking to prematurely terminate transmission. Therefore, an efficient load balancing technique should include the appropriate mechanisms to minimize the power consumption, such that the battery life is prolonged and the transmissions do not get interrupted.

2.4.1.3 Other Challenges in Load Balancing

Other challenges in load balancing include extra communication overhead and increasing implementation complexity. Ideally the extra communication overhead needed to achieve load balancing (network probing, exchange of network information messages, etc.) should be minimized. The communication overhead decreases the available bandwidth and increases the network load, e.g., some solutions require that the path state information must be updated often enough to minimize errors in the path selection, creating in this way a trade-off between minimizing the communication overhead and improving the load balancing accuracy.

Implementation complexity must also be considered when implementing a given solution. To be of practical use in real networks, installations of new components or modifications of existing ones should be minimized. For example, path delay, traffic rate measurements as well as packet counter information are typical components for performance improvement, but they may cause extra computational complexities and overheads.

2.4.2 Existing Solutions

Traffic load balancing over multipath networks has been an active research area in recent years [5, 47–49]. The existing models can be loosely categorized into packet-based and flow-based load balancing models.

2.4.2.1 Flow-based Load Balancing

The flow-based load balancing models address the problem by assigning packets of the same flow to the same path. Although the risk of packet reordering decreases, queuing packets over the same path usually causes the end-to-end delay to increase.

Flowlet Aware Routing Engine (FLARE) [50] allows a flow of packets to be split into a subset of packets of an original flow referred to as a flowlet; FLARE uses these flowlets as the balancing unit. All packets in a flowlet are destined for the same path, but packets heading for the same destination may be carried in different flowlets. Various flow characteristics can be taken into account in a splitting condition, e.g., packet inter-arrival time and packet arrival rate, depending on the load balancing objective.

The most important parameter of FLARE is the inter-arrival time threshold. The flowlet can be interpreted as a group of packets having their inter-arrival time smaller than the threshold. A packet arrived within the threshold is part of an existing flowlet and will be sent via the same path as the previous packet. Otherwise, the packet arrived beyond the threshold corresponds to the head of a new flowlet, and will be assigned to the path with the lowest load.

The performance of FLARE depends on the delay estimation accuracy. In a bursty traffic environment, a sudden increase in the packet arrival rate can cause underestimation errors. These estimation errors can be attenuated by more frequent measurements, but they incur in communication overhead, thus consuming additional bandwidth resources. Additionally FLARE does not consider the packet loss during the scheduling. As TCP

guarantees delivery, it retransmits the packet when packet loss occurs. The retransmission cost is high in FLARE because it may keep sending flowlets in congested paths. Nevertheless, FLARE provides a general approach that can be easily extended to wireless uplink systems and UDP traffic.

Another approach was presented in [51], where the authors investigate the challenge of splitting a traffic flow over WiMAX and Wi-Fi links and proposed an airtime-balance method. This method maps the traffic load to an airtime cost function and uses it to split traffic. The idea is to send packets to radio nodes so that the airtime cost is balanced. However, this solution does not deal with packet reordering.

Adaptive Load Balancing Algorithm (ALBAM) [52] focuses on the TCP drawbacks found in FLARE and proposes an algorithm that schedules traffic only when the packet inter-arrival time enables it to compensate for the path delay difference.

ALBAM can be divided into three parts. First, ALBAM performs delay measurement for each path. The path delay is used to split traffic across multiple paths. When ALBAM decides to switch traffic from one path to another, it calculates the delay difference between two paths and buffers the traffic until the interval is larger than the difference. This operation minimizes packet reordering. The second part is the traffic division. ALBAM switches the flow to different paths at the granularity of flowlets. ALBAM divides the flow into several virtual flowlets (VF). The third part is the Packet Number Estimation Algorithm (PNEA). ALBAM employs PNEA to monitor the buffer usage of each path and prevent buffer overflow during transmission. If buffer overflow is predicted ALBAM switches the traffic to another path. Although ALBAM has a more accurate way of evaluating the delay the main disadvantage is that it strongly relies on accurate delay estimations. Additionally, in the case of packet reordering, it is not trying to eliminate the main problem, but it is reacting to fix it.

Flow Slice (FS) [53] proposes to subdivide every flow into smaller pieces (flow slices) at every inter-flow slice interval larger than a predefined threshold and to balance the traffic load on a flow slice granularity. FS reduces the probability of packet reordering at the cost of high end-to-end delay.

2.4.2.2 Packet-based Load Balancing

Packet-based load balancing models manage to reduce the overall end-to-end delay, by using packets as the basic allocation unit. This strategy reduces their ability to achieve low levels of packet reordering.

Effective Delay Controlled Load Distribution (E-DCLD) [54] is driven by the realization that inefficient load balancing can degrade the network performance and tackles the problem by formulating an optimization problem that balances the end-to-end delay among all the available paths. One of the main concerns with E-DCLD is its low convergence time. E-DCLD takes into account the input traffic rate and the instantaneous queue size, which are locally available information, in determining how to split the traffic, and thereby properly responding to the network condition without additional network overhead. The idea is to minimize a delay cost function and then apply a Surplus Round Robin scheduling algorithm. One of the disadvantages of E-DCLD is that it needs a considerable amount of time to converge to the optimal solution.

Convex optimization-Based Method (CBM) [55] handles the low convergence time experienced by E-DCLD by formulating the load balancing problem as a convex optimization problem. Although CBM solves the low convergence time issue, its main disadvantage is that the input traffic is modeled as a Poisson distribution, but this assumption is not suitable for most video traffic applications.

An interesting approach is studied in [56], where the authors propose a machine learning algorithm that calculates the load balancing splitting ratio from the available QoS information. In order to obtain an optimal load balancing ratio from various QoS metrics such as throughput, delay, packet loss, etc. the authors of [56] apply a machine learning algorithm, which estimates optimal traffic allocation ratio from the current QoS information. The optimal traffic allocation ratio may depend on devices, applications, the transmission methods, and so on. By using a machine learning algorithm, they can control traffic allocation adaptively.

One important drawback that we should note here is that the learning algorithm learns the best allocation ratio for the set of various available QoS parameters. This means that if for some reason the conditions change, as if for example, severe congestion occurs on one of the paths, the algorithm may not find the best load balancing ratio.

Sub-Packet based Multipath Load Distribution (SPMLD) [57] proposes to minimize

the total packet delay by aggregating multiple parallel paths as a single virtual path. The main challenge for SPMLD is not only that paths might have different characteristics, e.g., variable bandwidth and propagation delay, but also the inherent characteristics of real-time multimedia traffic, e.g., variable flow rate and packet size.

We can also mention that SPMLD has formulated the packet splitting over multipath as a constrained optimization problem and has derived its solution based on the successive progressive approximation method. To derive the solution, a D/M/1 queuing model was introduced and two distributed algorithms that have to be implemented in the source and the destination were proposed.

Besides its complexity, since SPMLD split packets into sub-packets at the application layer, developing its solution for packet splitting and reassemble in the sender and receiver sides, SPMLD also introduces an extra overhead caused by splitting packets (reported to be smaller than 6.5%).

Power-efficient load distribution (PELD) [58] is one of the few research works that acknowledge that running multiple network interfaces simultaneously can significantly reduce the battery life of a mobile device.

To overcome the power-efficiency problem, PELD takes advantage of the sleep-mode mechanism available in Wi-Fi and WiMAX. PELD's proposed solution is based on solving a non-convex optimization problem, using a greedy algorithm, which minimizes the power consumption while taking into account the delay and the packet reordering risk. Although PELD suggests the possibility of reducing power consumption of multi-interfaced devices, load balancing accuracy was not considered.

Other proposed solutions, that focus on video delivery in multipath heterogeneous wireless networks [59,60], use application layer solutions that require modifications in the origin and in the destination.

To summarize, flow-based models manage to limit packet reordering to a negligible level, but at the cost of large end-to-end delay. On the other hand, packet-based models manage to reduce end-to-end delay, but their ability to reduce packet reordering has to be addressed especially when the traffic load is high. It is worth noticing that to the best of our knowledge, none of the literature deals directly with wireless uplink systems. Additionally, most of the papers focus their work only on the TCP scenario, not considering that UDP is the preferred choice when dealing with real-time applications.

2.5 Device-to-device Communication

This section presents a review on the device-to-device (D2D) communication strategies that can be applied in wireless cellular networks. We focus our attention on how to increase bandwidth capacity as it is an important requirement to address, specially when dealing with video transmission.

D2D communications have been extensively studied in non-cellular technologies such as Wi-Fi, near field communication (NFC), etc., but it is only until recently when trying to define the next generation of mobile networks that new proposals have tried to integrate D2D communications into cellular networks [7]. A survey of D2D communications in cellular wireless networks is presented in [61, 62]. There the authors explore the main advantages and challenges of D2D communications.

In general, D2D communication networks can be classified into stand-alone D2D communications and network-assisted D2D communications, see Fig. 2.5. The main difference between the two types is that stand-alone D2D does not have any infrastructure to organize the communications, while network-assisted D2D is supported by the radio node to assign resources and establish communications.

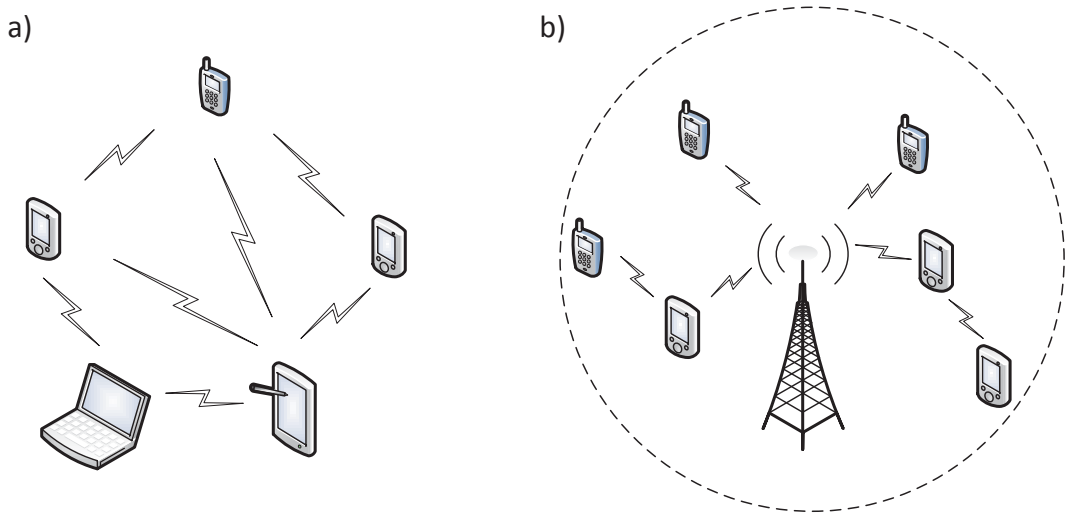


Fig. 2.5 D2D communications: a) Stand-alone D2D, b) Network-assisted D2D

2.5.1 Network-assisted D2D: Relaying

Unlike actual cellular networks, where mobile devices establish a direct communication link with their radio node, D2D communications offer the possibility to establish relay-aided communications. Relay-aided solutions could improve the network performance, especially when mobile devices are far from the radio node (cell edge).

In [63] an opportunistic coverage expansion is considered, where mobile devices should relay traffic to neighbor devices in order to get access to an alternative Wi-Fi access point. The authors of [64] also study offloading of cellular networks onto D2D links in unlicensed bands. It has been shown that offloading traffic can significantly reduce power consumption. Multi-hop relay communications in the context of decentralized infrastructure-less networks have been studied in [65]. It has been reported there that relay by smart-phones can successfully create a network powered by communication devices independently of network operators.

The problem of how to subdivide the spectrum to allow D2D communications was discussed in [66]. In [67], an analytical approach is used to evaluate the system capacity of overlay in-band, underlay in-band and out-of-band D2D scenarios. Another work that focuses on relay-aided communications is described in [68]. Unlike traditional solutions, the authors in [68] formulate a radio allocation problem that minimizes interference and propose a low complexity distributed solution. In [69], the authors focus their analysis on the data plane functionality, and present coverage improvement and system capacity results for a system that enables cellular controlled D2D relaying. In [70], under the assumption that not all the mobile devices might be interested in relaying, a mechanism to incentivize D2D relaying using tokens was proposed. The authors in [70] proposed a supervised learning algorithm that each mobile device can use to learn the optimal cooperation policy, which could be used to decide if the mobile device should relay or not in exchange for tokens.

In [71], the creation of D2D based clusters of users was proposed. Although the authors in [71] prove to increase the average spectral efficiency and network's capacity, in reality their solution only achieves so in the downlink, while in the uplink, which is typically the weakest point on wireless networks, their solution shows a significant capacity reduction.

Although there is a lot of literature on relaying-aided communications in wireless cellular networks, this thesis consider a new scenario where mobile devices are equipped with multiple network interfaces of the same technology that can act as full duplex relays (in the

context of this thesis full duplex refers to the ability to use one interface to transmit and one interface to receive data at the same time), and it is the cellular network who handles all the mechanisms needed to operate the network, i.e., channel assignments, interference avoidance, etc.

2.5.2 D2D Communication Challenges

Enabling network-assisted D2D relay communications on cellular networks might allow us to benefit from the potential advantages of this technology. However, some technical challenges and design problems need to be addressed, such as how to select relays, how to mitigate interference caused by new D2D communications, how to allocate wireless resources, how to reduce the overall power consumption and how to enhance security.

2.5.2.1 Relay Selection

Relay selection is an important challenge in network-assisted D2D networks; since in general the number of available relays is large, it should be a design goal to make this selection process as efficient as possible. The network plays an important role in relay selection and how we choose the relays affects not only the possible bandwidth gains, but also the interference levels and the power consumption [72, 73].

2.5.2.2 Interference Mitigation

In a D2D network, in which the uplink and the downlink wireless resources have to be reused, it is important not only to design a mechanism that prevents D2D access from interrupting other active communications, but also to maximize the system throughput for a given QoS. Mitigating interference can usually be achieved by carefully choosing the power levels and the wireless resource allocations [74, 75].

2.5.2.3 Resource Allocation

Another important aspect to consider in D2D communications is to efficiently manage the wireless resources such that the D2D users communicate over resource blocks that are not currently in use by other nearby mobile devices. Although many resource allocation techniques have been proposed in the literature [76, 77], in general the resource allocation

problem in network-assisted D2D networks is very challenging, mainly because of the large number of available mobile devices in the cell which result in an increased complexity.

2.5.2.4 Power Consumption

Power consumption is one of the main requirements of D2D communications because of the limited battery energy available in mobile devices. In this sense cooperative techniques could be applied to extend the lifetime of D2D networks. Another aspect to consider is that power consumption at the cell-edge is higher than at the center of the cell mainly due to the large distance between the transmitter and the receiver. Thus properly choosing the power levels not only impacts the interference levels, but also affects the throughput [78, 79].

2.5.2.5 Security

As mobile devices have lower computational capabilities than radio nodes, security protocols for D2D communication must be not only robust but also simple. Moreover, location, mobility and power resources of the mobile devices acting as relays must be taken into consideration to ensure service availability. Reliable mobile device authentication is very important in D2D because it helps to prevent attacks in which an attacker hidden in the network structure tries to access services using the identity of another mobile device [80,81].

2.6 Summary

In the above literature review, we can notice that although some research has been conducted on video transmission over wireless cellular networks, not much has been done on a multi-radio multi-interface system with focus on video traffic. The existing work that is most relevant might be the one from the OpenRoads project, which is the first to openly include multiple radios. In this context, we aim to take advantage of the SDN architecture in the wireless access to offer value-added wireless video transmission services to video content providers; assuming that any given video consumer can be served by a number of different access nodes simultaneously, potentially using different radio access technologies.

We explore the admission control, scheduling and virtualization-based slicing of traffic as the main avenues to explore the potential benefits of the SDN architecture, based on the idea that a centralized control plane might help to efficiently manage different access tech-

nologies. Together with this we also explore load balancing and D2D relaying techniques in the uplink as supporting technologies that can benefit the network by helping to handle the ever growing traffic demands.

Chapter 3

Video Traffic Management in Software Defined Networks

3.1 Introduction

The ever increasing video traffic demand on wireless cellular networks and the limited capacity to increase the radio spectrum allocation could reduce our ability to avoid network overload. For non-elastic traffic such as video or voice, admission control and scheduling are some of the flow management techniques that can be used to allow optimal network resource utilization [82].

This chapter focuses on developing a novel framework in wireless mobile cellular networks called JSM. A first step is to design an SDN management framework that ameliorates the quality of service for each user by isolating resources for video services. The main features are: first, it handles mobile devices connected to multiple radio nodes, enabling multi-node traffic management; second, it defines an admission control and scheduling policy across different types of traffic.

Considering that video streaming is dominating the mobile traffic, admission control presents many challenges. Firstly, pre-provisioning resources for each video flow can lead to either underutilize, or to overutilize wireless resources. Secondly, wireless channel capacity can fluctuate over time due to interference, user mobility, etc. Finally, because video lengths and rates can vary widely a system only considering average capacity can be biased against users with poor channel quality.

3.2 Joint Slice Manager (JSM) Framework

The JSM framework[1] builds on the concept of having a centralized controller that jointly handles flow admission control and scheduling in the downlink. JSM takes advantage of the SDN architecture which splits the data plane and the control plane, mainly to virtualize the wireless access network and to allow, consequently, for differentiated services for video traffic. As a result, one of the most important functionalities of JSM is the ability to handle mobile devices connected simultaneously to multiple radio nodes. JSM implements a solution that bundles groups of flows of similar characteristics (slices) within the data plane, and manages wireless resource allocation procedures across slices and across multiple radio nodes within the control plane. JSM has the following characteristics:

- Unlike existing SDN-based video frameworks
  - One instance of JSM can virtualize and handle multiple radio nodes, multiple radio access technologies and multiple frequency bands.
  - It dynamically adapts the allocated resources of video slices according to an optimization framework that aims to maximize utility functions of slices while trying to ensure a minimal QoS for each slice.
  - It balances the traffic between adjacent radio nodes in order to avoid congestion and maximizes the radio nodes utilization.
  - It increases the quality of experience of each user by minimizing the number of dropped packets.
- It enables the ability of each video slice to handle flow scheduling independently according to its needs.

Fig. 3.1 shows the JSM architecture. JSM is designed to be external to the radio nodes. It should be noticed that only one instance of JSM handles multiple radio nodes. While designing JSM, we make the following assumptions:

1. JSM should be capable of handling mobile devices connected to multiple radio nodes.

[1] A United States Patent application has been submitted, "System and Method for Transmission Management in Software Defined Networks", PCT international application number 14/516,394 filed on 10/16/2014

2. The minimum offered rate, and the maximum tolerable delay are set beforehand between users and their network operators.

3. Every radio node is able to provide periodic feedback to JSM of the link capacity of each mobile device on each network interface, i.e., in the specific case of LTE we need the information of three quantities:

   (a) the number of slots used for transmission and the total available slots in units of time (utilization);

   (b) the current modulation and coding scheme (MCS) of each active mobile device;

   (c) the Signal-to-Interference plus Noise Ratio (SINR) of each active mobile device.

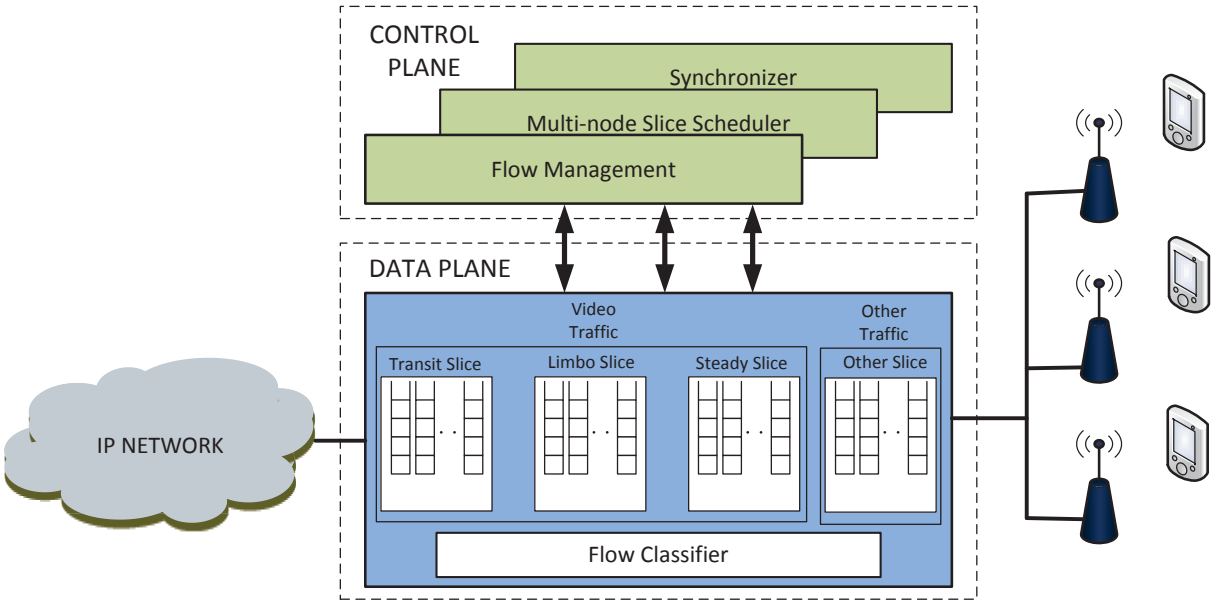


Fig. 3.1 JSM architecture

3.2.1 Overview

JSM is based on the virtualization paradigm that allows to treat groups of resources as slices. In our case a slice can be defined as a group of flows of similar characteristics. Slicing refers to the process of assigning flows (resources) to a network. Using slices have three important advantages: First, it allows the isolation of resources across slices, i.e., any

change in one slice does not lead to reduction in network resources for other slices. Second, it allows each slice to handle flow scheduling differently. Finally, it can dynamically adapt slice sizes such as to maximize network utilization.

JSM is instantiated as shown in Fig. 3.1. In an LTE network JSM can be deployed on the P-GW. The following steps give an brief overview of JSM:

1. JSM creates at least three slices for the case of video traffic: Two slices for flows of short duration (short flows), and one for flows of long duration (long flows). Initially, all flows entering to the system are classified as short flows.

2. JSM reserve minimum transmission resources for each slice according to a specific proportion.

3. A flow classifier sends packets to the appropriate slice. Short flows arriving to the system for the first time go to the transit slice, long flows go to the steady slice; if there is no enough capacity in transit or steady slices, flows go to the limbo slice.

4. The multi-node slice scheduler periodically forwards packets from the slices to the appropriate radio node for downlink transmission. To avoid interference from its own's radio node scheduler, the multi-node slice scheduler only forwards enough number of packets to the radio node so as to implicitly disable the radio node own's scheduler.

5. Every certain amount of time JSM evaluates the slice queues and decides whether to migrate flows to a more appropriate slice or to discard packets. Flows can be migrated based on available capacity, duration on the system, etc. Packets can be discarded due to capacity constraints or packet time-outs.

6. Finally, since each JSM instance handles multiple radio nodes, managing handoff consists of just updating the correct mobile device to radio node association, without affecting the slice management, i.e., it does not require for a flow to start over and enter to the transit slice as a new flow.

JSM can be considered an extension of the Multi-level Feedback Queue scheduler, studied in operations research [83], in the sense that additionally to the flow migration strategy, JSM is also able to serve multiple queues at the same time, while prioritizing queues according to the traffic conditions. Another feature of JSM is that it has a dynamic adaptation of the slice size, which will be discussed in Section 3.4.

3.2.2 Slices

JSM creates at least two main slices, one for video traffic and one for other traffic (as many slices as desired can be instantiated). The *other traffic* slice is assigned a fraction of the available resource slots. Inside the video traffic slice we create three slices:

1. Transit slice: it groups two types of flows; the flows that enter into the system for the first time (new flows), and the flows of short duration.

2. Steady slice: it groups the long running flows (flows of long duration). It receives flows from the transit slice.

3. Limbo slice: it is for temporal use; it receives flows from the transit slice and the steady slice. Flows in the limbo slice will be scheduled only if there are enough resources available.

Every slice handles its own scheduling policy. A given slice schedules packets only if there are enough resources, otherwise it tries to migrate flows to a more suitable slice, according to the flow management policies described in the next section.

3.2.3 Flow Management

The flow classifier directs each flow to the appropriate slice. Initially, each flow that enters the video slice is classified as a transit flow; if the transit slice is overloaded, the flow attempts to be served in the limbo slice. Flows already in the steady slice continue to be classified as steady flows. If the steady slice is overloaded, flows are temporarily assigned to the limbo slice. Notice that if there is not enough capacity in the limbo slice packets are discarded. The flow management is in charge of the migration strategy. JSM defines four migration procedures:

1. Migration from transit to limbo.

2. Migration from transit to steady.

3. Migration from limbo to transit.

4. Migration from steady to limbo.

3.2.3.1 Migration from Transit to Limbo:

This procedure is invoked every τ_1 units of time. The transit slice evaluates if there are enough slots to drain all its queues; if there are not enough slots, some flows with a maximum penalty metric value P_w (refer to Section 3.2.4) are migrated to the limbo slice such that the remaining queues meet the requirement.

Let us assume that $\mathcal{N}$ is the set of radio nodes, $\mathcal{W}_n$ is the set of flows served by radio node n. Given flow $w \in \mathcal{W}_n$, we define C_n to account for the capacity of flows F_w in the transit slice associated with the mobile device connected to radio node n plus the proportion of the capacity of flows F_w connected to multiple radio nodes:

$$C_n = \sum_{w \in \mathcal{W}_n} F_w + \sum_{w \in \mathcal{W}_{n'}, n' \neq n} F_w / n_w, \tag{3.1}$$

where n_w is the number of radio nodes flow w is connected to. Algorithm 1 describes the procedure to migrate from transit to limbo. Notice that we evaluate the capacity of each radio node based on the moving average of slots allocated to the transit slice.

Algorithm 1 Migration from transit to limbo

```
Every τ1 units of time
Compute capacity C_n and threshold T_n for each radio node n
for every radio node n do
    while C_n > T_n do
        migrate flow w with maximum P_w
        compute B_w as the number of bytes in queue by flow w
        update C_n ← C_n − B_w
    end while
end for
```

The threshold T_n is calculated as the moving average of the bytes served in the transit slice R_t multiplied by the maximum delay:

$$T_n = delay^{max} \times R_t. \tag{3.2}$$

Another way to migrate packets to the limbo slice is when packets reach their maximum delay. In this case the packets are migrated to the limbo slice with the hope of being served if there are enough resources.

3.2.3.2 Migration from Transit to Steady:

If flows are active for τ_3 units of time, they are migrated to the steady slice. If there is not enough capacity, flows are rejected from the system.

3.2.3.3 Migration from Limbo to Transit:

This procedure is invoked every τ_4 units of time. Flows that came from the transit slice attempt to regain access to the transit slice. If the transit slice is overloaded the flows remain in the limbo slice. Packets are dropped by time-out only if their delay is bigger than a preset threshold τ_2.

3.2.3.4 Migration from Steady to Limbo:

If a packet of a flow already being served by the steady slice does not gain access to the steady slice (because the slice is temporally overloaded), that packet attempts to enter the limbo slice. If there is not enough capacity in the limbo slice the packet is discarded. Notice that this action is for a packet, from a flow classified in the steady slice, that arrives to the system, not the packets that are already in the steady slice.

3.2.4 Penalty Metric

The penalty metric P_w is used to drive flow migration decisions. We define the penalty metric (normalized) in terms of two values : (1) The actual bit rate R_w, a flow w is at; (2) the average delay of each flow w $delay_w$, P_w is defined as:

$$P_w = \frac{delay_w}{R_w}, \tag{3.3}$$

The numerator of equation (3.3) indicates that flows with higher delays are less likely to be served unless the channel quality gets better; more directly, this parameter ensures that during overload, users with better channel quality are preferred. The denominator ensures that flows with higher bit rates receive better treatment.

3.2.5 Admission Control and Scheduling

Admission control is done when the packet is initially received in the transit slice, we admit users only if there are enough resources in the system. The scheduling module dequeue packets from the slices and forwards them to their respective radio node for downlink transmission. Packets are dequeued from the slices such that the number of transmission slots utilized by each slice is in the ratio of $(S_t + S_l) : S_s : S_o$, corresponding to the number of slots used by the transit (S_t) and the limbo (S_l) slices, the steady slice (S_s) and the other slice (S_o) respectively.

The transit slice, the steady slice and the other slice receive priority over the limbo slice during scheduling. The limbo slice gets scheduled only if flows in the transit slice or in steady slice do not have packets. The number of slots used by each packet is calculated as the ratio of the packet size in bytes and the current transmission bit rate (it depends on the MCS) of the link to the corresponding flow.

3.2.6 Multi-node Slice Scheduler

JSM defines different resource management mechanisms for each slice. We use weighted fair scheduling based on the sizes of the flows for the transit and steady slices, while the limbo slice uses shortest job first scheduler with the additional feature of giving higher priority to important packets. Here important packets refer to packets marked as important by the source, e.g., I frames in H.264 video coding. Using independent scheduling strategies aims at maximizing the probability of delivery of packets by defining the most suitable scheduling for each slice.

The output of each slice resource management unit is then fed to an inter-slice multi-node scheduler whose work is not only to decide on the scheduling order of each slice, but also on which interface (i.e., through which radio node) data should be transmitted.

The multi-node slice scheduler ensures that the downlink radio node scheduler does not interfere with the JSM's own scheduling mechanism. A reasonable solution is obtained by controlling implicitly the work of the radio node scheduling, by starving its buffers so its own scheduling algorithms do not have to kick in. To this effect JSM uses the utilization information periodically provided by each radio node (at least every δ units of time).

3.2.7 Multi-node Traffic Management

One of the important features of JSM is the ability to handle mobile devices connected to multiple radio nodes, e.g., two radio nodes serving two overlapping sets of mobile devices (see Fig. 3.2). In this scenario the problem is to allocate resources across radio nodes so as to maximize the overall user QoS. The possibility of jointly optimizing the scheduling in several radio nodes serving the same mobile device gives a strong advantage. The solution proposed by JSM requires a single slicing entity that is capable of handling two or more radio nodes, and efficiently admit and assign resources across several interfaces.

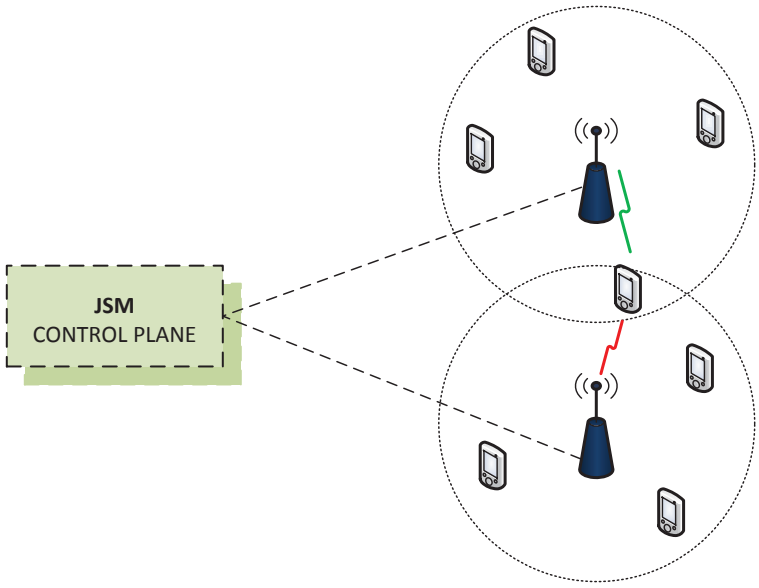


Fig. 3.2 Mobile device attached to two radio nodes

Algorithm 2 describes the operation of the basic traffic balancing module. Let us assume that Cap_n is the total achievable rate of radio node n in terms of bits per second (bps), and R_n is the total average served rate of radio node n. In order to effectively balance traffic we define the decision metric D_n as follows:

$$D_n = \frac{Cap_n}{R_n} \qquad n \in \mathcal{N}, \tag{3.4}$$

where Cap_n is evaluated as:

Algorithm 2 Traffic balancing (mobile device attached to 2 radio nodes)

```
update Cap_n every δ units of time
for Every packet to be scheduled do
    update decision metric D_n
    switch (packet)
    case packet directed to the mobile device connected to radio node x:
        Schedule packet on radio node x
    case packet directed to the mobile device connected to radio node y:
        Schedule packet on radio node y
    case packet directed to the mobile device connected to radio node x and y:
        if D_x < D_y then
            Schedule packet on radio node y
        else
            Schedule packet on radio node x
        end if
    end switch
end for
```

$$Cap_n = \sum_{w \in \mathcal{W}_n} Cap_w \qquad n \in \mathcal{N}, \tag{3.5}$$

and Cap_w is the total capacity estimated by each flow w served by radio node n. Cap_w depends on the MCS which depends on the radio channel conditions experienced by flow w. It is worth mentioning that mobile devices connected to multiple radio nodes are the ones that contribute to the traffic balancing procedure. The algorithm relies on the trade-off between the total achievable rate and the total served rate. When Algorithm 2 converges, the values of the decision metric D_n are the same. This is because if radio node x has a smaller value of D_x the system will route more traffic to radio node y, which increases the average rate and thus decreasing the value of the decision metric D_y.

3.2.8 Implementation

The implementation of JSM for the case of LTE systems was carried out under the NS-3 framework. Fig. 3.3 shows where the JSM implementation is situated in the overall NS-3 simulation setup.

In order to understand the way in which JSM is invoked within the NS-3 LTE implementation, consider a downlink data flow between Internet and a UE (See Fig. 3.3). First of all, a remote host connected to internet is responsible for generating IP packets addressed to the UE. After being routed to the SGW/PGW node through the internet routing mechanisms the IP packet reaches the generic NetDevice function of the SGW/PGW node. Then, the NetDevice forwards the received packet to the VirtualNetDevice function acting as a gateway to the local IP subnet, leading to the desired UE. The VirtualNetDevice sends the received packet to JSM which is responsible for classifying and queuing the packet within its corresponding slice as illustrated in Fig. 3.3. Then, JSM sends the packet to the EpcSgwPgw Application which follows the usual LTE data flow process until the packet reaches the UE.

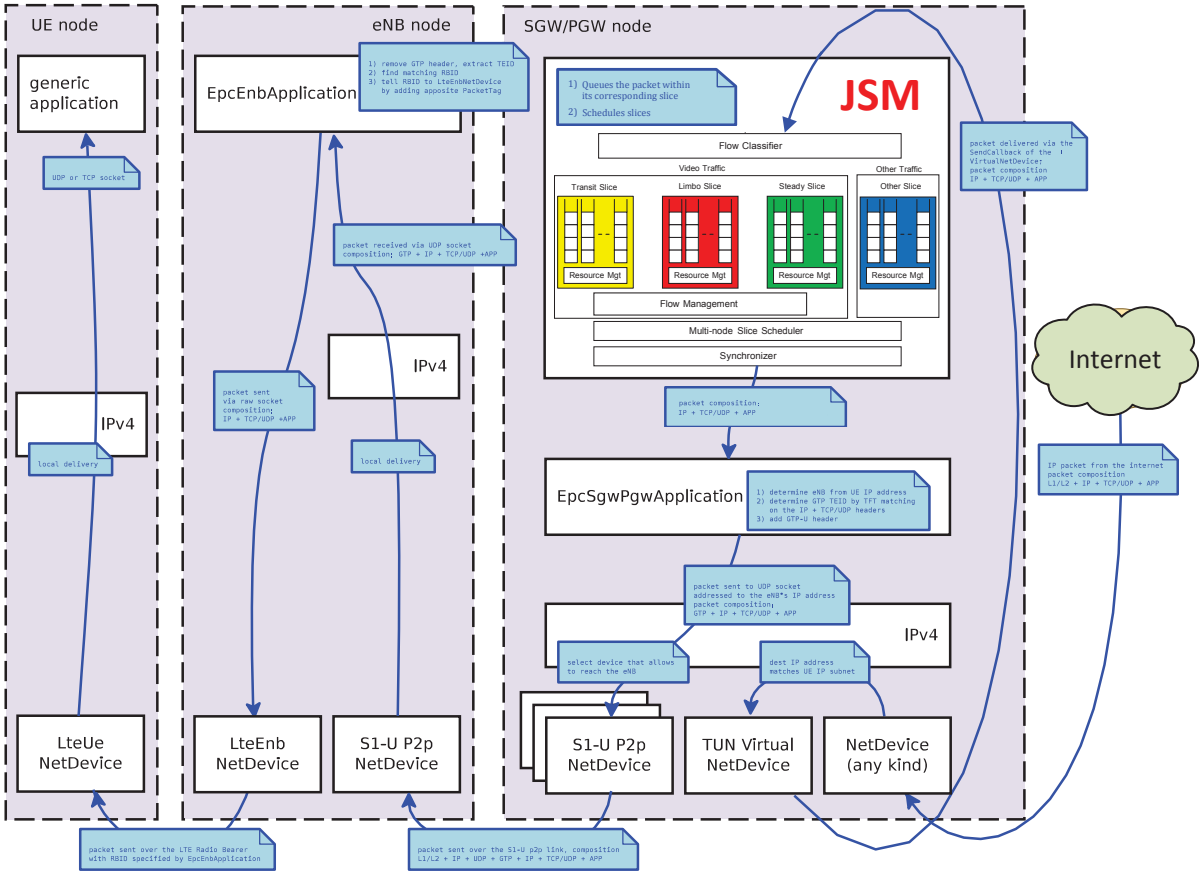


Fig. 3.3 Positioning of the Joint Slice Manager in the LTE NS-3 implementation.

3.3 Performance Evaluation

3.3.1 Metrics

In order to evaluate JSM performance we use the following QoS metrics:

- **Delay:** Records the end-to-end delay, i.e., the time the frame is received at the destination, minus the time the frame is transmitted by the source.
- **Drop:** Records the number of dropping events as the number of frames dropped, i.e., when a whole frame does not arrive at the destination (none of its packets arrive).
- **Corruption:** Records the number of corruption events as the number of frames corrupted, i.e., when not all the packets of a given frame arrive at the destination (only a few packets arrive).
- **Re-buffering:** This metric evaluates the amount of freezing time when viewing a given video flow, with the following assumptions: let T_i be the expected arrival time of frame i and let $\hat{T}_i$ be its actual arrival time. The arrival delay Δ_i for frame i is given by $\Delta_i = (\hat{T}_i - \hat{T}_{i-1}) - (T_i - T_{i-1})$. A positive value of Δ_i corresponds to a late arrival of frame i. If this frame delay exceeds the re-buffering threshold τ of the decoder, the play-out will be delayed and the video will freeze. The re-buffering threshold is a measure of how long we can wait until we consider that the video freezes. As illustrated in Fig. 3.4, the total amount of freezing time for the video can be deduced from a histogram of Δ, for a given re-buffering threshold τ.
- **Frame rate deviation:** Records the pre-defined frame rate minus the actual frame rate. The actual frame rate is the frame rate at the destination.

3.3.2 Simulation Results

To illustrate the feasibility of the proposed traffic balancing concept, we consider six adjacent radio nodes with overlapping coverage as shown in Fig. 3.5, where radio nodes 1 and 6 have two dual mobile devices, and radio nodes 2 to 5 has four dual mobile devices. Note that a dual mobile device is connected to two radio nodes within the overlapping area. All users are downloading low rate video flows. Simulation parameters are described in Table 3.1.

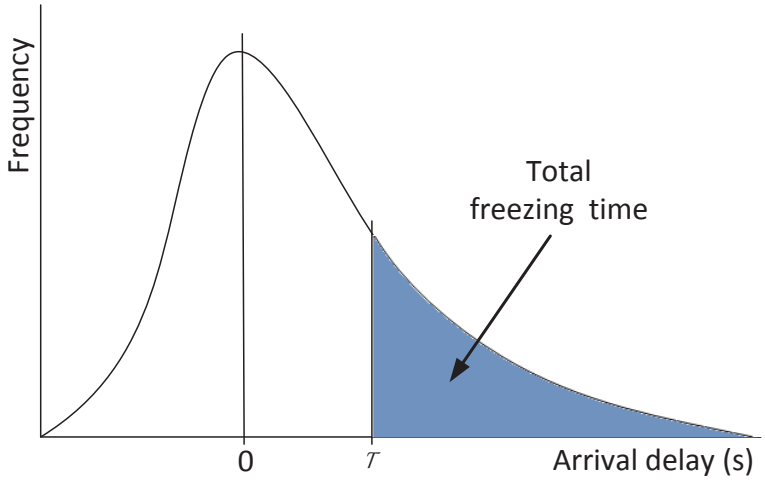


Fig. 3.4 Illustration of the computation of total freezing time from a histogram of arrival delay Δ_i.

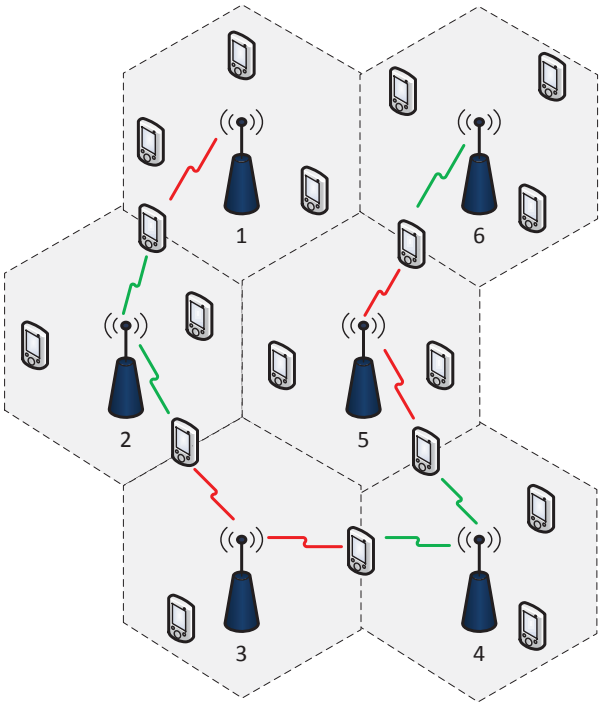


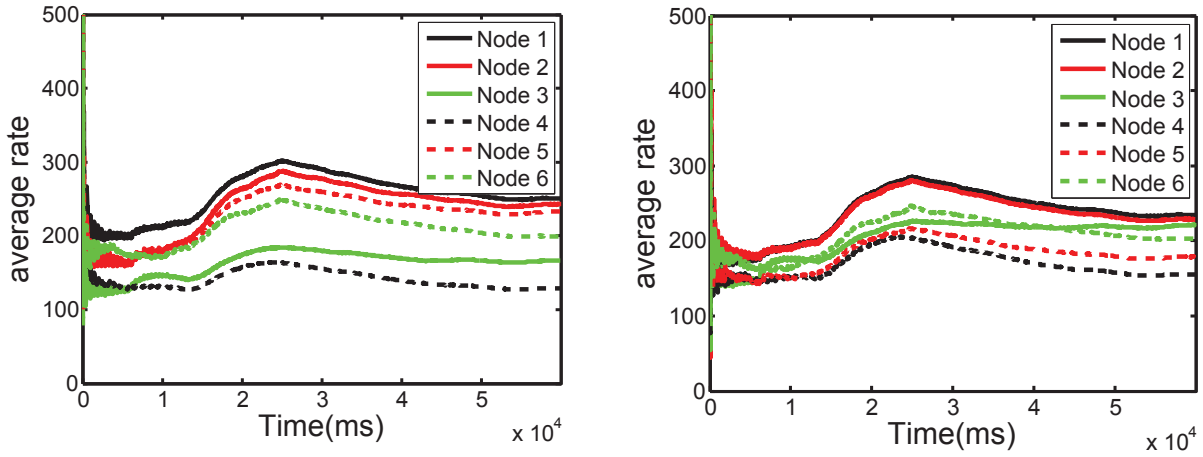
Fig. 3.5 Traffic balancing scenario

Fig. 3.6 and 3.7 compares the average rate served by LTE and JSM in scenarios 1 and 2 respectively. In the figures we can see that JSM tries to balance the traffic according to Algorithm 2. It is also worth noticing that since every radio node has different number of

Parameter	Scenario 1	Scenario 2	Scenario 3
Bandwidth	10 MHz	10 MHz	10 MHz
Fading model	Extended Typical Urban model [84]	Extended Typical Urban model	Extended Typical Urban model
User speed	60 Kmph	3 Kmph	3 Kmph
Number of radio nodes	6	6	6
Total number of flows	58	30	30
Flows by radio node	12,10,8,6,10,12 resp.	5	5
Simulation time	60 seconds	10 minutes	10 minutes
Video rate	160 Kbps	160 Kbps	2.4 Mbps

Table 3.1 JSM simulation parameters

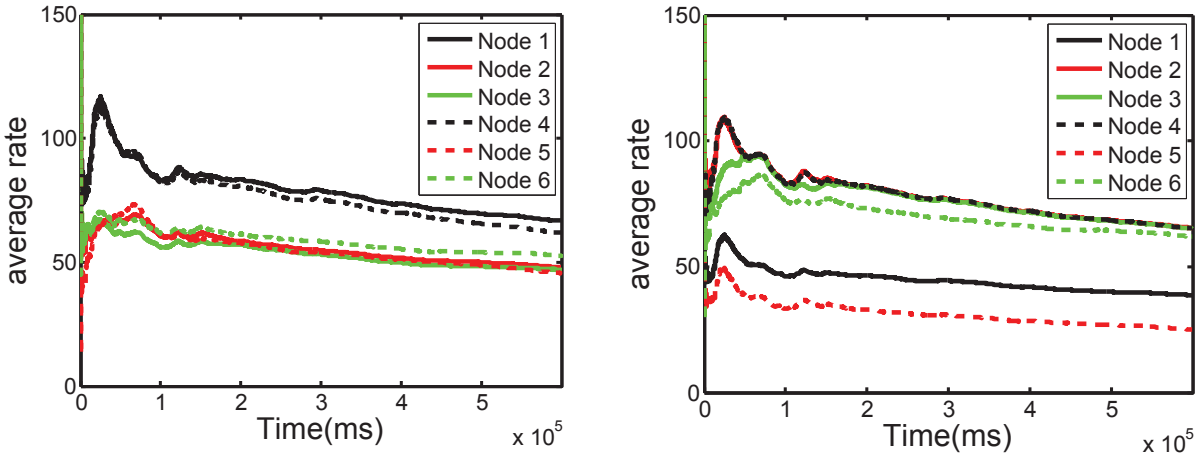
mobile devices attached, and due to the variable nature of video traffic, the total traffic injected into each radio node is different. In addition, the total achievable capacity changes over time due to fading.



(a) LTE (b) JSM

Fig. 3.6 Average rate by radio node (Scenario 1)

In the case of scenario 2, we can observe from Fig. 3.7 that the traffic of radio nodes 2, 3, 4, and 6 have similar average rates, due to the balancing Algorithm 2, nevertheless in order to explain the behavior of radio nodes 1 and 5 we have to analyze the decision metric D_n shown in Fig. 3.8. The decision metric shows that radio node 1 has the lowest value. This can be explained by the fact that its users are experiencing poor radio channel conditions



(a) LTE (b) JSM

Fig. 3.7 Average rate by radio node (Scenario 2)

or downloading larger amounts of traffic. The effect of the low value of the decision metric of radio node 1 directly reflects into its average rate. As a result, Algorithm 2 attempts to increase the decision metric value by reducing the average traffic injected, nevertheless the minimum traffic that radio node 1 can handle is limited by the number of non dual users. Fig. 3.9 show similar results for scenario 3.

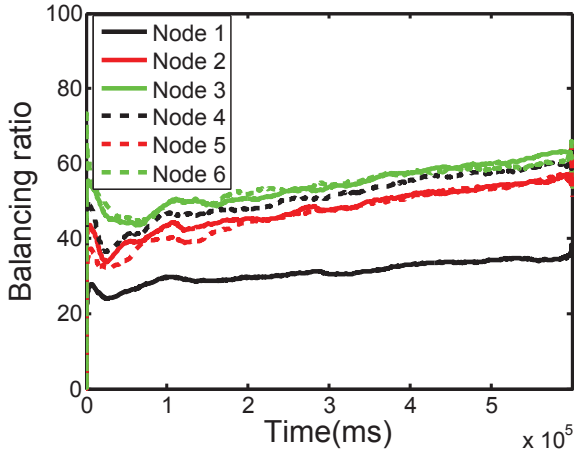
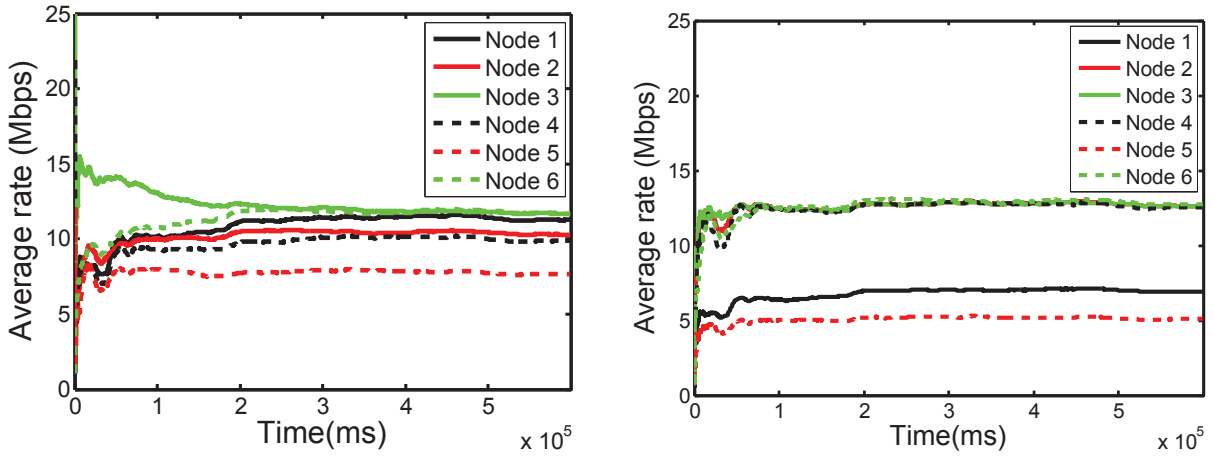


Fig. 3.8 Decision metric D_n (Scenario 2)

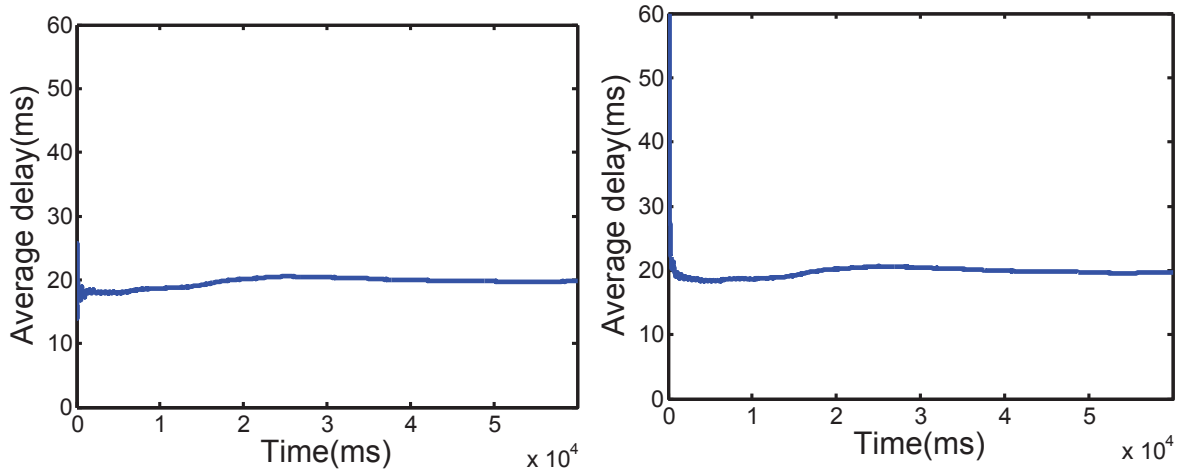
In order to evaluate the performance of JSM, we will discuss some of the QoS metrics defined in Section 3.3. Fig. 3.10 and 3.11 show that the average delay experienced by LTE



(a) LTE (b) JSM

Fig. 3.9 Average rate by radio node (Scenario 3)

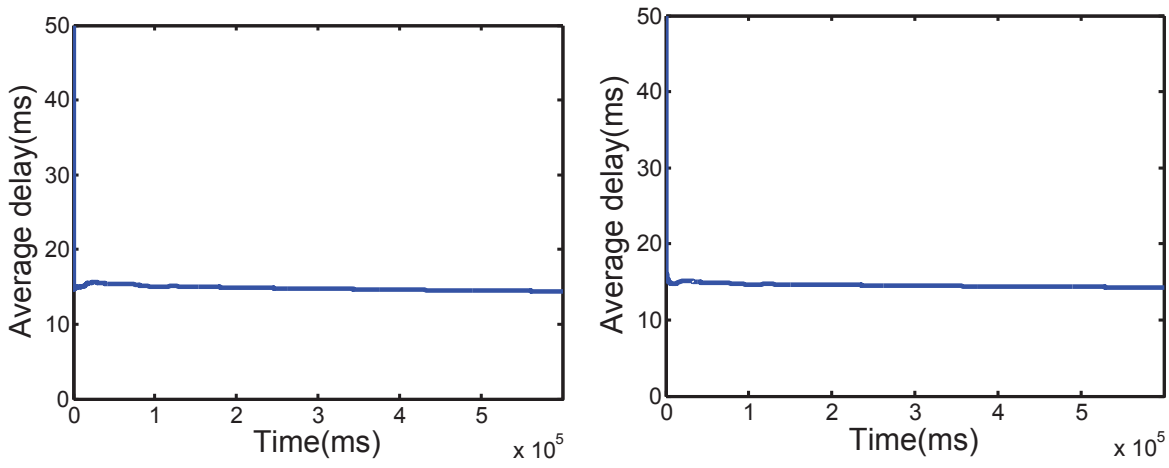
and JSM is comparable for scenarios 1 and 2 respectively. For scenario 1, JSM having a slightly more delay can be seen as the price paid for the extra processing necessary to balance the traffic.



(a) LTE (b) JSM

Fig. 3.10 Average delay (Scenario 1)

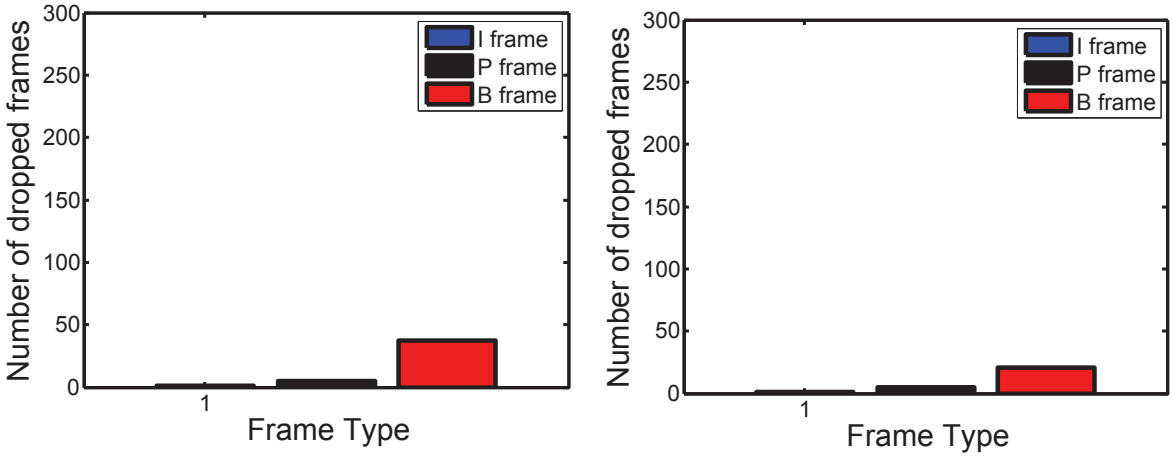
For Scenario 1, Fig. 3.12 shows the number of dropped frames, and Fig. 3.13 shows the number of corrupted frames by type of video frames. In both cases it can be observed that



(a) LTE (b) JSM

Fig. 3.11 Average delay (Scenario 2)

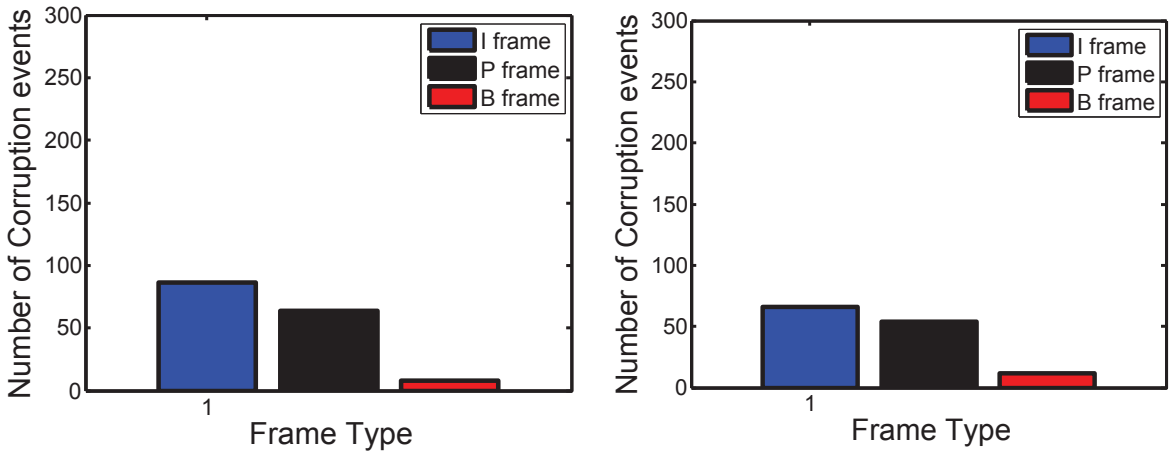
the JSM system behaves similarly to the LTE system. For Scenario 2, Fig. 3.14a and 3.14b show the number of dropped and corrupted packets for the LTE system. Notice that we do not show figures for the JSM system because the number of dropped and corrupted packets is zero. Results for scenario 3 are summarized in Fig. 3.15 and 3.16.



(a) LTE (b) JSM

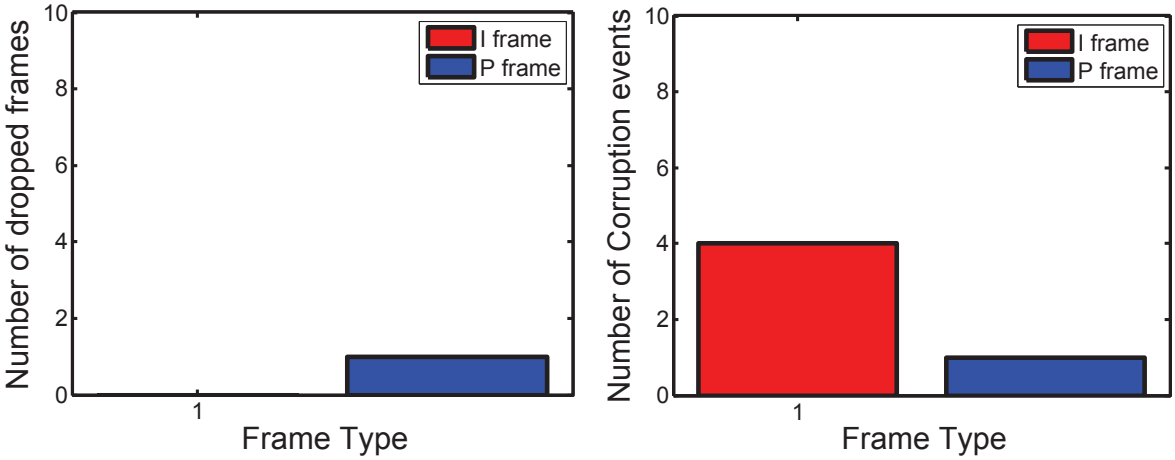
Fig. 3.12 Dropped frames (Scenario 1)

Two other QoS metrics that can be considered are the Frame rate deviation and the



(a) LTE (b) JSM

Fig. 3.13 Corrupt frames (Scenario 1)

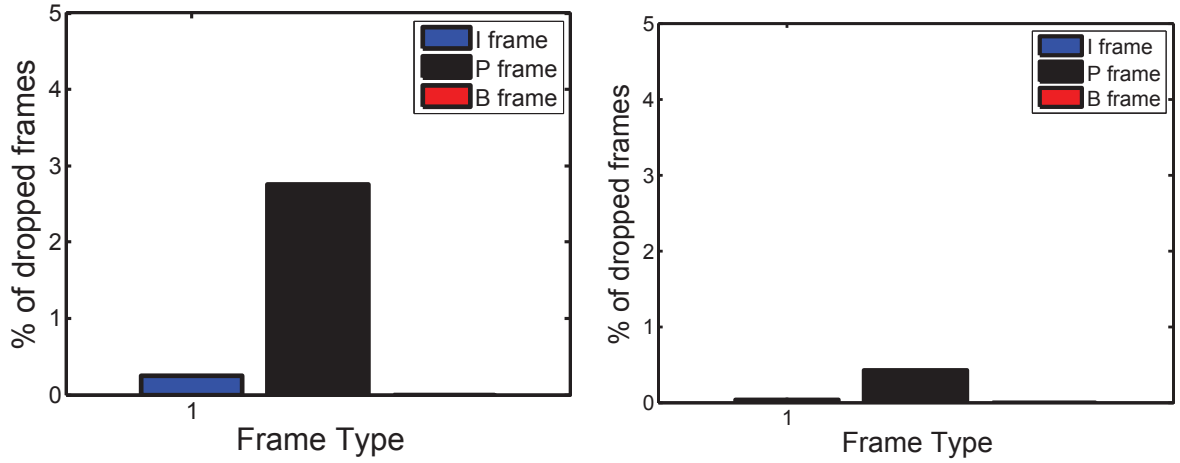


(a) Dropped frames (b) Corrupt frames

Fig. 3.14 Dropped and Corrupt frames LTE (Scenario 2)

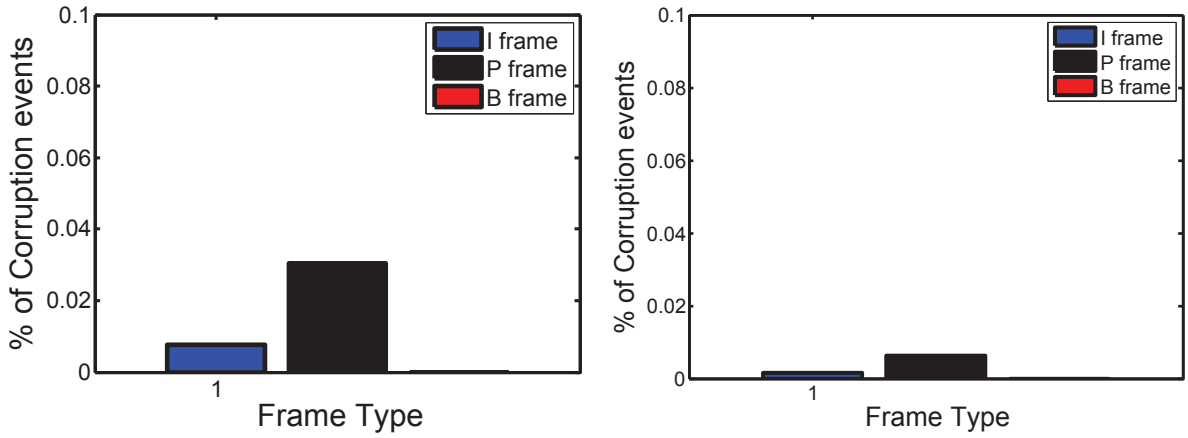
Re-buffering threshold. The values are reported in Table 3.2 and show that the JSM system outperforms the LTE system. This can be explained by the fact that the JSM system routes traffic to the radio nodes having less traffic load and better channel conditions.

For scenario 3, we also have shown in Fig. 3.17 the histogram of packet delays through LTE and JSM systems respectively. The baseline system shows a heavier tail on the right than the JSM system, which shows that packets are more delayed on average. This is



(a) LTE (b) JSM

Fig. 3.15 Dropped frames (Scenario 3)



(a) LTE (b) JSM

Fig. 3.16 Corrupt frames (Scenario 3)

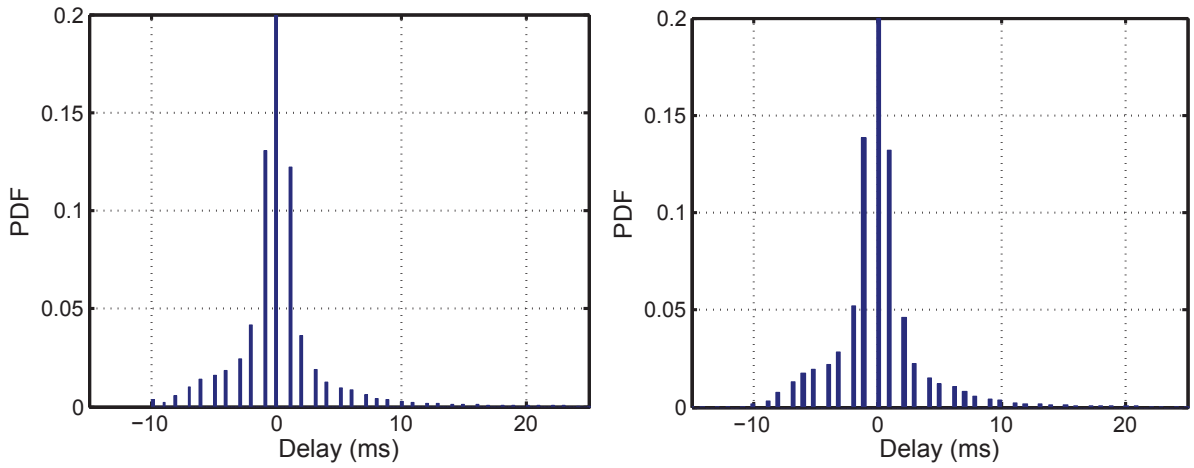
also illustrated in Fig. 3.18 thanks to a cumulative distribution function view of the same histograms.

3.4 Algorithm for the Dynamic Adaptation of Slice Sizes

Another important question to consider is how to determine the slice size. In this section we develop an algorithm that helps to dynamically choose the slice sizes. We define the

	QoS metric	LTE	JSM
Scenario 1	Frame rate deviation	0.0526	0.0409
	Re-buffering threshold (ms)	61	54
Scenario 2	Frame rate deviation	0.0005	0.0008
	Re-buffering threshold (ms)	46	28

Table 3.2 QoS metrics



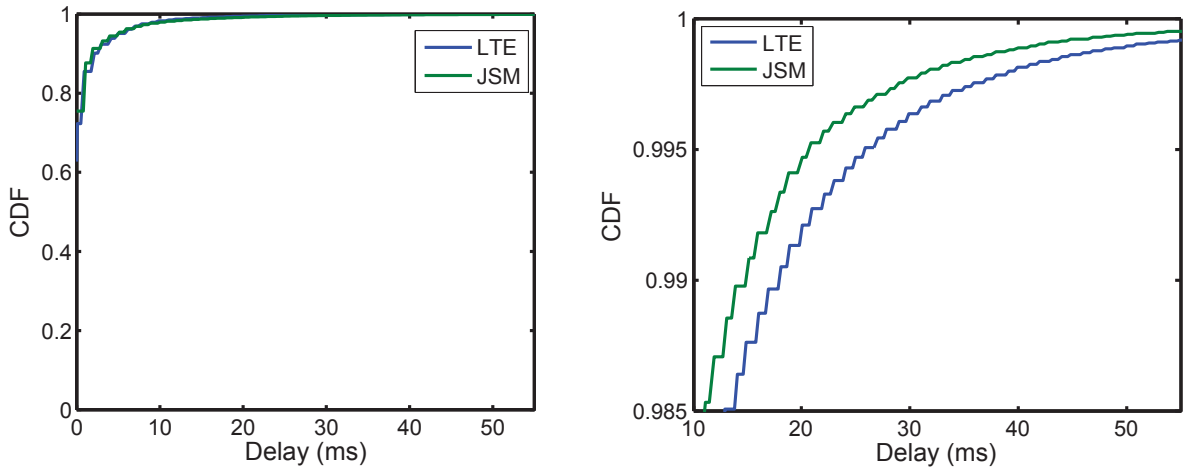
(a) LTE (b) JSM

Fig. 3.17 Histograms of delay (Scenario 3). A delay of 0 indicates that the corresponding packet arrived on time; a negative delay indicates that the packet arrived early, and a positive delay indicates that the packets arrived late.

slice allocation problem as a constrained utility maximization problem in the context of downlink wireless networks, subject to the QoS constraints.

3.4.1 Single Radio Node

The proposed algorithm is limited for now to the case in which each mobile device is associated only to one radio node. Let $\mathcal{W}$ be the set of flows, I be the index set of video services (corresponding to videos with short duration, like trailers, and videos with long duration, like full movies), with generic index i. Any given flow $w \in \mathcal{W}$ belongs to a given slice S_i. Let us also assume that each flow w might be composed of many packets p, R_{S_i} is the instantaneous achievable rate (bps) on each slice S_i, and each slice requires bandwidth provisioning. QoS is a function of the video quality that takes values between 0 and 1,



(a) full CDF (b) Zoomed CDF

Fig. 3.18 Cumulative distribution function of the delays for LTE and JSM for scenario 3.

e.g., one practical definition for QoS is to use the penalty metric P_w (see Section 3.2.4). This formulation solves the slice scheduling problem. For the resource management task, existing solutions (WFQ, PF, etc.) can be adopted.

For a given time step t, the maximization of the utilities can be written as follows:

$$\max \sum_{i \in I} U(R_{S_i}) \tag{3.6}$$

subject to:

$$R_{S_i} \geq R_{S_i}^{th} \qquad i \in I \tag{3.7}$$

$$\sum_i R_{S_i} \leq R \tag{3.8}$$

$$P_{IP_i} \geq P_{B_i} \qquad i \in I \tag{3.9}$$

$$\overline{QoS}_{S_i} \geq QoS_{S_i}^{min} \qquad i \in I, \tag{3.10}$$

where

- constraint (3.7) expresses the condition that the rate assigned to each slice is at least

equal to the reserved rate for each slice. $R_{S_i}^{th}$ should be chosen such that:

$$R_{S_i}^{th} = R_{S_i}^{min} \min\{1, P_i/\overline{QoS}_{S_i}\}, \tag{3.11}$$

where $R_{S_i}^{min} = \min\{R_{S_i}, \overline{R_{S_i}}\}$, $\overline{R_{S_i}}$ is the average slice rate taken over the last δ time steps, and $P_i = P_{B_i}/P_{IP_i}$ is defined as the ratio of the probability of unimportant packets P_{B_i} over the probability of important packets P_{IP_i},

- constraint (3.8) ensures that the total assigned rate is smaller than the total available rate R,
- constraint (3.9) ensures that the dropping probability of important packets $(1 - P_{IP})$ is smaller than the dropping probability of unimportant packets $(1 - P_B)$,
- constraint (3.10) guarantees a minimum QoS in all slices, with $\overline{QoS}_{S_i}$ the average quality of experience in slice S_i taken over the last δ time steps, and $QoS_{S_i}^{min}$ the minimum acceptable QoS value of each slice.

We model the utility function as a concave function with respect to R_{S_i}, so we conveniently rewrite equation (3.6) as:

$$\max \sum_{i \in I} U(R_{S_i}) = \max \sum_{i \in I} R_{S_i}^{th} \log(R_{S_i}). \tag{3.12}$$

3.4.1.1 Slice Scheduling Algorithm:

We first define the exponential moving average rate on each slice $R_{S_i}^{exp}$ as:

$$R_{S_i}^{exp}[t] = \gamma R_{S_i}[t] + (1 - \gamma) R_{S_i}^{exp}[t - 1], \tag{3.13}$$

where the index t denotes the time step, the constant parameter γ represents the forgetting function, which takes values between 0 and 1. Then we define a set of weights at every time step as:

$$weight_i = \frac{R_{S_i}^{th}}{R_{S_i}^{exp}} \qquad i \in I. \tag{3.14}$$

The slice scheduling algorithm (see Algorithm 3) defines the size of each slice according to the weights at each time step. Intuitively, the weight represents the marginal utility of each slice with respect to R_{S_i}, and $R_{S_i}^{th}$ help us take into account constraints (3.9) and (3.10). Algorithm 3 works at the slice level, consequently packet level scheduling must be implemented either on JSM or on the radio node. This solution solves equation (3.6) optimally, as we show below, except for the QoS constraint. If we relax the QoS constraint (3.10), we can find the optimal solution to the problem. Nevertheless, the proposed algorithm defines a sub-optimal solution that takes into account QoS by explicitly including it when evaluating $R_{S_i}^{th}$.

Algorithm 3 Slice scheduling

1: Every δ units of time
2: Compute $weight_i$ according to equation (3.14)
3: Normalize weights ($\sum_i weight'_i = 1$, where $weight'_i = \frac{weight_i}{\sum_i weight_i}$)
4: **for** every slice S_i **do**
5: Compute the slice capacity $= weight'_i \times R$
6: Send packets to the radio node according to the allocated slice capacity
7: **end for**

The slice scheduling algorithm described above converges to the optimal solution if the problem is feasible. Using the definition given in equation (3.12) we can rewrite the slice scheduling problem as:

$$\max \sum_{i \in I} R_{S_i}^{th} \log(R_{S_i}) \tag{3.15}$$

subject to:

$$R_{S_i} \geq R_{S_i}^{th} \qquad i \in I \tag{3.16}$$

$$\sum_i R_{S_i} \leq R \qquad i \in I \tag{3.17}$$

$$P_i \leq 1 \qquad i \in I \tag{3.18}$$

$$\overline{QoS}_{S_i} \geq QoS_{S_i}^{min} \qquad i \in I. \tag{3.19}$$

We can also see that when the weights converge $R_{S_i}^{est} \rightarrow R_{S_i}$ the following condition is satisfied:

$$\frac{R^{th}_{S_1}}{R_{S_1}} = \frac{R^{th}_{S_2}}{R_{S_2}} = \cdots = \frac{R^{th}_{S_i}}{R_{S_i}}. \tag{3.20}$$

This is because we assign resources proportional to the weights, which increases or decreases the values of $R^{exp}_{S_i}$ leading the system to a stable position. As a result all slices tend to converge to the same weights.

To solve optimization problem (3.15) we apply a similar procedure as described in [85], where we can find that the necessary and sufficient condition for the optimal solution is:

$$\frac{\partial f(R_{S_i})}{\partial R_{S_i}} \geq \frac{\partial f(R_{S_{i^*}})}{\partial R_{S_{i^*}}} \qquad i, i^* \in I, R_{S_i} > R^{th}_{S_i}, \tag{3.21}$$

where $f(R_{S_i}) = \sum_{i \in I} R^{th}_{S_i} \log(R_{S_i})$. Taking the derivative we have:

$$\frac{R^{th}_{S_i}}{R_{S_i}} \geq \frac{R^{th}_{S_{i^*}}}{R_{S_{i^*}}} \qquad i, i^* \in I, R_{S_i} > R^{th}_{S_i}. \tag{3.22}$$

We can prove by contradiction that the set of weights $\frac{R^{th}_{S_i}}{R_{S_i}}$ is constant for all $i \in I$. Assume that:

$$\frac{R^{th}_{S_i}}{R_{S_i}} < \frac{R^{th}_{S_{i^*}}}{R_{S_{i^*}}} \qquad i, i^* \in I, \tag{3.23}$$

which implies that $R_{S_i} = R^{th}_{S_i}$ from equation (3.22); if that is true:

$$1 \geq \frac{R^{th}_{S_{i^*}}}{R_{S_{i^*}}} \qquad i, i^* \in I. \tag{3.24}$$

However, if $R_{S_i} = R^{th}_{S_i}$, it contradicts condition (3.23) because $R_{S_{i^*}} \geq R^{th}_{S_{i^*}}$. Therefore the slice algorithm defined with proportional weights converges to the optimal solution.

3.4.2 Generalization for Multiple Radio Nodes with Traffic Balancing

In order to generalize the optimization problem with multiple radio nodes, we now index quantities by both a slice number $i \in I$ and a radio node number $n \in \mathcal{N}$. Each slice is virtually split between the different radio nodes, so that the rate allocated to slice i is the sum of the rates allocated to slice i over all nodes $n \in \mathcal{N}$. The optimization problem is then written as follows:

$$\max \sum_{n \in \mathcal{N}} \sum_{i \in I} U(R_{S_{n,i}}) \tag{3.25}$$

subject to:

$$R_{S_{n,i}} \geq R^{th}_{S_{n,i}} \qquad n \in \mathcal{N}, i \in I \tag{3.26}$$

$$R_{S_n} \leq R_n \qquad n \in \mathcal{N}, i \in I \tag{3.27}$$

$$\overline{R_{S_n}} \leq \frac{R_n}{K} \qquad n \in \mathcal{N}, i \in I \tag{3.28}$$

$$P_{IP_{n,i}} \geq P_{B_{n,i}} \qquad n \in \mathcal{N}, i \in I \tag{3.29}$$

$$\overline{QoS}_{S_{n,i}} \geq QoS^{min}_{S_i} \qquad n \in \mathcal{N}, i \in I, \tag{3.30}$$

where

- constraint (3.26) expresses the condition that the rate $R_{S_{n,i}}$ assigned to each slice is at least equal to the reserved rate for each slice $R^{th}_{S_{n,i}}$. $R^{th}_{S_{n,i}}$ should be chosen such that:

$$R^{th}_{S_{n,i}} = R^{min}_{S_{n,i}} \min\{1, P_{n,i}/\overline{QoS}_{S_{n,i}}\} \tag{3.31}$$

  and

$$R^{min}_{S_{n,i}} = \min\{R_{S_{n,i}}, \overline{R_{S_{n,i}}}\} \tag{3.32}$$

  $\overline{R_{S_{n,i}}}$ is the average slice rate taken over the last δ time steps, and $P_{n,i} = P_{B_{n,i}}/P_{IP_{n,i}}$;

- constraint (3.27) ensures that the total assigned rate is smaller than the total achievable rate on radio node n;

- constraint (3.28) ensures that the average rate managed by each radio node is pro-

portional to the total achievable rate R_n, where $R_{S_n} = \sum_i R_{S_{n,i}}$, K is a parameter that ensures traffic balancing, and can be chosen at iteration t such that:

$$K[t] = \frac{\overline{R_n[t-1]}}{\overline{R_{S_n}[t-1]}} \tag{3.33}$$

where $\overline{R_n[t-1]}$ and $\overline{R_{S_n}[t-1]}$ are averages taken over all nodes;

- constraint (3.29) ensures that the dropping probability of important packets $(1-P_{IP})$ is smaller than the dropping probability of unimportant packets $(1-P_B)$;
- constraint (3.30) guarantees a minimum QoS in all slices, with $\overline{QoS}_{S_{n,i}}$ the average quality of experience in slice S_i and radio node n taken over the last δ time steps, and $QoS_{S_i}^{min}$ the minimum acceptable QoS value on each slice.

We model the utility function as a concave function with respect to $R_{S_{n,i}}$, so we conveniently rewrite equation (3.25) as:

$$\max \sum_{n\in\mathcal{N}} \sum_{i\in I} U(R_{S_{n,i}}) = \max \sum_{n\in\mathcal{N}} \sum_{i\in I} R_{S_{n,i}}^{th} \log(R_{S_{n,i}}) \tag{3.34}$$

3.4.2.1 Multi-slice Scheduling Algorithm:

We first define the exponential moving average rate on each slice $R_{S_{n,i}}^{exp}$,

$$R_{S_{n,i}}^{exp}[t] = \gamma R_{S_{n,i}}[t] + (1-\gamma) R_{S_{n,i}}^{exp}[t-1] \tag{3.35}$$

where the index t denotes the time step, the constant parameter γ represents the forgetting function, which takes values between 0 and 1. Then we define a set of weights at every time step as:

$$weight_{n,i} = \frac{R_{S_{n,i}}^{th}}{R_{S_{n,i}}^{exp}}, \qquad n \in \mathcal{N}, i \in I \tag{3.36}$$

The multi-slice scheduling algorithm (see Algorithm 4) defines the size of each slice in each radio node n according to the weights at each time step. Intuitively, the weight represents the marginal utility of each slice with respect to $R_{S_{n,i}}$, and $R_{S_{n,i}}^{th}$ help us take into

account constraints (3.29) and (3.30). Algorithm 4 works at the slice level, consequently packet level scheduling must be implemented on JSM. This solution solves equation (3.25) optimally, in a similar way as done for the single radio node case.

Algorithm 4 Multi-slice scheduling

1: Every δ units of time
2: Compute $weight_{n,i}$ according to equation (3.36)
3: Normalize weights ($\sum_i weight'_{n,i} = 1$, where $weight'_{n,i} = \frac{weight_{n,i}}{\sum_i weight_{n,i}}$)
4: Compute K according to equation (3.33)
5: Calculate $R_{S_n} = \frac{R_n}{K}$
6: **for** every slice $S_{n,i}$ **do**
7: Compute the slice capacity $R_{S_{n,i}} = weight'_{n,i} \times R_{S_n}$
8: Send packets to the radio node according to the allocated slice capacity
9: **end for**

3.4.3 Simulation Results

In order to illustrate the feasibility of the proposed solutions, we provide simulation results conducted using MATLAB.

3.4.3.1 Single Radio Node

The scenario described here simulates an end-to-end wireless communication system with a single radio node, where 10 users are downloading UDP video traffic from the Internet. 5 users start downloading low quality video streams at different random times (long videos). The other 5 users are downloading low quality video streams of shorter duration, in an attempt to stress the wireless network. The following parameter is used: $T_L = 700$ ms is the maximum sojourn time that a packet can spend within the transit slice. This scenario considers an static positioning model. The simulation runtime is fixed for 5 seconds.

In this scenario we set the parameter $delay^{max} = 20$ ms as the maximum sojourn time that a packet can spend within the system. Only two slices, transit and steady are considered; the total available rate on radio node 1 is constant.

Fig. 3.19 draws the average rate and the dropping rate for radio node 1 with respect to time. It can be seen from the figure, that the average rate is greater, when the optimal solution proposed in Algorithm 3 is used; which naturally leads to a reduced dropping rate.

We can also observe in Fig. 3.20 and 3.21, the resources used and reserved on each slice. Fig. 3.20 shows how traffic is allocated on each slice. Notice that, due to a more optimal assignment of resources, the average traffic on each slice is increased. On the other hand, we can observe in Fig. 3.21 that because the number of resources is constant when more resources are reserved for transit slice less resources are reserved for steady slice.

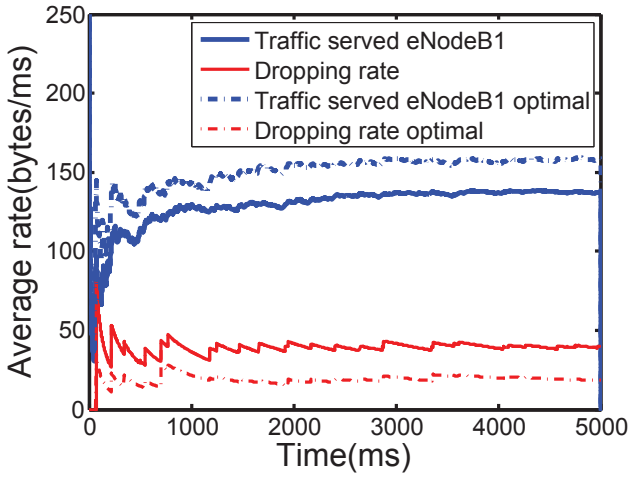


Fig. 3.19 Average served rate and average dropping rate by radio node 1

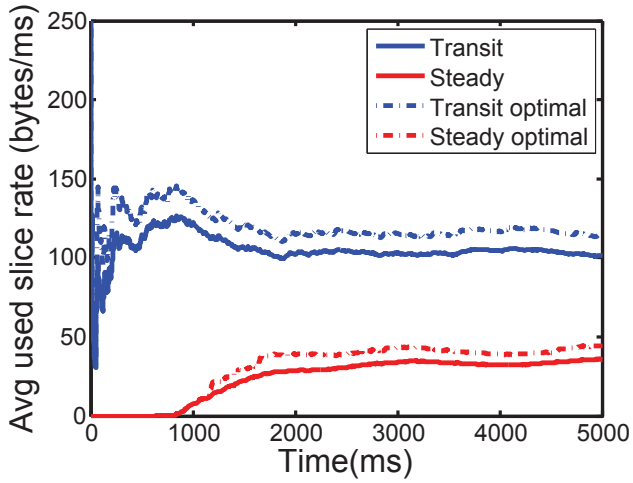


Fig. 3.20 Average served rate in transit and steady slices R_{S_i}

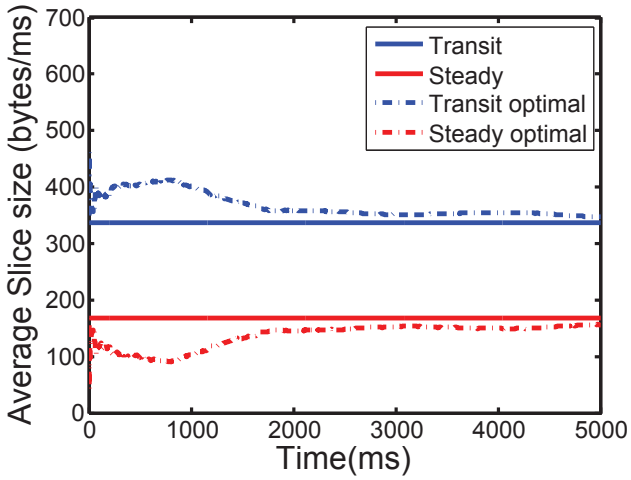


Fig. 3.21 Average reserved rate in transit and steady slices $R_{S_i}^{th}$

3.4.3.2 Multiple Radio Nodes

In this scenario we simulate an end-to-end wireless communication system with 2 radio nodes, each of them has 10 non-dual users and 10 dual users (mobile devices attached to two radio nodes) downloading UDP video traffic from the Internet. Users, start downloading low quality video streams at different random times (long videos). This scenario considers a constant position mobility model. The simulation runtime is fixed for 60 seconds. Only two slices transit and steady are considered, and the total available rates at radio nodes 1 and 2 are constant.

Fig. 3.22 shows the average rates with respect to time. It can be observed from the figure that the average rates are balanced when the solution proposed by Algorithm 4 is used. Additionally, we can see that balancing the traffic reduces the dropping rate by assigning resources more efficiently.

Fig. 3.23 shows the resources reserved for each slice in each radio node. It can be observed in this figure that our proposed algorithm assigns resources dynamically, as compared to a system that relies on fixed slice sizes.

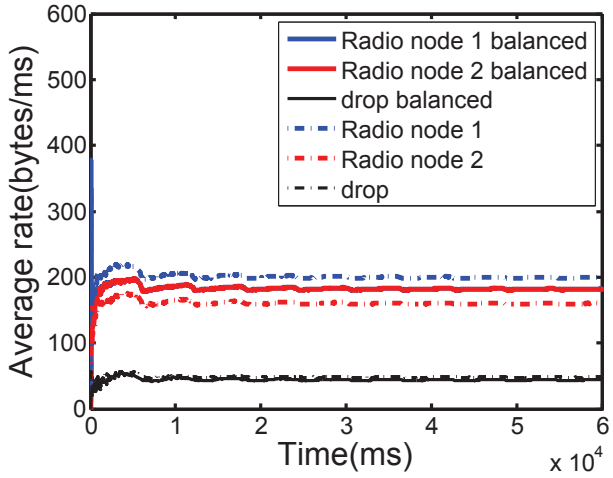
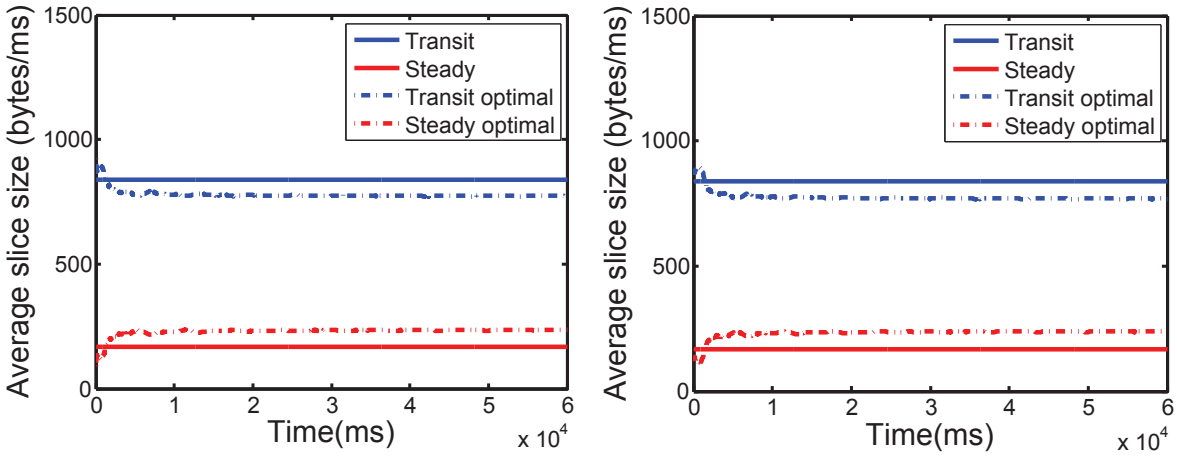


Fig. 3.22 Average served rate and average dropping rate at radio nodes 1 and 2



(a) Radio node 1 (b) Radio node 2

Fig. 3.23 Average reserved rate in transit and steady slices.

3.5 Summary

In this chapter, we have developed JSM, a flow management framework over wireless networks. While the problem of video traffic management has received significant attention in recent years, to the best of our knowledge, this is the first work that designs an SDN-based solution over wireless networks considering that mobile devices can be connected simultaneously to multiple radio nodes.

Although JSM performance evaluation is currently focused only on an LTE wireless network, building JSM over the SDN framework enables us to extend it over other wireless technologies. Additionally JSM does not require any modifications to the mobile device (client) or the server, facilitating a quicker implementation.

To evaluate JSM, we have implemented our solution on an LTE network, using the NS-3 network simulator. Our results show significant improvements not only in video quality, but also in radio node's utilization. In summary, our contributions are:

1. In the control plane, the flow manager is configured to classify the packet flow, serviced by more than one radio node, to a specific slice according to its nature (short or long), the available capacity of each slice, and the feedback information provided by the radio nodes. It also has the ability to reclassify the packet flow according to the traffic load, and the feedback information provided by the radio nodes.

2. The data plane is operatively coupled to the control plane, the data plane is configured to store the packets from the flows according with the classification information provided by the control plane.

3. Our network contribution includes the design of the first traffic management framework for wireless networks with the capability to give service to mobile devices connected simultaneously to more than one radio node of the same technology.

One important question to consider, when using mobile devices connected to multiple radio nodes, is how to split traffic (schedule packets) such that, the traffic load on each radio node remains at a given desired load, e.g., balance the traffic equally among all the radio nodes. In the next chapter, we address this question by modelling the problem as a constrained optimization problem and developing practical sub-optimal heuristic solutions.

Chapter 4

Uplink Multipath Load Balancing

4.1 Introduction

Advances in technology have made it possible for a mobile device to handle more than one radio interface at the same time. The presence of several radio interfaces allows mobile devices to use simultaneously multiple paths to establish connections to the destination. Fig. 4.1 depicts a basic uplink load balancing scenario in which a mobile device is attached to two radio nodes of potentially different technologies.

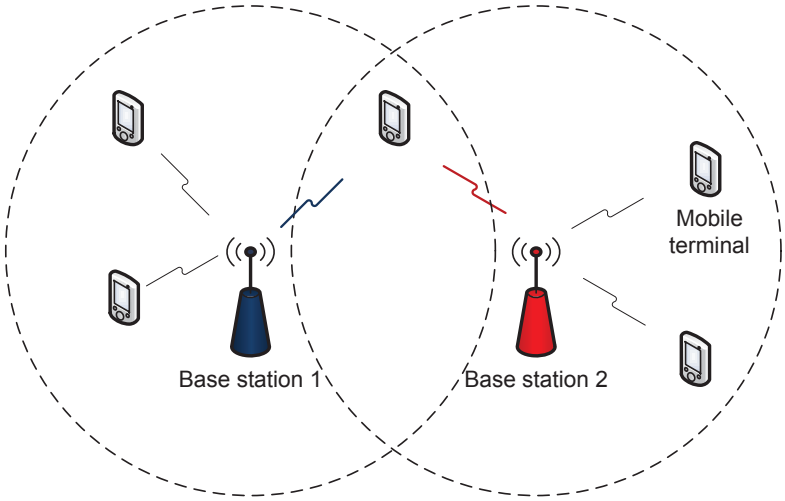


Fig. 4.1 Load balancing scenario: Mobile device attached to two radio nodes simultaneously.

The main research challenge in utilizing multiple paths in the context of uplink wireless networks is to split input traffic so as to provide acceptable QoS perceived by end users. Inefficient load balancing can significantly degrade the network performance, thus creating large end-to-end delay, packet reordering, etc.

The purpose of load balancing is to optimize resource use, i.e. minimize end-to-end delay and maximize throughput. Using multiple paths to the destination with load balancing instead of a single path may help to increase availability and reliability through redundancy. Additionally the ability to use multiple paths simultaneously increases the available bandwidth [86,87]. This Chapter develops three optimization models for the load balancing problem and proposes practical sub-optimal heuristic solutions that can be easily implemented.

4.2 Notation and Assumptions

This section describes some common assumptions and notations used in this chapter. We consider the uplink case of a wireless network in which mobile devices have more than one network interface and can be attached to multiple radio nodes at the same time, i.e., traffic generated in a mobile device can be routed to the destination through multiple radio nodes. We assume that each radio node knows the channel capacity; that is the maximum rate at which a mobile device can send traffic. Channel capacity information is reported to every mobile device by the radio nodes.

Let $\mathcal{N}$ be the set of radio nodes, $\mathcal{M}$ the set of mobile devices and $\mathcal{W}$ the set of flows. Path n refers to the connection between mobile device m and the destination passing through radio node n. At a given point in time, let $\mathcal{W}_m \in \mathcal{W}$ be the set of flows in mobile device m. $\mathcal{N}_m$ the set of radio nodes to which mobile device m is attached to. $\mathcal{P}_w$ be the set of packets of flow w in mobile m's buffer ready to be scheduled.

Let G_n^m be the channel capacity, $d_{n,w}^m$ the end-to-end delay as seen by flow w on path n and mobile device m, and $\hat{R}_n^m[t]$ be the rate estimate of the traffic sent from mobile device m to radio node n at time t. For simplicity, we suppress the explicit dependence on time t in the notation. We also define the desired load K_n^m as the system parameter that represents the amount of traffic expected to be transmitted by mobile m on path n such that it keeps the load at the desired level; it is assumed to be known, e.g., if three radio nodes are expected to receive the same amount of traffic from mobile m then $K_n^m = \frac{1}{3}\sum_n \hat{R}_n^m$.

4.3 Evaluation Metrics

- **Splitting error (SE)** measures the deviation of the desired load with respect to the actual load in each path. The smaller the value is, the more accurate load balancing we have. We can calculate the splitting error as:

$$SE = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{N}_m|} \sum_{n \in \mathcal{N}} \frac{|K_n^m - \hat{R}_n^m|}{K_n^m}. \tag{4.1}$$

- **End-to-end delay (d)** measures the time taken for a packet to be transmitted across a network from source to destination. It includes the transmission delay d_{tx}, the propagation delay d_p, the processing delay d_{pc} and the queuing delay d_q.

$$d = d_{tx} + d_p + d_{pc} + d_q. \tag{4.2}$$

- **Reorder Density (RD)** [88], this metric measures the amount of packet reordering in the arriving packet sequence. RD metric is defined as the distribution of packet displacements ($\sum_i RD(i) = 1$). Negative displacements in the RD distribution represent the number of positions by which the received packet is early, while positive displacements represent the lateness of the received packets.

- **Mean displacement of packets (M_D)** [89]. When calculating the mean of the reordering density RD, if all packets are included, the mean becomes zero. Therefore, the mean, when all packets are taken together, is not useful. Instead we can consider the magnitude to define a mean displacement M_D:

$$M_D = \sum_{i=-DT}^{+DT} |i| RD(i). \tag{4.3}$$

  where the displacement threshold DT is a threshold on the displacement of packets that allows the metric to classify a packet as lost or duplicate.

- **Reorder entropy (E_R)** [89]. Entropy is a concept that is used to define the randomness or the disorder and can be used as a convenient metric for a distribution's tendency to be concentrated or dispersed. As RD is a discrete probability distribu-

tion, that of packet disorder, we can define reorder entropy as:

$$E_R = -\sum_{i=-DT}^{+DT} RD(i) \ln RD(i), \tag{4.4}$$

If there is no packet reordering, i.e., $RD(0) = 1$, the reorder entropy is equal to zero. On the other hand, if the packet sequence has the highest variance, packets are displaced uniformly with equal probabilities. Then the upper bound for the reorder entropy is $\ln(2DT + 1)$.

- **Peak signal-to-noise ratio (PSNR)** is a metric for video quality that can be defined as the ratio between the maximum possible value (power) of the original video signal and the power of the distorting noise that affects the quality of its representation at the receiver. This metric is measured using Evalvid open source tool [90].

4.4 Dynamic Load Balancing

In this section, we propose QBALAN[1], a new load balancing algorithm that uses two main strategies: the long-term strategy splits traffic that remains on the system for long periods of time, and the short-term strategy splits traffic such that packet reordering is minimized. Numerical results show that our algorithm reduces the splitting error while reducing packet reordering.

4.4.1 Problem Statement

We propose an optimization model for the load balancing problem in the context of uplink wireless mobile networks, subject to the packet reordering constraint. It corresponds to a constrained optimization problem in which the objective is the maximization of a utility function that captures the reordering concern.

For a given transmission time interval (TTI), any given flow $w \in \mathcal{W}$ can be assigned for scheduling to only one radio node n. The goal is to find the right packet distribution such that we split traffic among radio nodes minimizing splitting error and packet reordering,

[1]Oscar Delgado and Fabrice Labeau, "Uplink load balancing over multipath heterogeneous wireless networks", IEEE Vehicular Technology Conference (VTC), pp. 686-691, (Glasgow, Scotland), May 2015

see Fig. 4.2. The task of assigning flows to paths can be subdivided into long-term load balancing and short-term load balancing strategies.

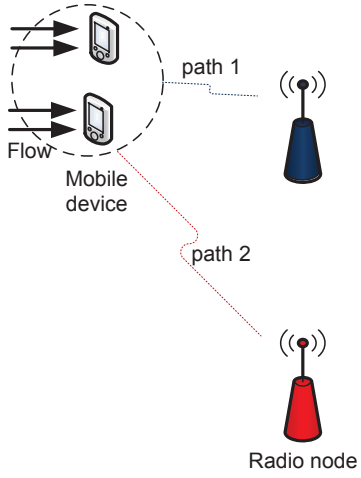


Fig. 4.2 Basic scenario: Mobile devices attached to two radio nodes, every flow takes one of the two possible paths to the destination.

4.4.1.1 Long-term Load Balancing

The long-term load balancing objective is to assign a subset of all flows to paths such that the radio nodes' traffic load remains as balanced as possible. We randomly choose, in every mobile device, half of the available flows to be the ones handled by the long-term load balancing, while the remaining flows will be handled by the short-term load balancing.

Let $\mathcal{W}_m^L$ be the subset of flows in mobile device m to be handled by the long-term load balancing algorithm, $\alpha_{w,n} \in \{0,1\}$ be the flow assignment indicator, $\alpha_{w,n} = 1$ if flow w is assigned to radio node n, $\hat{R}_n^m[t]$ be the rate estimate of the traffic sent to radio node n at time t, which can be defined as a function of the flow assignment indicator $\alpha_{w,n}$. $\hat{R}_n^m[t]$ is constantly being updated by each mobile device m as the exponential moving average [91] with parameter γ:

$$\hat{R}_n^m[t] = \gamma \sum_{w \in \mathcal{W}_m} \alpha_{w,n} R_w[t] + (1-\gamma)\hat{R}_n^m[t-1] \forall n \in \mathcal{N}_m, \tag{4.5}$$

where $\hat{R}_n^m[t-1]$ is the rate estimate in the previous TTI and $R_w[t]$ is the instantaneous flow rate. An exponential moving average $\hat{R}_n^m = \hat{R}_n^m[t]$ as defined in equation (4.5) is used to

smooth out short-term fluctuations and highlight longer-term trends. The long-term load balancing optimization is executed every δ_L ms. The objective is to find the value of $\alpha_{w,n}$ that minimizes the traffic load difference among paths:

$$\min_{\alpha_{w,n}:w\in\mathcal{W}_m^L}\left\{\max_{n\in\mathcal{N}_m}\left(1-\frac{\hat{R}_n^{m^{\text{est}}}}{0.8G_n^m}\right)\right\} \qquad \forall m\in\mathcal{M}, \tag{4.6}$$

subject to:

$$\sum_{n\in\mathcal{N}_m}\alpha_{w,n}=1 \qquad \forall w\in\mathcal{W}_m^L \tag{4.7}$$

$$\exists n\in\mathcal{N}_m:\hat{R}_n^{m^L}<0.8G_n^m, \tag{4.8}$$

where

- $\hat{R}_n^{m^{\text{est}}}=\hat{R}_n^{m^{\text{est}}}[t]$ is the rate estimate used during the long-term balancing optimization. The latter is defined as:

$$\hat{R}_n^{m^{\text{est}}}[t]=\sum_{w\in\mathcal{W}_m^L}\alpha_{w,n}\gamma R_w[t]+(1-\gamma)\hat{R}_n^m[t-1]+\gamma R_n^{m^S}[t-1]\forall n\in\mathcal{N}_m, \tag{4.9}$$

- $R_n^{m^S}[t-1]$ refers to the rate assigned in the short-term load balancing optimization during the previous TTI. $R_n^{m^S}[t]$ is defined as:

$$R_n^{m^S}[t]=\sum_{w\in\mathcal{W}_m^S}\alpha_{w,n}R_w[t] \qquad \forall n\in\mathcal{N}_m, \tag{4.10}$$

- $\hat{R}_n^{m^L}=\hat{R}_n^{m^L}[t]$ is the rate estimate of the flows handled by the long-term load balancing algorithm. The latter is defined as:

$$\hat{R}_n^{m^L}[t]=\sum_{w\in\mathcal{W}_{m^L}}\alpha_{w,n}\gamma R_w[t]+(1-\gamma)\hat{R}_n^{m^L}[t-1]\forall n\in\mathcal{N}_m. \tag{4.11}$$

- constraint (4.7) refers to the fact that a given flow can only be assigned to one path at a time.

- constraint (4.8) ensures that flows are only assigned to a specific path if the corre-

sponding rate $\hat{R}_n^{m^L}$ will not exceed the maximum rate G_n^m. The 0.8 factor is used to reduce the risk of saturating G_n^m. The system is not very sensitive to this factor, but it should allow the network to handle the long-term load balancing as well as to reserve traffic for the short-term load balancing.

In order for (4.6) - (4.8) to have a solution, there should exist at least one n that satisfies (4.8). If there is no feasible solution, packets of flow $w \in \mathcal{W}_m^L$ will be assigned to a path according to the short-term load balancing algorithm. Radio node n sends G_n^m information to every mobile device at least every δ_L ms.

4.4.1.2 Short-term Load Balancing

The objective is to assign the flows not being handled by the long-term load balancing algorithm, to paths such that the actual load remains as close to the desired load as possible.

Let $\mathcal{W}_m^S$ be the subset of flows in mobile device m to be handled by the short-term load balancing algorithm. The short-term load balancing optimization is executed every TTI. The objective is to find the value of $\alpha_{w,n}$ that:

$$\min_{\alpha_{w,n}:w\in\mathcal{W}_m^S} \left\{ \max_{n\in\mathcal{N}_m} \left(1 - \frac{\hat{R}_n^m}{G_n^m} \right) \right\} \qquad \forall m \in \mathcal{M}, \tag{4.12}$$

subject to:

$$\sum_{n\in\mathcal{N}_m} \alpha_{w,n} = 1 \qquad \forall w \in \mathcal{W}_m^S \tag{4.13}$$

$$|d_{n,w}^m - d_{n',w}^m| < \varepsilon_w^m \qquad \forall n, n' \in \mathcal{N}_m, \forall w \in \mathcal{W}_m^S \tag{4.14}$$

$$\exists n \in \mathcal{N}_m : \hat{R}_n^m < G_n^m, \tag{4.15}$$

where (4.14) is the delay constraint, $d_{n,w}^m$ and $d_{n',w}^m$ are the end-to-end delays as seen by flow w on paths n and n' respectively. The threshold ε_w^m is the maximum tolerable delay variation between paths that can be estimated as the inter-arrival time of flow w. Delay values should be reported by radio nodes or by higher layers every δ_S ms. If packets of a flow can not be assigned to any radio node, those packets will be discarded.

4.4.2 QBALAN Algorithm

The load balancing algorithm QBALAN aims to minimize packet reordering while splitting traffic in a balanced way. One of the advantages of subdividing the load balancing task into long-term and short-term load balancing algorithms is that it reduces the average number of times a flow switch paths which in practice causes an increase in packet reordering. Additionally, by using the short-term load balancing algorithm, that is executed every TTI, the system maintains the ability to quickly adapt to the load balancing requirements. The QBALAN algorithm is a heuristic way of solving the optimization problems laid out in the previous section and it has two main components.

4.4.2.1 Long-term Load Balancing

QBALAN uses a long-term strategy to reduce packet reordering. The strategy consists in assigning some of the flows to specific paths. This assignment is updated every δ_L ms.

Long-term load balancing allows QBALAN to assign flows to paths minimizing packet reordering. Algorithm 5 summarizes the long-term load balancing strategy. Every δ_L ms the long-term load balancing algorithm is executed. During the execution of the algorithm we update $\hat{R}_n^{m^{now}}$ according to:

$$\hat{R}_n^{m^{now}} = \hat{R}_n^{m^{last}} + R_w[t]. \tag{4.16}$$

The update equation for $\hat{R}_n^{m^{last}}$ takes into account the fact that flows are being assigned to radio nodes. In order to appropriately choose the path to be used by every flow, we calculate the deficit $U_n^{m^L}$ as the normalized difference between the maximum rate that mobile device m can transmit and its actual rate. The 0.8 factor is used to reduce the risk of saturating G_n^m.

$$U_n^{m^L} = 1 - \frac{\hat{R}_n^{m^{now}}}{0.8 G_n^m}. \tag{4.17}$$

Flows are assigned to the path n_* with smaller load.

$$n_* = \arg\max_n \{U_n^{m^L}\}. \tag{4.18}$$

Algorithm 5 Long-term load balancing

Every δ_L ms
Initialize: $\hat{R}_n^{m^{last}} \leftarrow \hat{R}_n^m[t-1]$ and $\hat{R}_n^{m^{\text{temp2}}} \leftarrow \hat{R}_n^{m^L}[t-1]$
Choose the subset $\mathcal{W}_m^L$ (half the total number of flows $|\mathcal{W}_m|$)
for every chosen flow $w \in \mathcal{W}_m^L$ **do**
  Update $\hat{R}_n^{m^{now}}$ according to equation (4.16) for all n
  Update $\hat{R}_n^{m^{\text{temp1}}} \leftarrow \hat{R}_n^{m^{\text{temp2}}} + R_w[t]$ for all n
  Calculate $U_n^{m^L}$ according to equation (4.17) for all n
  if $\exists\ n$ such that $\hat{R}_n^{m^{\text{temp1}}} < 0.8 G_n^m$ **then**
    Evaluate n_* according to equation (4.18)
    Assign flow w to radio node n_*
    if flow w was not assigned to path $n*$ in the previous TTI **then**
      Update $\hat{R}_{n*}^{m^{last}} \leftarrow \hat{R}_{n*}^{m^{now}}$
      Update $\hat{R}_{n*}^{m^{\text{temp2}}} \leftarrow \hat{R}_{n*}^{m^{\text{temp1}}}$
    end if
  else
    Flow w will be assigned in the short-term load balancing algorithm
  end if
end for

4.4.2.2 Short-term Load Balancing

QBALAN uses a short-term strategy to ensure load balancing while minimizing packet reordering. The short-term strategy described in Algorithm 6 consists in assigning flows to paths when it is "safe to do so": the assignment is executed at every TTI, and it ensures that a flow can switch paths only if the difference in the delay between paths is smaller than the threshold ε_w^m. The latter can be estimated as the difference between the current time t and the last time t_w flow w transmitted a packet.

Short-term load balancing is executed in a packet by packet basis. δ_w^m in Algorithm 6 represents the maximum difference in delay on every path and is defined as:

$$\delta_w^m = \max\left(d_{n,w}^m - d_{n',w}^m\right) \qquad n, n' \in \mathcal{N}_m, w \in \mathcal{W}_m^S. \tag{4.19}$$

The parameter δ_w^m is updated at least every δ_S units of time. In order to appropriately chose the path to be used by every packet, we calculate $U_n^{m^S}$ as the normalized difference

between the maximum rate that mobile device m can send and its actual load.

$$U_n^{m^S} = 1 - \frac{\hat{R}_n^m[t-1]}{G_n^m}, \tag{4.20}$$

where $\hat{R}_n^m[t-1]$ is the rate estimate in the previous TTI. Packets are assigned to the path $n_\circ$ with smaller load as:

$$n_\circ = \arg\max_n \{U_n^{m^S}\}. \tag{4.21}$$

Algorithm 6 Short-term load balancing

```
Every TTI
for every flow w ∈ W_m^S do
    Calculate U_n^{m^S} according to equation (4.20) for all n
    if ∃ n such that condition (4.15) is satisfied then
        Calculate ε_w^m = t − t_w
        if δ_w^m ≤ ε_w^m then
            Evaluate n_∘ according to equation (4.21)
            Assign flow w to radio node n_∘
        else
            Keep flow w on the same path
        end if
    else
        Discard packets of flow w
    end if
end for
```

4.4.3 Simulation Results

To evaluate the performance of our proposed algorithm QBALAN, we conduct experiments in MATLAB. In our simulation scenario 10 mobile devices are connected to two radio nodes. Each mobile device generates five different video flows (150Kbps) to send to the radio nodes (UDP traffic). To model the delay of each path between a mobile device and the destination we use a Pareto distribution. The Pareto distribution offers a good compromise between high approximation and low complexity [92]. The load share of the two radio nodes is set

to 0.5. This means that the two radio nodes are expected to receive the same amount of traffic (desired load $K_n^m = \sum_n \hat{R}_n^m/2$).

We present our results only for the case of mobile devices located in a circular-shape area (hot-spot) inside the radio nodes' coverage area (see Fig. 4.2). We locate all mobile devices in a hot-spot such that we can vary the delay difference and the maximum rate ratio between path 1 and path 2.

First we show that QBALAN accurately matches the desired load. The mean values of the splitting error for 50 runs of the same experiment with average delay difference of 35ms and maximum rate ratio of 35/65 are shown in Fig. 4.3. For a quick visual assessment of the data, box-plot diagrams are used. FLARE (Section 2.4.2.1) and QBALAN splitting error lie below 2. Fig. 4.3 shows that QBALAN achieves smaller splitting error. Considering that typical end-to-end delays from coast to coast in North America are less than 40ms, an average delay difference of 35ms should apply to many practical scenarios.

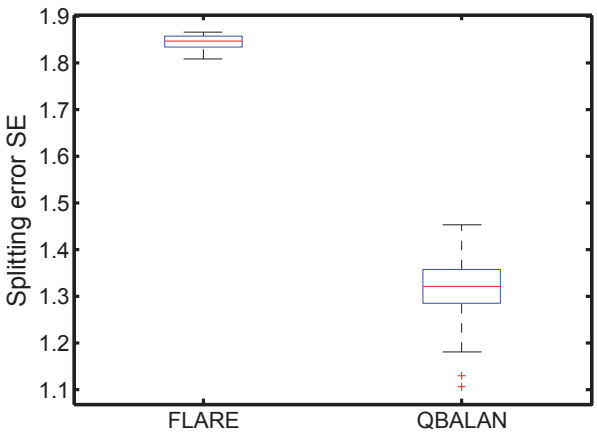


Fig. 4.3 Splitting error, box-plot 50 iterations, average delay difference 35ms.

Fig. 4.4 shows the average splitting error SE as a function of the average delay difference, for the case of a maximum rate ratio of 35/65. It is clear that QBALAN performs better than FLARE. This is because QBALAN uses the long-term load balancing algorithm to split traffic, and then refines the load balancing, by appropriately choosing when to switch paths, with the short-term load balancing algorithm.

An important aspect to consider is the packet reordering. Fig. 4.5 shows the reorder density (RD%) evaluated at time $t = 0$ms as a function of the average delay difference and maximum rate ratio of 35/65. We can observe in the figure that FLARE and QBALAN

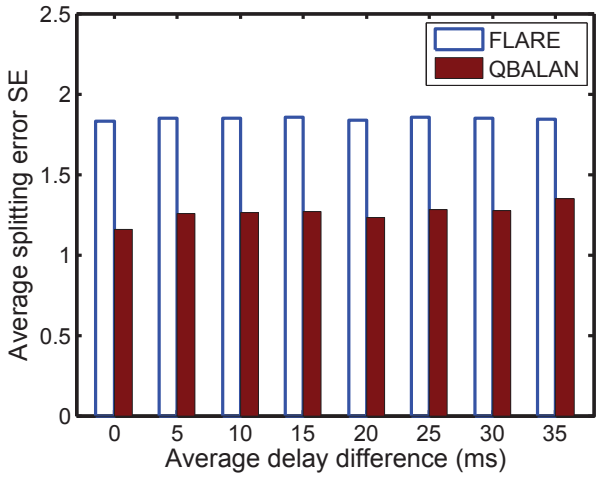


Fig. 4.4 Average splitting error SE versus average delay difference.

concentrate almost all of the packets around time zero even as the delay difference increases. Another metric that can help us understand the level of packet reordering is the reorder

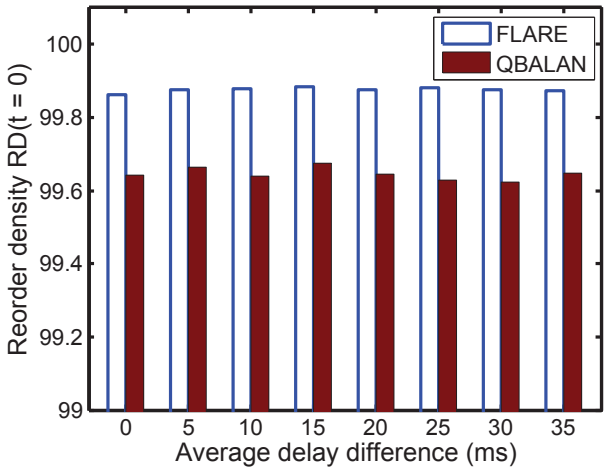


Fig. 4.5 Reorder density (RD%) evaluated at time $t = 0$ versus average delay difference.

entropy depicted in Fig. 4.6 as a function of the average delay difference, for the case of a maximum rate ratio of 35/65. We can see in the figure that QBALAN has a slightly higher reorder entropy. This can be seen as the price paid by QBALAN to have a better splitting error. Notice that the scale is really small, confirming the observation made when looking at the reordering density RD.

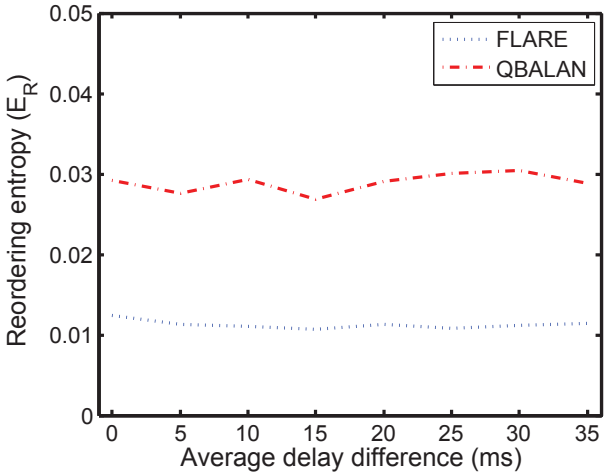


Fig. 4.6 Reorder entropy (E_R) versus average delay difference.

It is also important to analyse the end-to-end delay, as intuitively we know that one way to avoid packet reordering is to buffer the packets so that they can be transmitted on the same path. Fig. 4.7 presents the average end-to-end delay as a function of the average delay difference. In the figure we can observe that QBALAN has better end-to-end delay than FLARE, in the order of 2.5 to 1. Nevertheless, we can also notice that QBALAN makes extensive use of the buffers on each path so as to avoid packet reordering. This behaviour is not desirable for real-time applications which require smaller end-to-end delays.

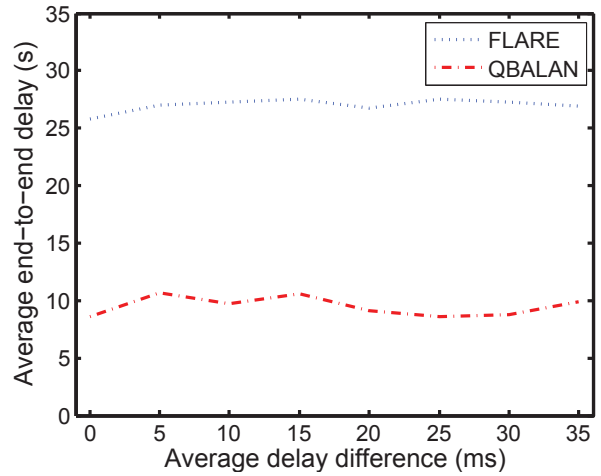


Fig. 4.7 Average end-to-end delay versus the average delay difference.

4.4.4 Discussion

We developed QBALAN a multipath load balancing algorithm for uplink heterogeneous wireless mobile systems that addresses the problem of how to accurately split traffic to multiple paths without introducing packet reordering. Specifically we focus our experiments on the case of real time UDP video traffic applications. Real time applications are sensitive to packet reordering because it increases the possibility of buffer overflow and packet loss due to time-out.

Simulation results show that highly accurate load balancing can be achieved by subdividing the load balancing task into long-term and short-term load balancing algorithms without adding complexity. QBALAN strategy effectively reduces the splitting error by reducing the average number of times a flow switch paths, while not significantly affecting the packet reordering. Nevertheless, another metric that has not been considered directly by QBALAN is the end-to-end delay. The average end-to-end delay is especially important in real-time applications because it directly impacts the quality of service.

4.5 Delay Aware Load Balancing

In this section, we address two key issues in the context of uplink wireless mobile networks: First, how to accurately split traffic among multiple paths, and second, how to minimize the end-to-end delay without increasing packet reordering. We propose the Delay Aware Load Balancing Algorithm (DALBA), a novel strategy that splits traffic at the granularity of packet. DALBA[2] aims to minimize the splitting error and the end-to-end delay by effectively using all the available paths.

4.5.1 Problem Statement

We formulate a multi-objective optimization model for the load balancing problem over multiple paths. In a given TTI, any flow $w \in \mathcal{W}$ can be scheduled through multiple radio nodes $n \in \mathcal{N}$. The main objective is to find the appropriate packet assignment such that we sent traffic to radio nodes minimizing the splitting error and the end-to-end delay while limiting the negative effects of packet reordering. Each mobile device m is attached to

[2]Oscar Delgado and Fabrice Labeau, "Delay aware load balancing over multipath wireless networks", submitted to IEEE Transactions on Vehicular Technology (TVT), 2015

multiple radio nodes, see Fig. 4.8.

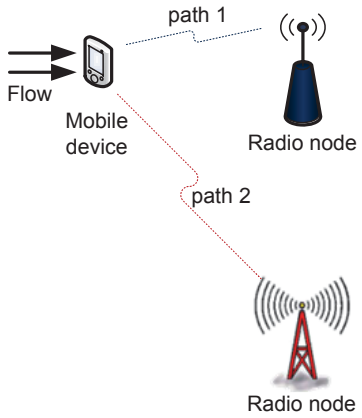


Fig. 4.8 Basic scenario: Mobile device attached to two radio nodes, transmitting two traffic flows.

Let us define $\alpha_{p_w,w,n} \in \{0,1\}$ as the packet assignment indicator, $\alpha_{p_w,w,n} = 1$ if packet $p_w \in \mathcal{P}_w$ is assigned to radio node n. Different from equation (4.5) that defines $\hat{R}_n^m[t]$ as a function of the flow assignment indicator $\alpha_{w,n}$ we now express $\hat{R}_n^m[t]$ as a function of the packet assignment indicator $\alpha_{p_w,w,n}$. $\hat{R}_n^m[t]$ is constantly updated on each mobile device m by an the exponential moving average [91] with parameter $\gamma \in (0,1)$:

$$\hat{R}_n^m[t] = \gamma \sum_{p_w \in \mathcal{P}_w, w \in \mathcal{W}_m} \alpha_{p_w,w,n} R_{p_w,w}[t] + (1-\gamma)\hat{R}_n^m[t-1] \qquad \forall n \in \mathcal{N}_m, \tag{4.22}$$

where $\hat{R}_n^m[t-1]$ is the rate estimate in the previous TTI and $R_{p_w,w}[t]$ is the instantaneous packet rate. An exponential moving average $\hat{R}_n^m[t]$ as defined in equation (4.22) is used to estimate longer-term trends. Notice that this definition of $\hat{R}_n^m[t]$ is used to smooth the estimated rate, i.e., it is bias towards long term trends in order to avoid being affected by short term fluctuations. For simplicity, we suppress the explicit dependence on time t in the notation. Given a set of available paths $n \in \mathcal{N}_m$ and similarly to QBALAN' objective functions, we define the splitting ratio as:

$$\Psi_n^m = \frac{G_n^m - \hat{R}_n^m}{G_n^m}. \tag{4.23}$$

Notice that the end-to-end delay $d_{n,w}^m$ of flow w can be expressed as $d_{n,w}^m = f(\alpha_{p_w,w,n})$,

making explicit that the end-to-end delay depends on the assignment indicator $\alpha_{p_w,w,n}$, and that in order to reduce packet reordering the end-to-end delay difference should be as small as possible.

The objective is to find the value of the packet assignment indicator $\alpha_{p_w,w,n}$ that minimizes the splitting ratio Ψ_n^m and the end-to-end delay $d_{n,w}^m$ at the same time. Therefore the optimal packet assignment indicator $\alpha_{p_w,w,n}$ can be found by solving the following multi-objective optimization problem:

$$\min_{\alpha_{p_w,w,n}:w\in\mathcal{W}_m} \left\{ \max_{n\in\mathcal{N}_m} \left(\Psi_n^m, d_{n,w}^m\right) \right\} \forall m \in \mathcal{M}. \tag{4.24}$$

subject to:

$$\hat{R}_n^m \leq G_n^m \qquad \forall n \in \mathcal{N}_m \tag{4.25}$$

$$\sum_{n\in\mathcal{N}_m} \alpha_{p_w,w,n} = 1 \qquad \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m \tag{4.26}$$

$$\alpha_{p_w,w,n} \in \{0,1\} \qquad \forall p_w \in \mathcal{P}_w \forall w \in \mathcal{W}_m, \forall n \in \mathcal{N}_m. \tag{4.27}$$

Notice in the min-max equation (4.24) that $(\Psi_n^m, d_{n,w}^m)$ is a vector of objectives of the form $F(x) = (f_1(x), f_2(x))$. Constraint (4.25) ensures that packets are only assigned to a specific path if the corresponding rate $\hat{R}_n^m$ does not exceed the maximum rate G_n^m. Constraints (4.26) and (4.27) refer to the fact that a given packet can only be assigned to one path at a time.

In order for the set of equations (4.24) - (4.27) to have a feasible solution, there should exist at least one path n that satisfies equation (4.25). If there is no practical solution, packets of flow $w \in \mathcal{W}_m$ will be dropped. Radio node n sends G_n^m and $d_{n,w}^m$ information to all mobile devices m every δms.

Solving a multi-objective optimization problem is a challenging task. Nevertheless, there exist multiple methods to deal with multi-objective optimizations [93]. A classical approach is to formulate the multi-objective optimization problem as a single-objective optimization problem by means of scalarization such that the optimal solutions are Pareto optimal. A common difficulty with this method is to find the most appropriate parameters for the scalarization (different parameters means different optimal solutions).

Our proposed solution is related to the multi-level programming method. The multi-

level, and especially the bi-level optimization method is designed to deal with problems with two objectives in which the optimal decision of one of them is constrained by the decision of the second objective [94]. The second-level objective optimizes its solution under a feasible region that is defined by the first-level objective. In our problem the bi-level optimization method can be roughly modelled as in equations (4.28)-(4.29): minimize the end-to-end delay $d^m_{n,w}$ (second-level objective) subject to: maximize the splitting ratio Ψ^m_n (first-level objective) [95, 96].

$$\min \ d^m_{n,w} \tag{4.28}$$

$$\text{s.t. } \max \ \Psi^m_n \tag{4.29}$$

When solving this multi-objective optimization problem, we should remember that in general, specially in the context of real-time applications, the problem of finding the optimal solution to the load balancing problem is NP-hard [97, 98], therefore intractable when the number of packets exceeds a few units. This is the reason why we focus our attention to the use of heuristics in the next section.

4.5.2 DALBA Algorithm

In order to solve optimization problem (4.24) we propose the Delay Aware Load Balancing Algorithm DALBA implemented at the source mobile device. DALBA can be described in two main steps loosely corresponding to the bi-level optimization method:

- *First:* Assign the portion of traffic of flow $w \in \mathcal{W}_m$ to each path $n \in \mathcal{N}_m$ so as to minimize the splitting ratio Ψ^m_n difference among paths.
- *Second:* Assign packets of flow w to each path n so as to minimize the end-to-end delay $d^m_{n,w}$.

In order to appropriately choose the path to be used by every packet, we calculate the ratio λ^m_n as the product of the splitting ratio Ψ^m_n and the actual load G^m_n.

$$\lambda^m_n = \frac{\Psi^m_n \cdot G^m_n}{\sum_j (\Psi^m_j \cdot G^m_j)}. \tag{4.30}$$

Replacing equation (4.23) in equation (4.30), λ_n^m can be presented as:

$$\lambda_n^m = \frac{G_n^m - \hat{R}_n^m}{\sum_j (G_j^m - \hat{R}_j^m)}. \tag{4.31}$$

Ratio λ_n^m is the normalized difference between the maximum rate at which mobile device m can send traffic and its actual load. The intuition behind the ratio λ_n^m is based on the realization that in order to balance traffic load we should transmit traffic on each path proportionally to the available capacity $G_n^m - \hat{R}_n^m$.

The proposed algorithm DALBA runs in every mobile device and it works as follows: At every TTI, for every flow, it calculates the splitting ratio λ_n^m, then it evaluates if there exist enough available capacity among all the available paths to receive all the packets of the flow (if not, packets of that flow are discarded). if there is enough capacity it assigns packets to paths according to the splitting ratio (first main step).

Having the path ratio decided, DALBA decides which packets should be sent first (remember that the order in which packets are sent is important to avoid packet reordering). While there are packets to be sent DALBA selects the path that has the smallest end-to-end delay and sends the first packet in queue on that path (second main step). Notice that the end-to-end $d_{n^*,w}^m$ must be updated at every iteration inside the while loop by considering that a packet has being assigned for scheduling on path n^* (scheduling delay). The load balancing algorithm is described in Algorithm 7.

4.5.2.1 Complexity Analysis

It is worth noticing that flows are composed of packets. Therefore, packets are the smallest balancing units used by algorithm DALBA. The proposed algorithm allocates each packet after performing a linear search on the path that minimizes the end-to-end delay. Hence, the complexity to allocate the first packet is $\mathcal{O}(N)$, the complexity to allocate the second packet is $\mathcal{O}(N)$, and so on until all packets are allocated. Consequently, the total complexity of the algorithm is $\mathcal{O}(NP)$. i.e. the algorithm has linear complexity in the number of paths and in the number of packets, and thus could be easily implemented in real-time.

Algorithm 7 DALBA

Every TTI
$\alpha_{p_w,w,n} \leftarrow 0, \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m, \forall n \in \mathcal{N}_m$
for every flow $w \in \mathcal{W}_m$ **do**
  Calculate the ratio λ_n^m according to equation (4.31) $\forall n \in \mathcal{N}_m$
  if $\exists\ n$ such that $\hat{R}_n^m < G_n^m$ **then**
    $sPacket_n \leftarrow |\mathcal{P}_w| \times \lambda_n^m, \forall n \in \mathcal{N}_m$
    $counter_n \leftarrow 0, \forall n \in \mathcal{N}_m$
    while $\mathcal{P}_w \neq \emptyset$ **do**
      Find path $n^* = \arg\min_n \{d_{n,w}^m\}$
      if $counter_{n^*} < sPacket_{n^*}$ **then**
        $\alpha_{p_w,w,n^*} \leftarrow 1$ (assign packet $p_w \in \mathcal{P}_w$ to radio node n^*)
        $counter_{n^*} \leftarrow counter_{n^*} + 1$
        $d_{n^*,w}^m \leftarrow$ Update (scheduling delay)
        $\mathcal{P}_w \leftarrow \mathcal{P}_w \setminus \{p_w\}$
      else
        $d_{n^*,w}^m \leftarrow \infty$
      end if
    end while
  else
    Discard packets of flow w
  end if
end for

4.5.3 Simulation Settings

In this section, we present the simulation settings used to assess the performance of our load balancing algorithm DALBA and then we analyze the experimental results.

4.5.3.1 Evaluation Environment

We evaluate DALBA using trace-driven simulations in MATLAB. Our data set contains videos comprised of action, comedy and drama movie trailers. The architecture for simulations is depicted in Fig. 4.9.

4.5.3.2 Simulation Setup

We consider an LTE wireless mobile cellular system with 30 radio nodes and 100 mobile devices in which each mobile device is simultaneously connected to multiple radio nodes.

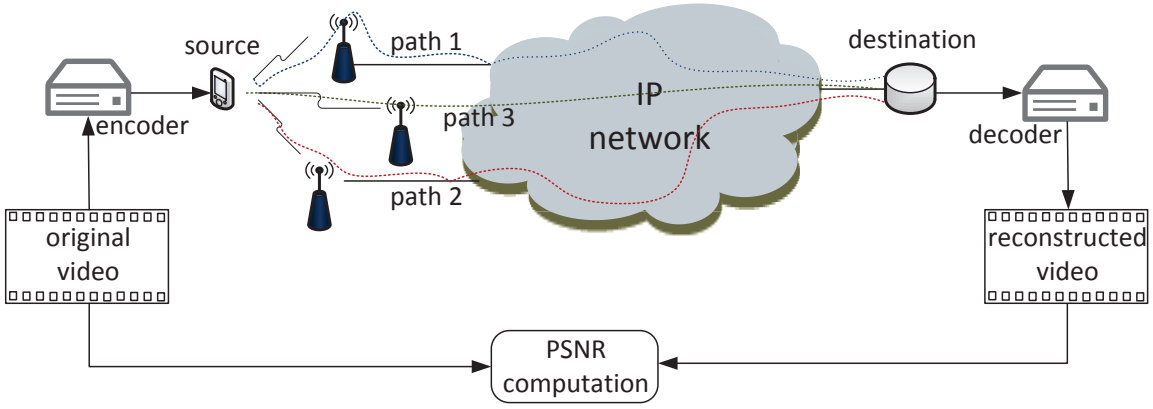


Fig. 4.9 Simulation setup: basic architecture for simulations.

We have two scenarios. In scenario 1, all mobile device's are connected to 3 radio nodes, and in scenario 2, 50% of the mobile device's are connected to 2 radio nodes and 50% are connected to 3 radio nodes. We show the general simulation network topology in Fig. 4.10.

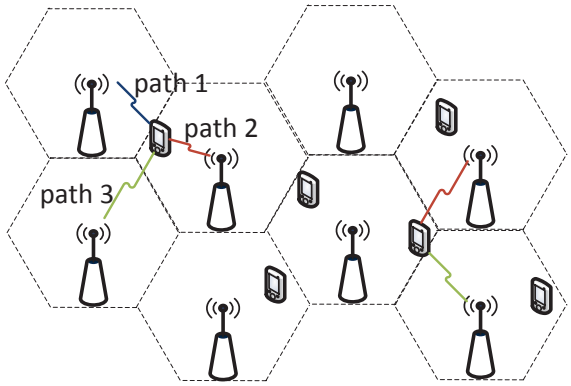


Fig. 4.10 Simulation network topology: 30 radio nodes, 100 mobile devices uniformly distributed.

We assume that all radio nodes belong to the same network operator, and that radio nodes have a perfect knowledge of the maximum rates (channel capacity) a mobile device can transmit on each path. These information is provided to each mobile device by the radio nodes at least every δms. We also assume that mobile device's are able to estimate the end-to-end delay at least every δms.

Our experiments assume on demand streaming H.264 videos. Ffmpeg software is adopted as the video codec tool [99]. The movie trailers are encoded with variable bit rate (VBR) using mp4 format at 24 fps (frames per second). To ensure diversity on the

movie trailers used we uniformly choose them among 3 video categories (action, comedy and drama). A GOP (group of pictures) consists of 12 frames (IBBPBBPBBPBB). Input traffic on each mobile device is composed of four video streams. The video frames are transmitted to the destination using UDP packets. We further assume that mobile devices are static and that the buffer size at the destination is large enough to accommodate all the arriving packets. Hence, packet loss could only be caused by late packets.

We run our simulations for 100s as it is the minimum time a video trailer usually last. Although not shown here DALBA is not very sensitive to small changes on the value of the exponential moving average parameter γ, Nevertheless, quickly reacting to the video throughput fluctuations (high values of γ) causes the end-to-end delay to increase, which affects negatively the overall PSNR measurement.

In our simulations we use the concepts of decoding threshold time and traffic load. The decoding threshold time is defined as the maximum allowable time a packet can remain in the buffer after the play-out deadline. We use a decoding threshold time of 500ms for a video on demand service that buffers 500ms and then plays out the video. Traffic load is the total traffic carried by the network and it is represented as a percentage of the total network capacity.

In order to model the total multi-hop end-to-end delay of any given path between a mobile device and the destination we use the Pareto distribution. The Pareto distribution offers a close approximation to the real end-to-end delay with low complexity [92, 100]. Table 4.1 summarizes the simulation parameters.

4.5.3.3 Compared Load Balancing Schemes

We compare DALBA with three existing load balancing models, which are summarized as follows:

- **Flowlet Aware Routing Engine (FLARE)** [50] (see Section 2.4.2.1) groups packets into flowlets and uses these flowlets as the balancing unit. The round trip delay is checked every 500ms.

- **Sub-Packet based Multipath Load Distribution (SPMLD)** [57] (see Section 2.4.2.2) minimizes the end-to-end delay solving a constrained optimization problem. The predominant parameters C_1, C_2 and C_3 are set to 1.2, -0.17 and -0.005 respectively.

Table 4.1 DALBA Simulation parameters

Parameter	**value**
Simulation time	100s
Radio nodes	30
Mobile devices	100
Mobile device location	uniform distribution
Mobile device speed	static
Paths per mobile device (scenario 1)	3
Paths per mobile device (scenario 2)	2 (50%), 3 (50%)
Flows per mobile device	4
Traffic type	UDP
Delay difference between paths	5ms
Delay variance	$1ms^2$
Average video rate per flow	1.6Mbps
Frames per second	24
Group of pictures (GOP)	12 frames (IBBP..BBP)
Video category	action, comedy, drama
Updating time δ	500ms
Exponential moving average parameter γ	0.001

- **Load balancing algorithm (QBALAN)**(see Section 4.4) subdivides flows into two groups, short-term and long-term, and uses these groups to minimize the splitting error and packet reordering. The round trip delay is checked every 500ms for the short-term load balancing algorithm, and every 7s for the long-term load balancing algorithm.

4.5.4 Simulation Results

4.5.4.1 Splitting Error

First, we show the percentage of splitting error as a function of the traffic load for scenarios 1 and 2 in Fig. 4.11. The results indicate that DALBA accurately splits traffic, managing to achieve the lowest splitting error in both scenarios. Contrary to SPMLD, as the traffic load increases DALBA manages to maintain the splitting error almost constant. This is because DALBA quickly adjusts the splitting ratio such as to sent packets to the paths with smaller load.

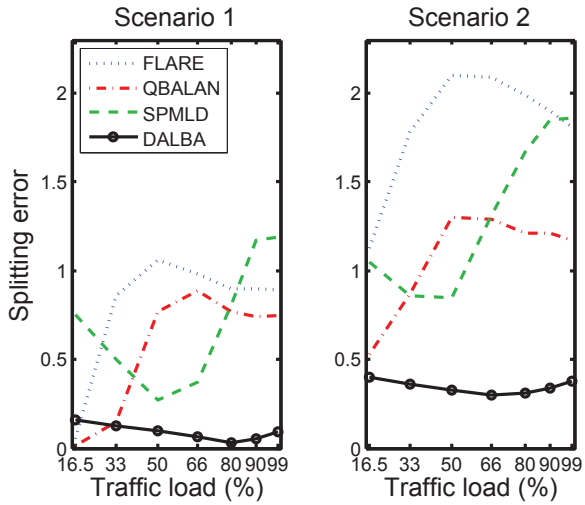


Fig. 4.11 Average splitting error (SE) versus Traffic load for scenarios 1 and 2.

4.5.4.2 Packet Reordering

Fig. 4.12 shows the reordering density at 80% of the traffic load for scenario 1. Here DALBA has lower packet reordering than SPMLD. DALBA keeps the packet reordering below five packets which is the price to pay if we want to reduce packet loss. Contrary to DALBA, QBALAN and FLARE have lower packet reordering at the cost of higher packet loss as we can observe in Fig. 4.15.

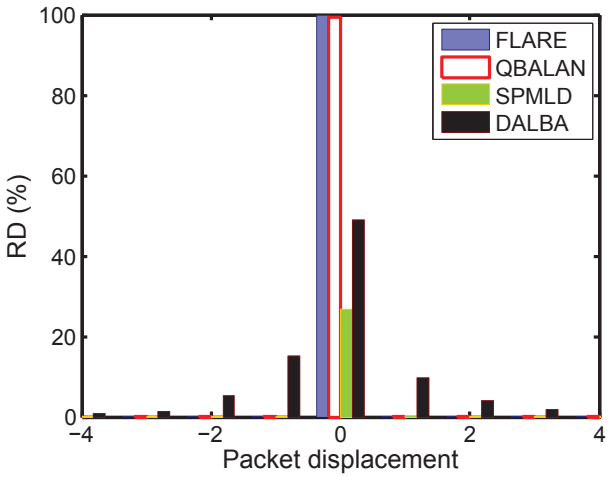


Fig. 4.12 Reorder density at 80% traffic load for scenario 1.

In order to have a sense of the level of dispersion of packet reordering we use the reordering entropy. Fig. 4.13 summarizes the reorder entropy as a function of the traffic load for both scenarios. According to Fig. 4.13 DALBA has lower reordering than SPMLD but higher than QBALAN and FLARE. This could be explained by the fact that DALBA tries to distribute packets of the same flow over all the available paths, which might cause packet reordering. Nevertheless, we also have to analyse how deep the reordering is, in order to assess the real packet reordering impact.

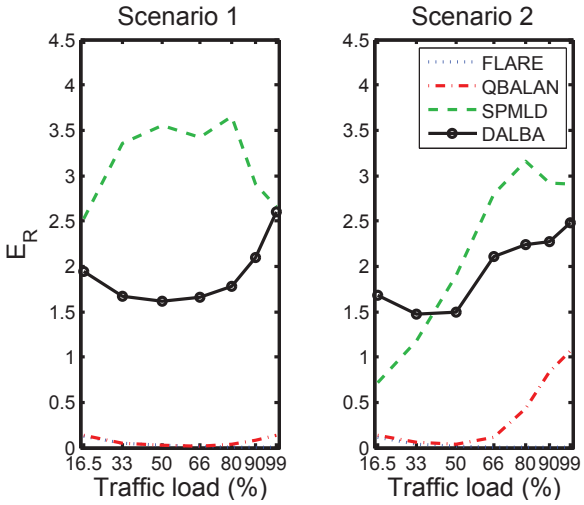


Fig. 4.13 Reorder entropy (E_R) versus Traffic load for scenarios 1 and 2.

Another metric that allows us to analyze the packet reordering behaviour is the mean displacement of packets shown in Fig. 4.14 for scenario 1. In this figure DALBA, QBALAN and FLARE have the lowest levels, DALBA keeps the mean displacement of packets below 5 packets even at high traffic loads. We should remember that these metrics capture reordering at the packet level and are not directly associated with the decoding threshold time (maximum allowable time a packet can remain in the buffer after the play-out deadline). Therefore, in order to determine the acceptable level of reordering we must also verify its impact at the application level.

Fig. 4.15 shows that DALBA has the smallest packet loss especially when the traffic load increases. This is due to the fact that DALBA's strategy is to deliver packets as fast as it can, therefore meeting the decoding threshold time.

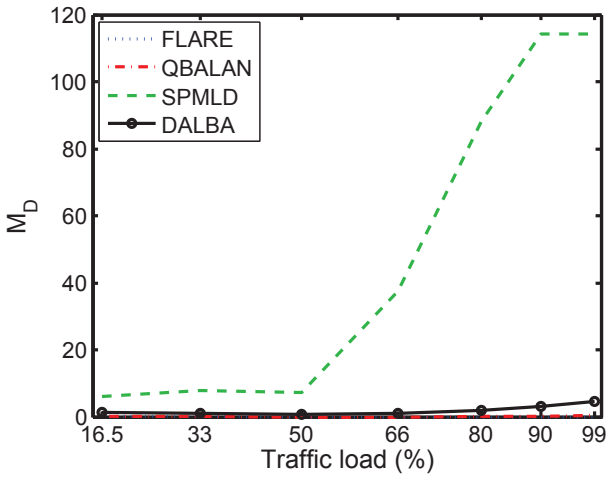


Fig. 4.14 Mean displacement of packets versus Traffic load for scenario 1.

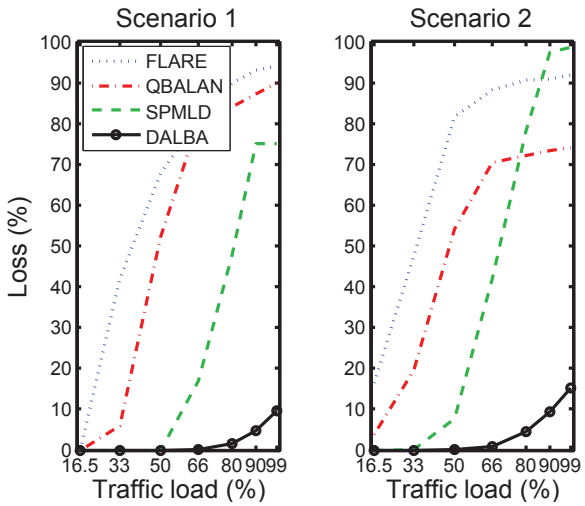


Fig. 4.15 Packet loss versus Traffic load, non real-time (decoding threshold time 500ms) for scenarios 1 and 2.

4.5.4.3 End-to-end Delay

Fig. 4.16 shows that DALBA has superior end-to-end delay performance as compared to the other algorithms. This can be attributed to the ability of DALBA to send packets to the paths that has smaller delay. QBALAN and FLARE manage to have smaller packet reordering due to its strategy of sending packets of the same flow by the same path, which in turn increases the end-to-end delay. SPMLD makes an effort to keep low delay but it struggles when the traffic load increases.

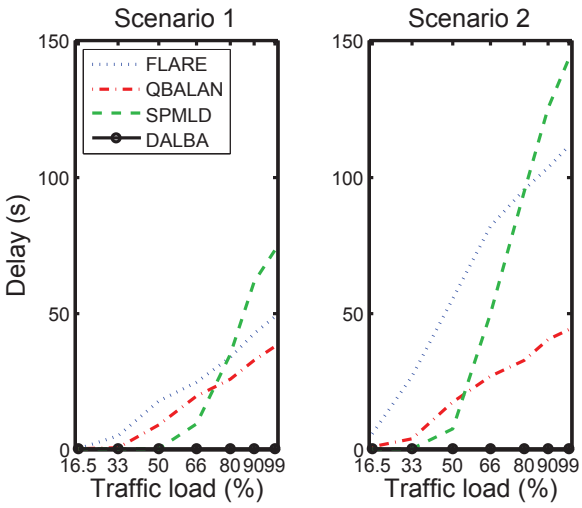


Fig. 4.16 End-to-end delay versus Traffic load for scenarios 1 and 2.

Fig. 4.17 shows the end-to-end delay for scenario 1 for the packets that meet the decoding threshold time of 500ms (packets that arrive late are discarded). In this figure we can see that DALBA achieves smaller end-to-end delay. When the traffic load increases SPMLD has smaller end-to-end delay at the cost of higher packet loss, as can be seen in Fig. 4.15. FLARE and QBALAN reduce packet reordering by assigning packets to the same path but this strategy generates congestion and therefore higher end-to-end delay and packet loss.

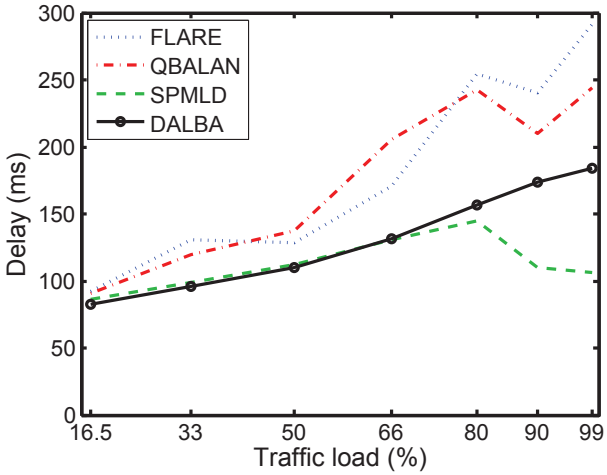


Fig. 4.17 End-to-end delay versus Traffic load, non real-time (decoding threshold time 500ms) for scenario 1.

4.5.4.4 PSNR

Fig. 4.18 shows the average PSNR values versus different traffic loads for scenario 1. It can be noticed that DALBA has better PSNR performance. We can also notice that the PSNR pattern is almost the opposite to that of packet loss (see Fig. 4.15). Larger end-to-end delay leads to more packet loss and that degrades the quality of the video. It is worth mentioning that DALBA's good PSNR performance holds true for larger values of delay variance and delay differences between paths.

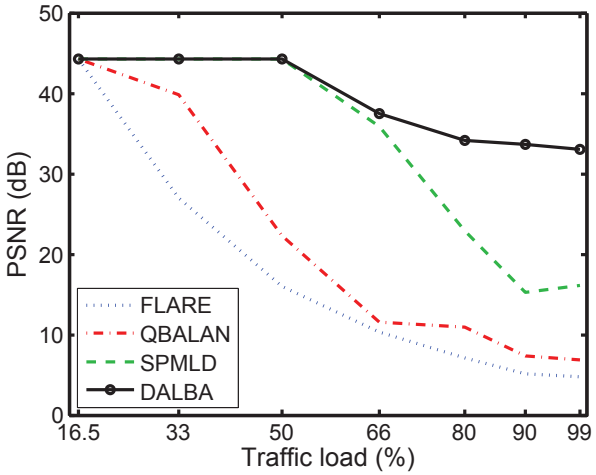


Fig. 4.18 PSNR versus Traffic load, non real-time (decoding threshold time 500ms) for scenario 1.

DALBA outperforms the other algorithms by delivering more video frames within the decoding threshold time. We can see this especially when the traffic load increases forcing the system to efficiently use all the available paths. We also plot the PSNR distribution for scenario 1 for the case of a decoding threshold time of 500ms at 80% traffic load in Fig. 4.19 and the PSNR standard deviation as a function of the traffic load in Fig. 4.20. In both figures we can observe that DALBA achieves higher PSNR values (Fig. 4.19) with smaller variations (Fig. 4.20) while the other compared algorithms struggle to maintain its performance at high traffic loads.

4.5.5 Discussion

In this section, we have proposed DALBA a multipath load balancing method for heterogeneous wireless uplink systems that deals with the problem of how to distribute traffic

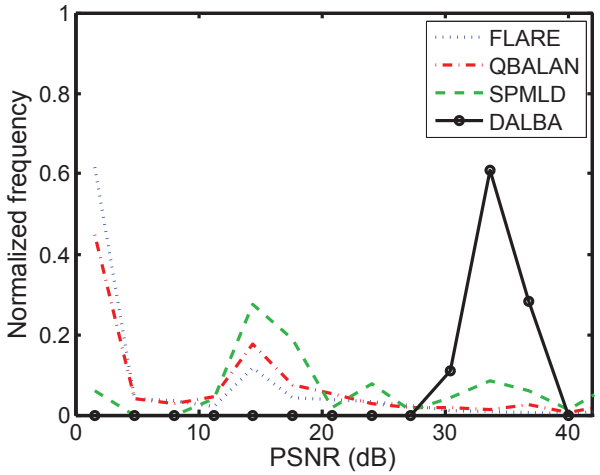


Fig. 4.19 PSNR distribution at 80% traffic load (decoding threshold time 500ms) for scenario 1.

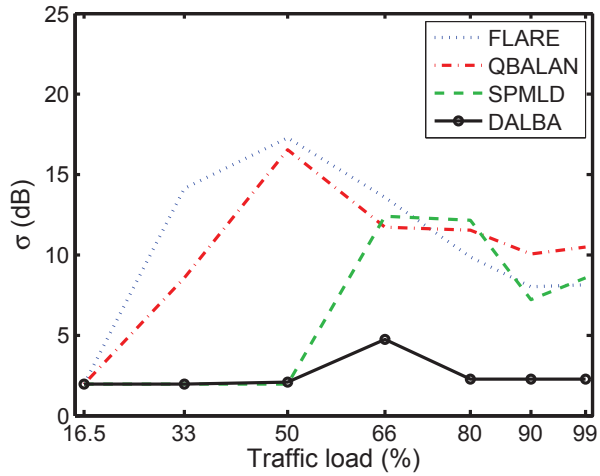


Fig. 4.20 PSNR standard deviation versus Traffic load (decoding threshold time 500ms) for scenario 1.

without causing excessive packet reordering, while keeping the splitting error and the end-to-end delay as small as possible. DALBA is a sub-optimal heuristic solution specifically tested in the context of UDP multimedia applications.

In order to analyze DALBA's performance, we conduct extensive simulations in MATLAB using H.264 video streaming. Simulation results demonstrate that: (1) DALBA accurately splits traffic while reducing the average packet loss and the average end-to-end delay; (2) DALBA is robust to different traffic loads. It exhibits superior PSNR perfor-

mance in high traffic load conditions; (3) DALBA outperforms previous algorithms in total end-to-end delay and reduces packet reordering.

4.6 Energy Efficient Load Balancing

In this section, we expand the way we handle the load balancing problem by incorporating the power consumption. This is a very important aspect to consider especially in wireless networks where the main concern is not to run out of battery too quickly, and that is not generally taken into account in the literature. We examine three important issues: First, how to dynamically distribute traffic among each network interface such that the load is balanced. Second, how to minimize the mobile device's power consumption. Third, how to reduce packet reordering without increasing the end-to-end delay.

4.6.1 Problem Statement

Here, we address the problem of load distribution over multiple network interfaces; our analysis is restricted to the uplink scenario. We introduce the optimization problem for solving the mobile device's power consumption and average delay minimization with splitting error constraint. The basic network scenario is shown in Fig 4.21.

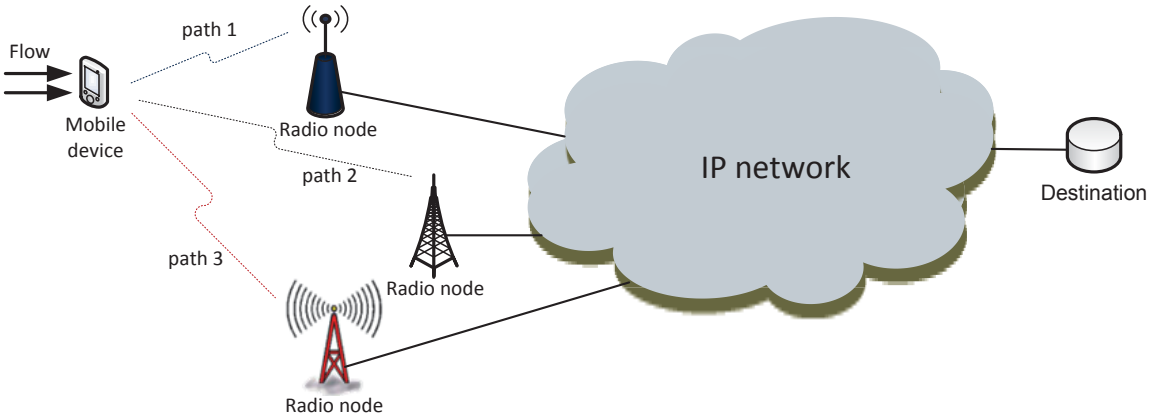


Fig. 4.21 5G Basic scenario, mobile device connected to multiple radio nodes.

In a given TTI, any flow $w \in \mathcal{W}$ can be scheduled through multiple radio nodes $n \in \mathcal{N}$. The main objective is to find the most convenient packet assignment such that we sent

traffic to radio nodes minimizing the mobile device's power consumption and the end-to-end delay while limiting the splitting error and the negative effects of packet reordering.

Similarly to the last section where we developed the DALBA algorithm, let us define $\alpha_{p_w,w,n} \in \{0,1\}$ as the packet assignment indicator, $\alpha_{p_w,w,n} = 1$ if packet $p_w \in \mathcal{P}_w$ is assigned to radio node n, let us also introduce here ξ_n^m as the average power consumption of mobile device m when sending traffic to radio node n. $\hat{R}_n^m$ is updated on each mobile device as described in equation (4.22). Different from DALBA (equation (4.23)), we present here a more accurate definition of the splitting ratio, which explicitly takes into account the desired load K_n^m, instead of the channel capacity G_n^m. Thus, the splitting ratio can be defined as:

$$\Psi_n^m = \frac{|K_n^m - \hat{R}_n^m|}{K_n^m}. \tag{4.32}$$

Notice also here that the end-to-end delay $d_{n,w}^m$ of flow w can be expressed as $d_{n,w}^m = f(\alpha_{p_w,w,n})$, showing that the end-to-end delay $d_{n,w}^m$ depends on the assignment indicator $\alpha_{p_w,w,n}$, and that to reduce packet reordering the end-to-end delay difference should be as small as possible. $d_{n,w}^m$ is affected by $\alpha_{p_w,w,n}$ because more packets on a path may cause higher queuing and processing delay. During the optimization window depending to the path chosen, we set the end-to-end delay of that packet equal to the end-to-end delay associated with the path.

The objective is to find the value of the packet assignment indicator $\alpha_{p_w,w,n}$ that minimizes the total average mobile device's power consumption $\sum_n \xi_n^m$ and the end-to-end delay $d_{n,w}^m$ for each mobile. Then, the problem can be formulated as a constrained multi-objective optimization problem as follows:

$$\min_{\alpha_{p_w,w,n}:w\in\mathcal{W}_m} \left\{ \max_{n\in\mathcal{N}_m} \left(\sum_n \xi_n^m, d_{n,w}^m \right) \right\} \forall m \in \mathcal{M} \tag{4.33}$$

subject to:

$$\hat{R}_n^m \leq G_n^m \quad \forall n \in \mathcal{N}_m \tag{4.34}$$

$$\Psi_n^m \leq \varepsilon \quad \forall n \in \mathcal{N}_m \tag{4.35}$$

$$\sum_{n \in \mathcal{N}_m} \alpha_{p_w,w,n} = 1 \quad \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m \tag{4.36}$$

$$\alpha_{p_w,w,n} \in \{0,1\} \quad \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m, \forall n \in \mathcal{N}_m, \tag{4.37}$$

where constraint (4.34) guarantees that packets are only assigned to a given path if the rate estimate $\hat{R}_n^m$ does not exceed the maximum rate G_n^m. Notice that the idea is to transmit at the desired load $K_n^m \leq G_n^m$. Constraint (4.35) ensures that the splitting error does not exceed threshold ε. Constraints (4.36) and (4.37) state that a specific packet cannot be assigned to more than one path at a time. For the set of equations (4.33)-(4.37) to have a feasible solution, it is necessary that at least one path n satisfies constraint (4.34). If it is not feasible to find a practical solution, packets of flow $w \in \mathcal{W}_m$ will be discarded. Radio node n transmits G_n^m, K_n^m and $d_{n,w}^m$ information to each mobile device m every δms.

There are many methods to solve multi-objective optimizations. Instead of using the classic approach of reducing the multi-objective to a single objective by trying to find a relationship between power and delay, our proposed solution is related to the bi-level optimization method. The bi-level method deals with problems in which the optimal decision of one of the objectives is constrained by the decision of the other objective. The second-level objective optimizes its solution under a feasible region that is determined by the first-level objective. We must take into consideration that although solving this multi-objective optimization problem might be theoretically possible, finding all the feasible solutions is generally impractical specially in the case of real-time applications. For that reason we focus our attention to the use of heuristics in the next section.

4.6.2 GEL Algorithm

In order to deal with the multi-objective optimization problem (4.33)-(4.37) we propose the load balancing algorithm GEL[3] implemented in each mobile device. GEL relaxes the

[3]Oscar Delgado and Fabrice Labeau, "Uplink energy-efficient load balancing over multipath wireless networks", IEEE Wireless Communication Letters (WCL), pp. 424-427, August 2016

splitting ratio constraint (4.35) allowing violations to the threshold ε. GEL can be described in two main steps:

1. Determine the portion of traffic of flow $w \in \mathcal{W}_m$ to be assigned to each path $n \in \mathcal{N}_m$ so as to minimize the average power consumption ξ_n^m such that the splitting ratio Ψ_n^m does not exceed the threshold ε. We should notice that GEL does not strictly guarantee that the splitting ratio will be below the threshold ε.

2. Assign packets of flow w to each path n such that it minimizes the end-to-end delay $d_{n,w}^m$ difference. This step reduces the possibility to experience packet reordering.

To choose the best path to be used by each packet, we determine the ratio λ_n^m as:

$$\lambda_n^m = \frac{G_n^m - \hat{R}_n^m}{\tilde{\xi}_n^m \sum_j \frac{G_j^m - \hat{R}_j^m}{\tilde{\xi}_j^m}}, \tag{4.38}$$

where $\tilde{\xi}_n^m = \frac{\xi_n^m}{\hat{R}_n^m}$ is the average power consumption per bit. λ_n^m is related to the available capacity $G_n^m - \hat{R}_n^m$ and the inverse of the power consumption per bit $\tilde{\xi}_n^m$. It regulates the way we balance the power consumption among paths. The load balancing algorithm GEL is described in Algorithm 8. Notice that the ratio λ_n^m defined here is different from the one defined in the case of the DALBA algorithm (equation (4.31)) in which the power consumption ξ_n^m was not taken into account.

4.6.2.1 Complexity Analysis

The convergence time of the proposed algorithm GEL can be analysed as follows. GEL allocates each packet after carrying out a linear search on the path that minimizes the end-to-end delay. The complexity to allocate the first packet is $\mathcal{O}(N)$, the second packet is also $\mathcal{O}(N)$, and so on, until all packets are allocated. Consequently, GEL's computational complexity is on the order of $\mathcal{O}(NP)$, i.e. GEL has linear complexity in the number of paths and in the number of packets. Our proposed heuristic algorithm can be quickly implemented to find a pseudo-optimal real-time solution with low complexity.

Algorithm 8 Green Energy-efficient Load Balancing (GEL)

Every TTI $\alpha_{p_w,w,n} \leftarrow 0, \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m, \forall n \in \mathcal{N}_m$
Calculate the splitting error Ψ_n^m according to equation (4.32) $\forall n \in \mathcal{N}_m$
for every flow $w \in \mathcal{W}_m$ **do**
  if $\exists\ n$ such that $\hat{R}_n^m < G_n^m$ **then**
    Calculate the ratio λ_n^m according to equation (4.38) $\forall n \in \mathcal{N}_m$
    if $\Psi_n^m > \varepsilon$ **then**
      $\lambda_n^m = \frac{G_n^m - \hat{R}_n^m}{\sum_j (G_j^m - \hat{R}_j^m)}$
    end if
    $sPacket_n \leftarrow |\mathcal{P}_w| \times \lambda_n^m, \forall n \in \mathcal{N}_m$
    $counter_n \leftarrow 0, \forall n \in \mathcal{N}_m$
    while $\mathcal{P}_w \neq \emptyset$ **do**
      Find path $n^* = \arg\min_n \{d_{n,w}^m\}$
      if $counter_{n^*} < sPacket_{n^*}$ **then**
        $\alpha_{p_w,w,n^*} \leftarrow 1$
        $counter_{n^*} \leftarrow counter_{n^*} + 1$
        $\mathcal{P}_w \leftarrow \mathcal{P}_w \setminus \{p_w\}$
      else
        $d_{n^*,w}^m \leftarrow \infty$
      end if
    end while
  else
    Discard all packets of flow w
  end if
end for

4.6.3 Numerical Results

4.6.3.1 Simulation Settings

We evaluate the performance of GEL using trace-driven simulations in MATLAB. Our experiments assume on demand streaming H.264 videos. The simulation architecture is depicted in Fig. 4.21.

We assume that one network operator handles all radio nodes, and that the radio nodes have a perfect knowledge of the desired rate and the channel capacity of each mobile device. This information is transmitted to every mobile device by the radio nodes at least every δms. We also assume that mobile devices are static and that the buffer size at the destination is large enough to accommodate all the arriving packets. Thus, packet loss

could only be caused by late packets. In our simulations we represent the percentage of the total network capacity as the traffic load. We model the multi-hop end-to-end delay using a Pareto distribution. The Pareto distribution offers a close approximation to the real end-to-end delay with low complexity. Table 4.2 summarizes the simulation parameters.

Table 4.2 GEL Simulation parameters

Parameter	**value**
Simulation time	100s
Radio nodes	30
Mobile devices	100
Mobile device location	Poison point process
Mobile device speed	static
Paths per mobile device	3
Flows per mobile device	4
Traffic type	UDP
Delay difference between paths	5ms
Delay variance	$1ms^2$
Video rate per flow	1.6Mbps
Frames per second	24
Group of pictures (GOP)	12 frames (IBBP...BBP)
Video category	action, comedy, drama
Updating time δ	500ms
Exponential moving average parameter γ	0.001
Path-loss exponent α_1, β	2, 4
Mobile device maximum transmission power ξ_{max}	200mW
Power control parameter ξ_0	0.8μW
Cell radius r	500m
Path-loss breakpoint g	400m

We model the mobile device power consumption per bit $\tilde{\xi}_n^m$ assuming a slow power control mechanism as defined in [101]. This model uses a dual-slope propagation path loss model to describe a typical wireless network characterized by distance-dependent path loss, shadowing and fading.

$$\tilde{\xi}_n^m = \min\left(\xi_{max}, \xi_0 \frac{r^{\alpha_1}(1+r/g)^{\beta}}{L_K}\right), \tag{4.39}$$

where ξ_{max} is the maximum power a mobile device can transmit, ξ_0 is the cell specific parameter used to control the target SINR, r is the distance between mobile device m and

radio node n, g is the breakpoint of a path loss curve, α_1 is the basic path-loss exponent before breakpoint, α_2 is the additional path-loss exponent, $\beta = \alpha_2 - \alpha_1$ and L_K is the path loss constant.

4.6.3.2 Simulation Results

First in Fig. 4.22 we present the normalized power consumption as a function of the traffic load in the system. We normalize the power consumption as the actual power consumption per maximum power consumption. The power consumption experienced by GEL is observed to be smaller than all the other algorithms. Here GEL can get up to 12% power consumption reduction with respect to its closest competitor DALBA (at 66% traffic load GEL consumes 3.22W, DALBA 3.67W and SPMLD 4.66W). We can also observe that with high traffic loads SPMLD does not manage to keep low power consumption levels. This is because when trying to minimize packet reordering it pushes mobile devices to transmit on paths with higher power consumption.

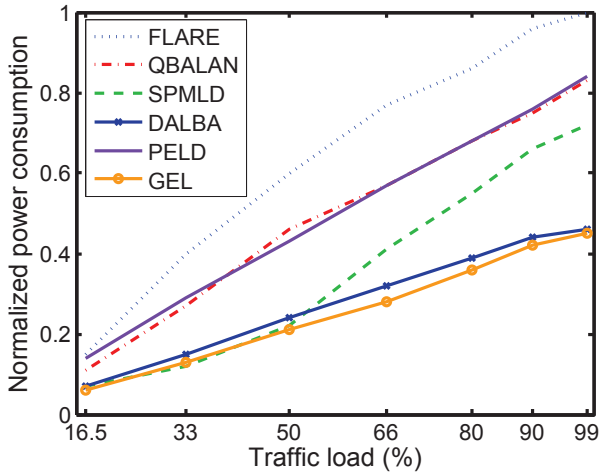


Fig. 4.22 Normalized power consumption versus Traffic load.

Fig. 4.23 shows the average PSNR for different traffic loads. We can observe in this figure that GEL manages to achieve similar performance as DALBA while reducing the power consumption. GEL PSNR performance is due to its ability to intelligently adjust the splitting ratio, such as to send packets through the paths with smaller power cost.

In order to analyze the impact of the splitting ratio we show in Fig. 4.24 the overall splitting error versus the traffic load. The results reveal that GEL slightly loses its splitting

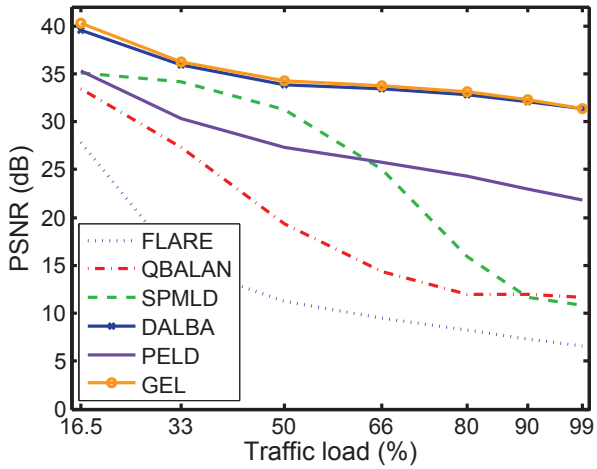


Fig. 4.23 PSNR (real-time decoding) versus Traffic load.

accuracy due to the fact that packets are sent to the paths with smaller power consumption instead of the paths that will reduce the splitting error. Notice that even though there is a trade-off between splitting error and power consumption, this increased splitting error is still smaller than most of the other algorithms.

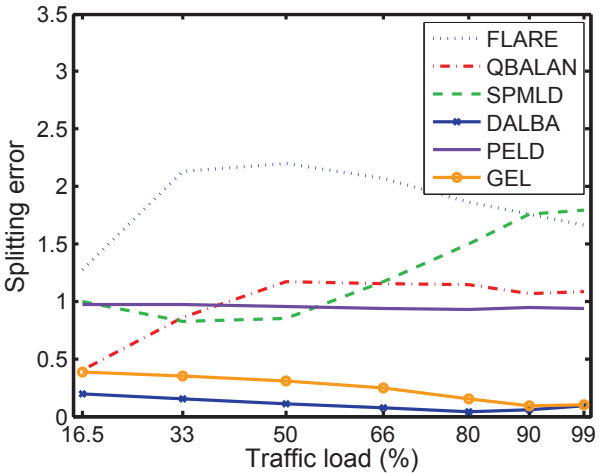


Fig. 4.24 Average splitting error SE versus Traffic load.

Other important metrics to consider are the end-to-end delay shown in Fig. 4.25 and the packet loss shown in Fig. 4.26. Here we can observe that although the average end-to-end delay for the packets that meet the real-time decoding threshold is not the smallest one, it

actually presents a smaller packet loss percentage, which explains why the perceived PSNR has higher values even at close to saturation traffic loads. FLARE and SPMLD do not manage to schedule packets on time causing higher packet losses which explains their poor PSNR performance at high traffic loads.

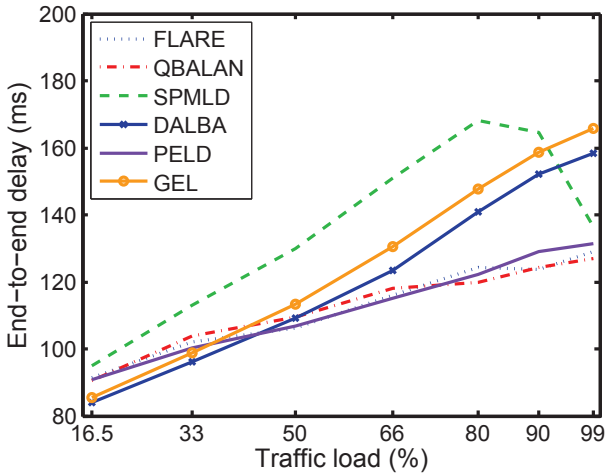


Fig. 4.25 End-to-end delay versus Traffic load.

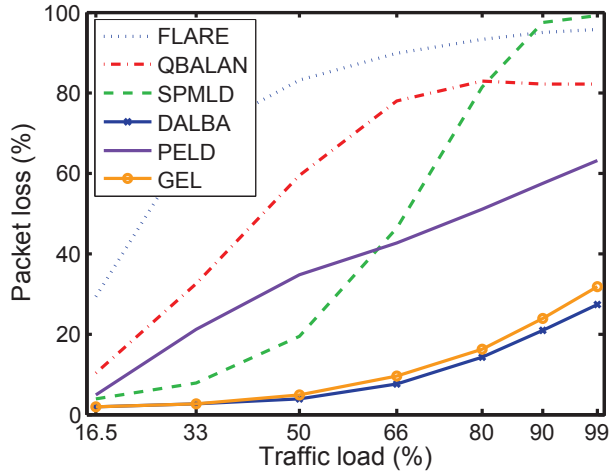


Fig. 4.26 Packet loss versus Traffic load, real-time decoding.

4.6.4 Discussion

In this section, we have dealt with the problem of how to split traffic in wireless uplink mobile systems such that the mobile device's power consumption and end-to-end delay is minimized. First we have proposed a multi-objective optimization problem, and then we have proposed a sub-optimal heuristic solution GEL that achieves the desired objective by relaxing the splitting error constraint.

Our simulation results demonstrate: First, that our proposed algorithm can reduce mobile device's power consumption while satisfying the basic QoS requirements. Second, GEL demonstrates that there is a trade-off between power consumption and splitting error. Third, GEL outperforms other solutions by about 10% while reducing the negative effects of packet reordering and long end-to-end delays across different traffic loads.

4.7 Summary

In this chapter, we have developed three multipath load balancing algorithms that deal with the problem of how to split traffic such that the traffic load remains at the desired load. While the load balancing problem has received significant attention in recent years, to the best of our knowledge, this is one of the few works that considers a scenario where mobile devices with multiple network interfaces are connected simultaneously to multiple radio nodes.

The performance evaluation focuses on uplink wireless networks transmitting UDP video traffic. To evaluate our heuristics, we have implemented our solution on an LTE network using MATLAB simulations. Our results show significant improvements not only in splitting error, but also in PSNR. Additionally, our proposed algorithms do not require any complex modifications on the radio node, nevertheless they require that mobile devices have access to channel capacity information and end-to-end delay measurements. In summary, our contributions are:

1. QBALAN shows that highly accurate load balancing can be achieved by subdividing the load balancing task into long-term and short-term load balancing algorithms without adding complexity. This strategy effectively reduces the splitting error by reducing the average number of times a flow switch paths, while not significantly affecting the packet reordering.

2. DALBA accurately splits traffic while reducing the average packet loss and the average end-to-end delay. It is robust to different traffic loads and it exhibits superior PSNR performance in high traffic load conditions. Additionally, DALBA outperforms previous algorithms by reducing the total end-to-end delay and packet reordering.

3. GEL manages to reduce the mobile devices' power consumption while maintaining acceptable QoS levels. It also demonstrates that there is a trade-off between power consumption and splitting error that should be considered when evaluating a load balancing strategy.

Another important question to consider, when working with wireless networks, is how to increase the bandwidth capacity such that, we can use this capacity to improve the communication quality, e.g., high definition videos require higher data rates. In the next chapter, we address this question by considering a device-to-device communication approach.

Chapter 5

Device-to-device Communication in Cellular Networks

5.1 Introduction

Wireless networks are without a doubt becoming the most important means of having access to network connectivity. At the same time the growing demand for high quality multimedia communications requires new technologies to be capable of providing the expected high network capacity [102]. The new generation of mobile networks (5G) aims to provide not only higher data rates, enhanced coverage area, lower power consumption, etc., but also fairness in user throughput [103].

Power consumption efficiency is an important requirement of 5G technologies that can help to reduce the total network operating cost, enable to extend connectivity to remote areas, and also help to provide a more sustainable and efficient way of using network resources [104]. Another important aspect to consider is fairness. Although there is no consensus on a definition of fairness, in our context, we consider that being fair means to provide equal throughput to all users. It is easy to notice that in a wireless cellular system users close to the center of the cell will naturally have higher throughput as compared to the users at the edge of the cell [105].

One new feature that we consider in 5G networks is that new mobile devices could have more than one network interface of the same or different technology that can be enabled to transmit or receive traffic simultaneously. Therefore, we can use this new feature to

enable mobile devices to act as full duplex relays. The concept of device-to-device (D2D) communications is not new in the literature. Nevertheless, we analyze the possible benefits of using mobile devices with multiple network interfaces of the same technology that can be enabled by the network to establish simultaneous connections to multiple destinations.

We have to be aware that opening the possibility of D2D communications creates new challenges such as how to handle the inter-tier interference or how to get channel information for the new possible connections. Some of the ways to manage the D2D-to-cellular interference consist of adopting a power control mechanism or properly allocating channel resources to D2D transmitters [106]. To establish any communication, mobile devices must get channel information from their neighbors. One possible way of getting this information is by utilizing the sounding reference signal (SRS) channel in LTE, which has been shown to be feasible in a D2D context with manageable complexity [107, 108]. Furthermore, the signaling overhead is expected to increase due to the exchange of necessary information between nodes. Therefore, smart ways to minimize the signaling overhead must be implemented [109]. Another aspect to consider, that is important for the success of D2D communications and that is not the focus of this thesis, is security. D2D could be more vulnerable to security issues due to the limited computational capacity of mobile devices, the relay transmission structure, etc.. A review on security can be found in [80].

In this chapter, we propose an infrastructure based solution (controlled by the network operator) that enables relaying capabilities on mobile devices with multiple network interfaces. In this context, we describe a heuristic solution that aims to maximize the network capacity.

5.2 Problem Statement

We consider a single cell of a wireless network that has mobile devices with two network interfaces which can be used simultaneously. The network allows mobile devices to establish direct communication between each other, making possible for mobile devices to act as full duplex relays. We limit any transmission to using only one relay (two hops).

To simplify our analysis, we assume that every mobile device tries to communicate with a destination outside of the cell, and that an efficient spectrum allocation scheme is being used [110]. We further assume that, due to this new feature, the control signal information (signaling overhead) used to establish communications might at most double.

Fig. 5.1 shows a basic scenario.

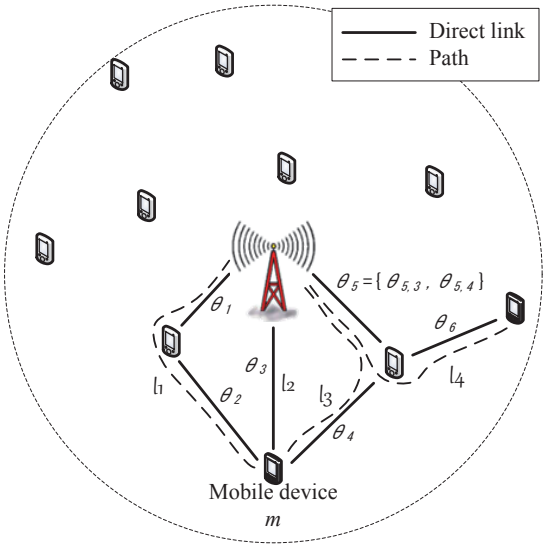


Fig. 5.1 Basic network diagram showing paths l_j and direct links $\theta_{k,j}$ for mobile device m in a single cell scenario.

In this scenario, we use two types of connections: 1) The concept of physical direct link refers to the one-hop wireless connection from a mobile device to a radio node or from a mobile device to another mobile device. 2) We also use the notion of path as the potentially two-hop connection between a mobile device and a radio node. A path is composed of one or more physical direct links, which make possible that several paths use the same physical direct link.

At a given point in time, we define $\mathcal{M}$ as the set of mobile devices, $\mathcal{M}^a$ as the set of active mobile devices (establishing a call) and $\mathcal{M}^i$ as the set of idle mobile devices (waiting for a call) such that $\mathcal{M} = \mathcal{M}^a \cup \mathcal{M}^i$ and $\mathcal{M}^a \cap \mathcal{M}^i = \emptyset$. K is the set of physical direct link indexes with elements $k \in K$ and J is the set of path indexes with elements $j \in J$, as shown in the next section. We assume that there is a logical way to enumerate the direct links k and the path indexes j. $\Theta = \{\theta_{k,j} : \forall k \in K, \forall j \in J\}$ is the set of all possible direct links $\theta_{k,j}$ between any $m \in \mathcal{M}^a$ to radio node n or between $m \in \mathcal{M}^a$ to $m' \in \mathcal{M}^i$. Path $l_j = \{\theta_{k,j} : \theta_{k,j} \in \Theta, \forall k \in K\}$ is the set of all direct links that connect $m \in \mathcal{M}^a$ with radio node n directly (one hop) or through $m' \in \mathcal{M}^i$ (two hops $m \in \mathcal{M}^a$ to $m' \in \mathcal{M}^i$ and $m' \in \mathcal{M}^i$ to radio node n). $\theta_k = \{\theta_{k,j} : \theta_{k,j} \in \Theta, \forall j \in J\}$ is the set of direct links $\theta_{k,j}$ that are using the same physical direct link (same physical resource).

Notice that a direct link $\theta_{k,j}$ is a label that identifies a physical direct link that is used by path l_j, therefore a physical direct link can have multiple labels associated with it. $S_m = \{l_j : (m-1)|\mathcal{M}| + 1 \leq j \leq m|\mathcal{M}|, \forall m \in \mathcal{M}^a\}$ is the collection set of paths from active mobile device m to radio node n. We define the path assignment indicator x_j as:

$$x_j = \begin{cases} 1, & \text{if path } l_j \text{ is selected for transmission} \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

The direct link assignment indicator $\alpha_{k,j}$ is defined as:

$$\alpha_{k,j} = \begin{cases} 1, & \text{if } x_j = 1, \forall k \in K \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

We also define the path capacity $\mathcal{C}_j$ such that every path l_j has a path capacity associated with it. The path capacity is defined as:

$$\mathcal{C}_j = \min_{k:\theta_{k,j} \in l_j} \{\alpha_{k,j} c_k\}, \tag{5.3}$$

where c_k is the link capacity, defined in the simplest case where it is subject to additive white Gaussian noise (AWGN), and can be calculated according to the Shannon-Hartley theorem [111] as:

$$c_k = B \log_2(1 + \text{SINR}_k), \tag{5.4}$$

where SINR is the signal to interference plus noise ratio, and B is the channel bandwidth.

The objective is to find the values of the path assignment indicator x_j that maximize the total capacity. This problem can be formulated as the following mixed integer non-linear programming optimization problem:

$$\max_{x_j} \sum_{j \in J} x_j \mathcal{C}_j \tag{5.5}$$

subject to:

$$\sum_{j:l_j \in S_m} x_j = 1 \qquad \forall m \in \mathcal{M}^a \tag{5.6}$$

$$\sum_{k:\theta_{k,j} \in l_j} \alpha_{k,j} \leq 2 \qquad \forall l_j \in S \tag{5.7}$$

$$\sum_{j:\theta_{k,j} \in \theta_k} \alpha_{k,j} \leq 1 \qquad \forall \theta_k \in \Theta \tag{5.8}$$

$$x_j \in \{0,1\} \qquad \forall j \in J, \tag{5.9}$$

where constraint (5.6) ensures that only one path is selected by any mobile device m, constraint (5.7) makes sure that there are at most two hops, and constraint (5.8) guarantees that any mobile device m can only act as a single relay, i.e., it can only relay one other mobile device at a time.

When solving this optimization problem, we should remember that in general, specially in the context of real-time applications, the problem of finding the optimal solution is NP-hard, therefore intractable when the number of mobile devices exceeds a few units. This is the reason why we focus our attention to the use of heuristics in the next section.

5.2.1 Index Assignment

We provide a detailed description of the set of rules used to enumerate the direct links and paths as well as how to create the sets used in this chapter.

- The set of path indexes J is defined as:

$$J = \{j : 1 \leq j \leq |\mathcal{M}|^2, j \in \mathbb{N}\}. \tag{5.10}$$

- The set of physical direct link indexes K is defined as:

$$K = \{k : 1 \leq k \leq \frac{|\mathcal{M}|(|\mathcal{M}|+1)}{2}, k \in \mathbb{N}\}. \tag{5.11}$$

- Assuming that paths are composed of a maximum of two hops (two direct links), we can define path l_j as:

$$l_j = \{\theta_{k,j} : k \in K_j\}, \tag{5.12}$$

where K_j is the set of all possible direct link indexes that path l_j can use. In order to simplify notation, let us first define $z = j - (m-1)|\mathcal{M}|$, then set K_j can be defined as:

$$K_j = \begin{cases} \{m\} & \text{if } z = 1 \\ \{z-1, & \\ \quad m - z + 1 + \sum_{i=|\mathcal{M}|-z+2}^{|\mathcal{M}|} i\}, & \text{if } 1 < z \leq m \\ \{z, & \\ \quad z - m + \sum_{i=|\mathcal{M}|-m+1}^{|\mathcal{M}|} i\}, & \text{otherwise ,} \end{cases} \tag{5.13}$$

where m is defined as:

$$m = \left\lceil \frac{j}{|\mathcal{M}|} \right\rceil. \tag{5.14}$$

- Given the definition of l_j we can write Θ as:

$$\Theta = \{\theta_{k,j} : \theta_{k,j} \in l_j, \forall k \in K, \forall j \in J\}. \tag{5.15}$$

- Then θ_k can also be defined as:

$$\theta_k = \{\theta_{k,j} : \theta_{k,j} \in l_j, \forall j \in J\}. \tag{5.16}$$

5.3 Greedy Algorithm

In this section[1], we present a heuristic method that finds a family of sub-optimal solutions to the maximization problem described in section 5.2. The goal of the algorithm is to find

[1]Oscar Delgado and Fabrice Labeau, "D2D relay selection and fairness on 5G wireless networks", accepted for publication in IEEE Globecom workshops, December 2016

the association between active mobile devices and relays that will help to improve their channel capacity. The algorithm can be summarized in three main steps:

- *First:* Get the capacity information of all the paths that have only one hop (direct link from mobile device to radio node),
- *Second:* Choose an active mobile device according to a given rule (minimum capacity, maximum capacity, choose at random, etc.),
- *Third:* From that specific active mobile device, find the path that has the largest capacity.

A detailed version of the greedy algorithm is described in Algorithm 9.

Greedy Algorithm 9 describes a family of solutions that depend on the order in which we choose the active mobile devices, i.e., which mobile device should have priority to start with the relay selection.

5.4 Complexity Analysis

Our greedy algorithm selects the best relay after conducting a linear search on the relay that maximizes the channel capacity. The complexity to select the best path is $\mathcal{O}(|\mathcal{M}^i|)$, the complexity of choosing the active mobile device is $\mathcal{O}(|\mathcal{M}^a|)$. Consequently, the computational complexity is on the order of $\mathcal{O}(|\mathcal{M}^a||\mathcal{M}^i|)$, i.e. the greedy algorithm has linear complexity in the number of active mobile devices and in the number of idle mobile devices. Our proposed greedy algorithm can further reduce complexity if we assume that every active mobile device can only communicate with a subset of the idle mobile devices so it is possible to find a good real-time solution with low complexity.

5.5 Numerical Results

In this section, we present the simulation settings used to assess the performance of our proposed solution and the numerical results. Fig. 5.2 presents the basic network architecture.

Algorithm 9 Greedy algorithm

Assumption: paths have maximum two links
Input: $\mathcal{M}^a, K, J, S_m, l_j, \Theta, \theta_k, c_k$
Calculate $\mathcal{C}_j, \forall j \in J$ assuming $\alpha_{k,j} \leftarrow 1, \forall \theta_{k,j} \in \Theta$
Set $\alpha_{k,j} \leftarrow 0, \forall \theta_{k,j} \in \Theta$
Set $x_j \leftarrow 0, \forall j \in J$
Define $S \leftarrow \mathcal{M}^a$
Define $C = \{\mathcal{C}_j : j \in J\}$
while S is not empty **do**
  Find $J^* = \{j : |l_j| = 1, \forall j \in \{j : \mathcal{C}_j \in C\}\}$
  $j' = f(\mathcal{C}_j)$ (we define $f(\mathcal{C}_j)$ in the next section)
  $m^* = \{m : l_{j'} \in S_m, \forall m \in \mathcal{M}^a\}$
  $Capacity \leftarrow 0$
  for every path $l_i \in S_{m^*}$ **do**
    $H = \{h : h = \sum_{j:\theta_{k^*,j} \in \theta_{k^*}} \alpha_{k^*,j}, \forall k^* \in \{k : \theta_{k,i} \in l_i\}\}$
    if $\sum_{h \in H} h < 1$ **then**
      if $\mathcal{C}_i > Capacity$ **then**
        $j^* \leftarrow i$
        $Capacity \leftarrow \mathcal{C}_i$
      end if
    end if
  end for
  $\alpha_{k,j^*} \leftarrow 1, \forall k \in \{k : \theta_{k,j^*} \in l_{j^*}\}$
  $x_{j^*} \leftarrow 1$
  $C \leftarrow C \setminus \mathcal{C}_j, \forall j \in \{j : l_j \in S_{m^*}\}$
  $S \leftarrow S \setminus m^*$
end while
$TotalCapacity = \sum_{j \in J} x_j \mathcal{C}_j$

5.5.1 Simulation Settings

We evaluate the performance of our greedy algorithms using simulations in MATLAB. Table 5.1 summarizes the simulation parameters.

We model the locations of the mobile devices as a homogeneous Poisson Point Process [112], meaning that after choosing the number of mobile users in a certain area, their locations follow a uniform distribution inside that area.

For simulation purposes we take into consideration the signaling overhead by using a reduction factor when calculating the channel capacity described by equation (5.3), see

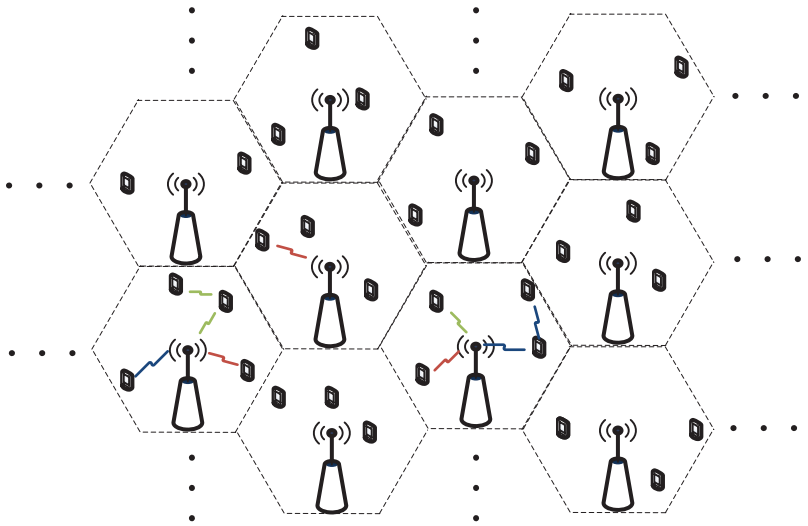


Fig. 5.2 Radio nodes deployment model

Table 5.1 D2D simulation parameters

Parameter	**value**
Cell radius r	500 m
Radio nodes	100
Mobile devices per radio node	100 to 1000
Mobile device location	Homogeneous PPP
Mobile device speed	static
Mobile device maximum transmit power ξ_{max}	max 200 mW
Power control parameter ξ_0	0.8μW
Active mobile devices per radio node	40%
Path loss exponent α_1, β	2, 4
Path loss breakpoint g	400 m
Signaling overhead Standard connectivity	max 3.5%
Signaling overhead Relaying (greedy algorithm)	max 7%

Table 5.1.

We model the transmission power consumption, assuming a slow power control mechanism [101] as:

$$\xi = \min\left(\xi^{max}, \xi_0 \frac{r^{\alpha_1}(1 + r/g)^{\beta}}{L_K}\right), \tag{5.17}$$

where ξ_{max} is the maximum power a mobile device can transmit, ξ_0 is the cell specific parameter used to control the target SINR, r is the distance between the mobile device and the radio node, g is the breakpoint of a path loss curve, α_1 is the basic path-loss exponent before breakpoint, α_2 is the additional path-loss exponent, $\beta = \alpha_2 - \alpha_1$ and L_K is the path loss constant.

In order to evaluate the performance of the network we use four scenarios:

- *Standard connectivity:* Every active mobile device establishes a direct communication with its respective radio node without using relays.
- *Relaying min:* We use greedy algorithm 9 with function $f(\mathcal{C}_j)$ defined as:

$$f(\mathcal{C}_j) = \arg\min_{j \in J^*} \{\mathcal{C}_j\}, \tag{5.18}$$

  where if there exists more than one $\mathcal{C}_j$ that minimizes equation (5.18), we randomly select one of the solutions.
- *Relaying random:* We use greedy algorithm 9, where function $f(\mathcal{C}_j)$ takes values chosen from a uniform distribution.
- *Relaying max:* We use greedy algorithm 9 with function $f(\mathcal{C}_j)$ defined as:

$$f(\mathcal{C}_j) = \arg\max_{j \in J^*} \{\mathcal{C}_j\}, \tag{5.19}$$

  where if there exists more than one $\mathcal{C}_j$ that maximizes equation (5.19), we randomly select one of the solutions.

5.5.2 Simulation Results

5.5.2.1 Impact on Fairness

One way to quantify the degree of fairness is shown in Fig. 5.3. We present the Jain's fairness index [113] (see equation 5.20) as a function of the total number of mobile devices. It could be observed that even though our optimization model did not explicitly include fairness, our greedy algorithms offer an improvement of more than 8% in the fairness index

with respect to the standard connectivity (direct connection to the radio node).

$$\text{Jain's fairness index} = \frac{(\sum_{j\in J} x_j \mathcal{C}_j)^2}{|\mathcal{M}^a| \sum_{j\in J} (x_j \mathcal{C}_j)^2} \tag{5.20}$$

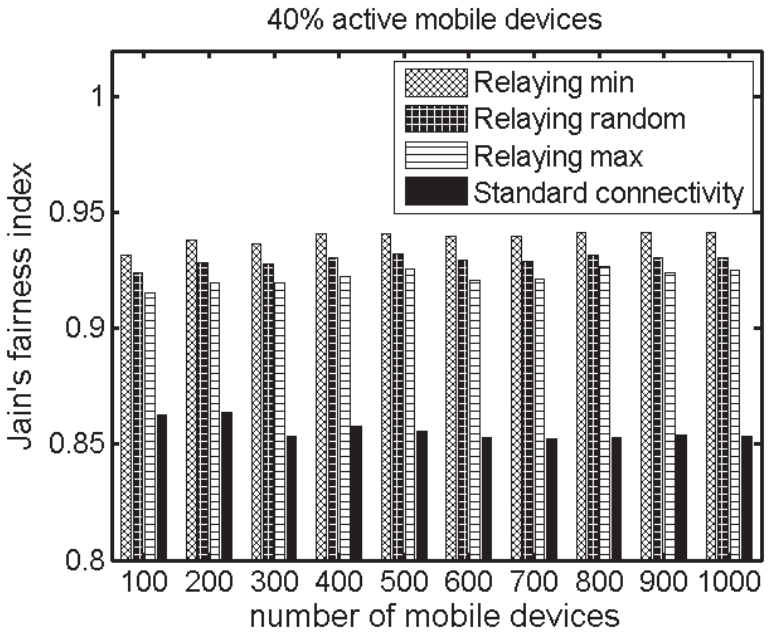


Fig. 5.3 Jain's fairness index versus number of mobile devices

Another way to see the fairness improvement is by looking at the CDF of the channel capacity on each mobile device as observed in Fig. 5.4. There we can see an improvement of about 50% in channel capacity on the mobile devices that have smaller capacity, while keeping the capacity of the mobile devices that already have higher capacity. This shows that the proposed strategy benefits the mobile devices at the edge of the cell without affecting the mobile devices that are close to the center of the cell.

5.5.2.2 Impact on Network Capacity

In order to analyze the impact in the overall system capacity, we present in Fig. 5.5 the total capacity versus the number of mobile devices. The results show a capacity gain of around 20% due to the fact that mobile devices at the edge of the cell have benefited from the relay selection strategy.

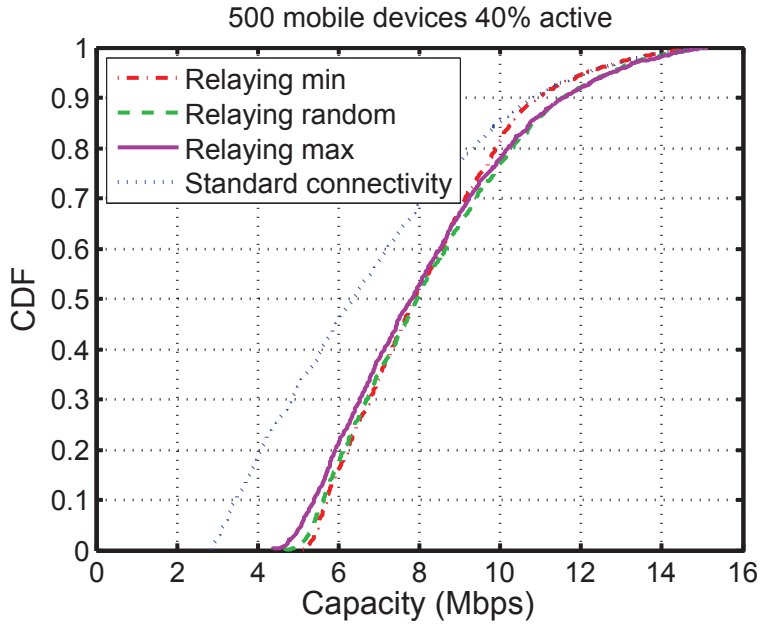


Fig. 5.4 Mobile device's throughput CDF

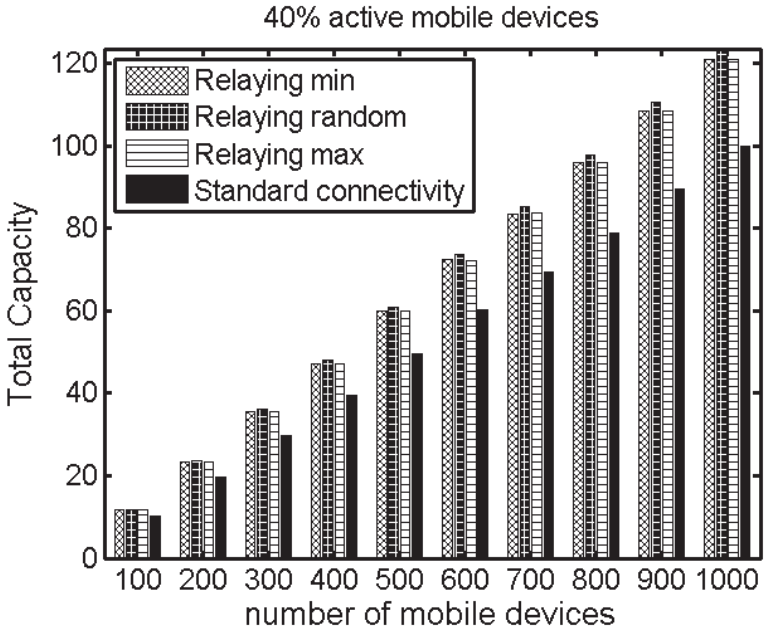


Fig. 5.5 Total capacity versus number of mobile devices

5.5.2.3 Impact on Power Consumption

Power consumption for different number of mobile devices is shown in Fig. 5.6. Notice that even though usually there is a trade-off between channel capacity and power consumption, in our case we observe an 80% reduction in transmitting power due to the fact that, by relaying, we are able to use direct links that are shorter, allowing us to reduce the amount of power needed to have a successful transmission.

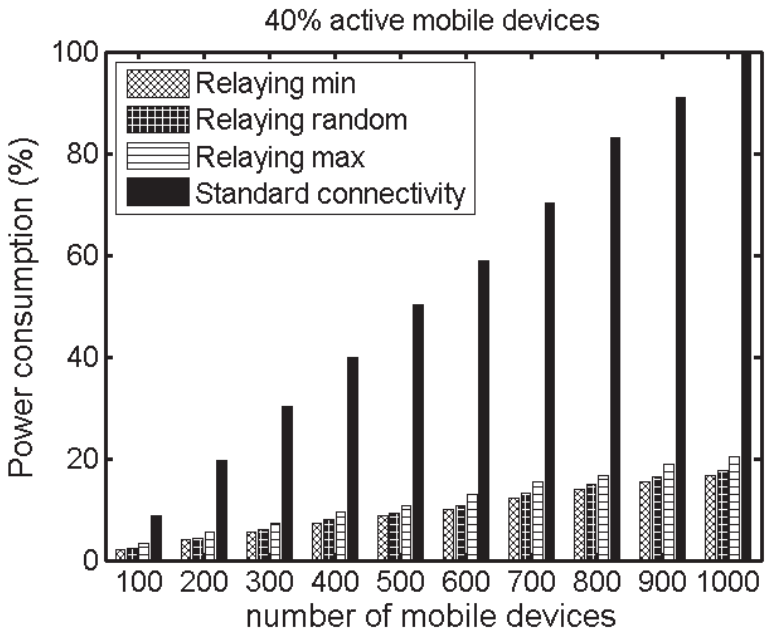


Fig. 5.6 Transmission power consumption versus number of mobile devices

Given that this new relay function consumes resources, it is also important to analyze the power consumption of the mobile devices acting as relays. To have an intuition on this we present in Fig. 5.7 the average power consumption of the mobile devices acting as relays and the power consumption of the active mobile devices over 100 runs. In the figure is clear to see that mobile devices acting as relays consume the same if not more than the active mobile devices. Nevertheless, we have to remember that in the long run all mobile devices not only move, but also arrive or leave the system, which could help either to harm or to benefit all the mobile devices.

We show in Fig. 5.8 the proportion of active mobile devices that use one hop and two hops connections. It is important to notice that the relaying min algorithm manages to

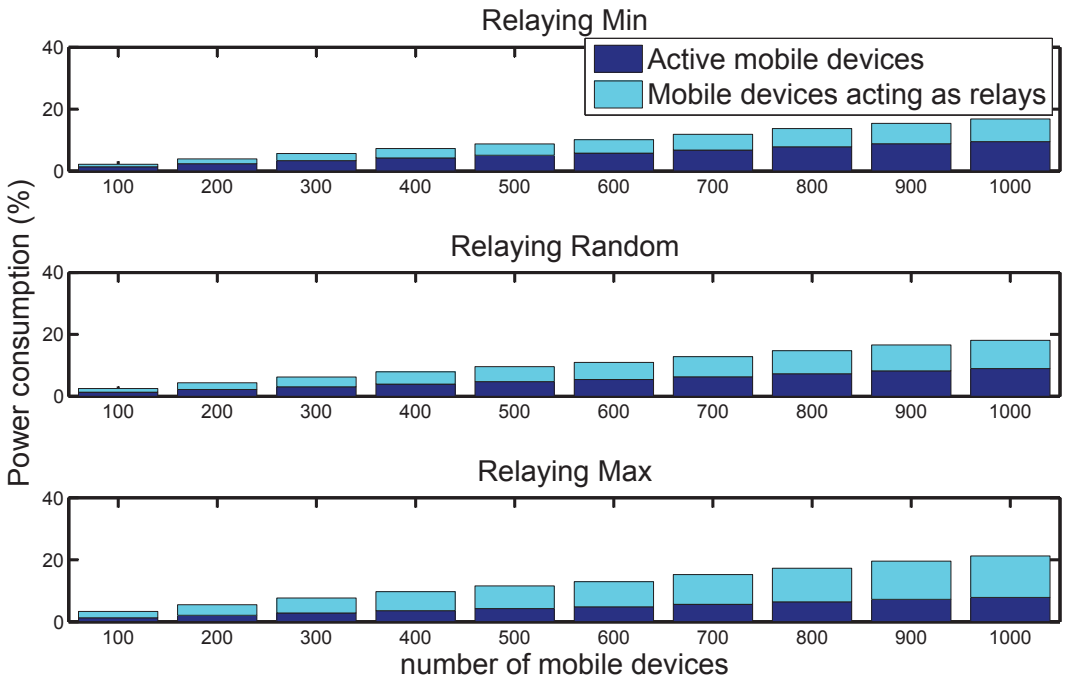


Fig. 5.7 Transmission power consumption per mobile device's type, 40% active mobile devices

use a higher number of one hop connections, allowing the system to be more fair as seen in Fig. 5.3, while the relaying random algorithm achieves slightly higher capacity gains.

We can also evaluate the system across multiple percentages of active mobile devices. Fig. 5.9 shows Jain's fairness index. We can observe from the figure that Relaying min and Relaying random strategies, manage to always improve fairness, even at a higher percentage of active mobile devices, but the Relaying max strategy improves fairness at lower percentages of the active mobile device. This behavior can be explained by the fact that Relaying max first assigns relays to the mobile devices closer to the radio node, consuming the "best" relays, consequently leaving the system with a smaller number of relaying options. It is important to mention here that in practice network operators dimension the wireless network such that the percentage of active mobile devices is smaller than 60%.

Fig. 5.10 shows results on the total network capacity. There we can see that higher gains can be obtained if there are less number of active mobile devices. This is because if there are more idle mobile devices available to choose from, this enables the system to have the opportunity to maximize capacity gains.

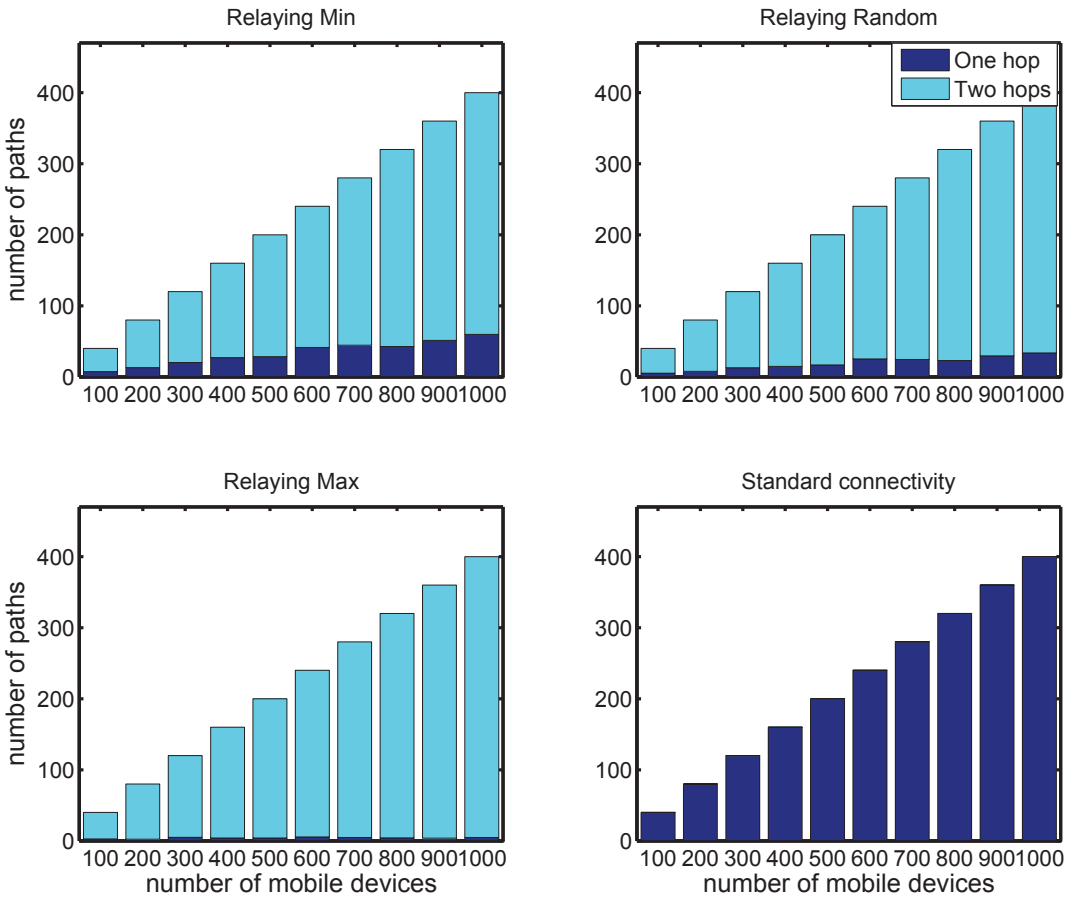


Fig. 5.8 Path assignment (number of paths using one hop and two hops), 40% active mobile devices

In Fig. 5.11 we can observe the transmission power gains for different values of the percentage of active mobile devices. Again, we can see here that important gains can be obtained using Relaying min and Relaying random strategies. Although Relaying max does not manage to report transmission power gains when the percentage of active mobile devices is above 50%, it still has a marginal improvement in total capacity gains. This can be explained by the fact that even though relaying can offer benefits in maximizing capacity, if the mobile devices are too far away from each other, it forces the active mobile devices to transmit at close to maximal mobile device power, which at the end increases the overall power consumption.

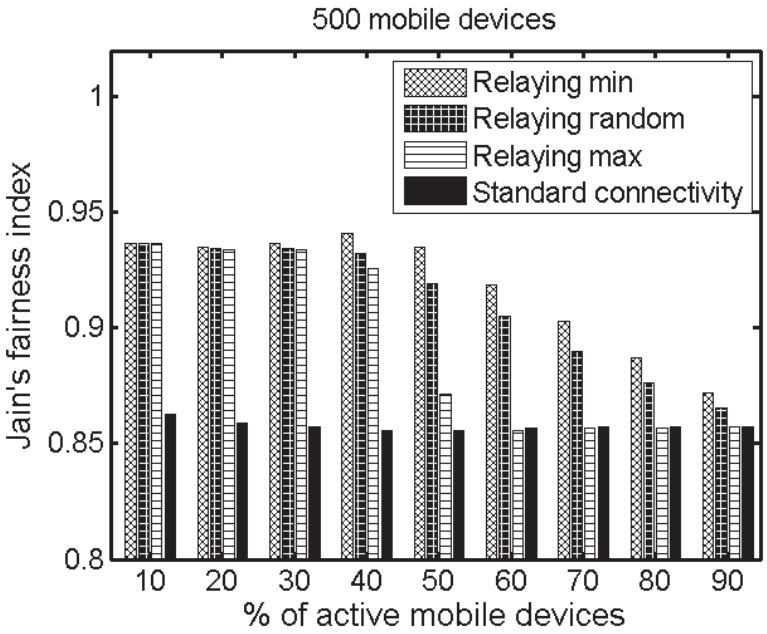


Fig. 5.9 Jain's fairness index versus percentage of active mobile devices

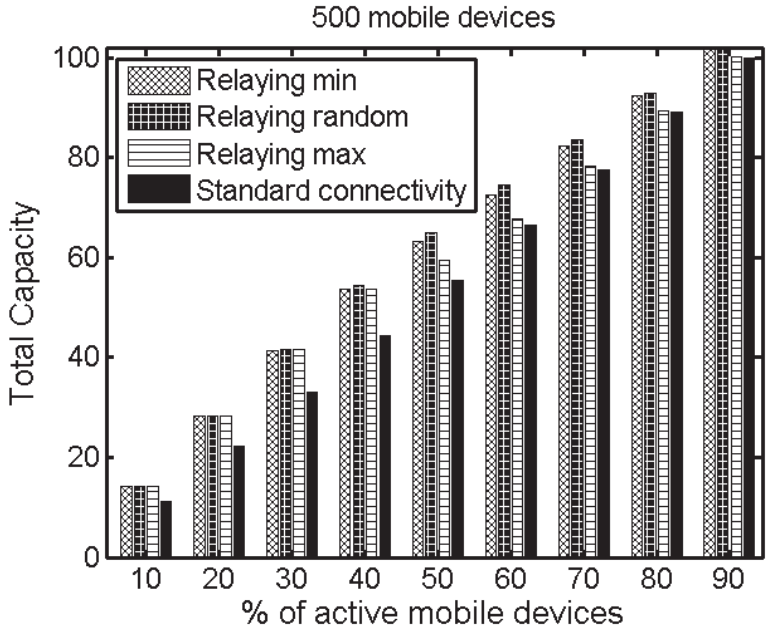


Fig. 5.10 Total capacity versus percentage of active mobile devices

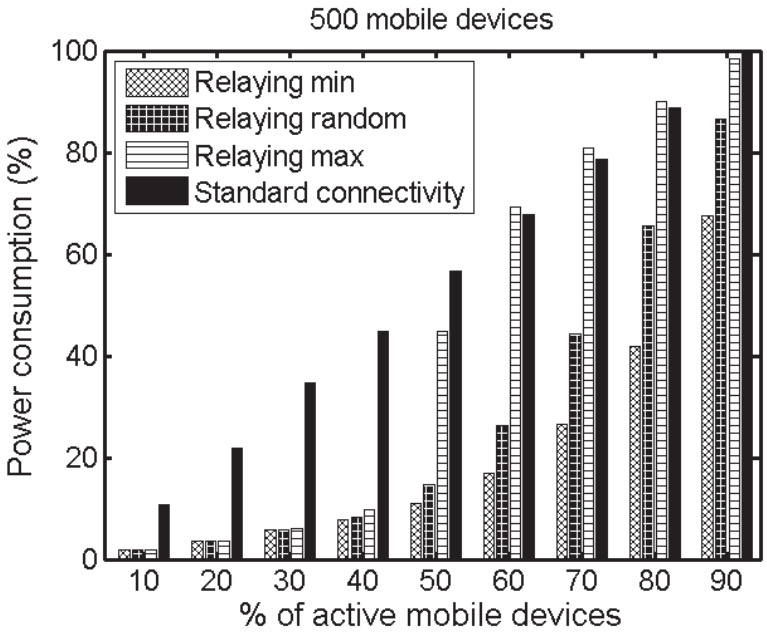


Fig. 5.11 Transmission power consumption versus percentage of active mobile devices

5.6 Conclusion

In this chapter, we have dealt with the problem of how to improve mobile devices' throughput at the cell edges (throughput fairness) by devising the use of mobile devices with multiple network interfaces that act as full duplex relays. First, we have proposed an objective optimization problem, and then we have described a detailed family of greedy algorithms that finds a low complexity set of sub-optimal relay selection solutions.

We formulate the problem as an optimization problem and propose a heuristic method for selecting the relays. To evaluate our proposed algorithms, we have implemented our solution using MATLAB simulations. The contributions of this chapter can be described as follows:

- We propose an infrastructure based solution (controlled by the network operator) that enables relaying capabilities on mobile devices with multiple network interfaces.
- We define an objective optimization problem that maximizes the total network capacity, considering that connections can have at most one relay (two hops).
- We propose a sub-optimal family of greedy algorithms, that exhibit good performance

in terms of total capacity, average transmission power, and user throughput fairness.

- Numerical results show that our relay selection strategies manage to increase the total network capacity while reducing the transmission power consumption and increasing the fairness of the system.

Chapter 6

Conclusions and Future Work

In this thesis, we have investigated the video traffic management problem over wireless networks. While video traffic management has received significant attention in recent years, to the best of our knowledge, this is the first work that designs an SDN-based solution over wireless networks considering that mobile devices can be connected simultaneously to multiple radio nodes of the same technology. In order to deal with the network capacity constraints derived from the expected demand of video traffic, we have proposed new load balancing and D2D relaying strategies that help us to make a better use of the resources on the first case and to increase the channel capacity in the second case. It has been shown that it is possible not only to make a better use of the network capacity resources available and to accomplish significant capacity gains, but also to reduce power consumption, without sacrificing quality of service. This chapter summarizes the work presented in this thesis and suggests potential topics for future research.

6.1 Thesis Summary

In Chapter 2, we have highlighted the necessity to improve the current video transmission management techniques over wireless cellular networks, especially given that not much has been done on a multi-radio multi-interface system with focus on video traffic. The literature review shows that SDN and network virtualization offers good tools to tackle the admission control and scheduling challenges of video management, it also exposes the problems of network capacity saturation and load balancing due to the multipath assumption.

In Chapter 3, we have explored the admission control, scheduling and virtualization-

based slicing of traffic to design and propose the Joint Slice Manager (JSM) framework. Although JSM performance evaluation is currently focused only on an LTE-based wireless network, building JSM over the SDN framework enables us to extend it over other wireless technologies. Additionally JSM does not require any modifications to the mobile device (client) or the server, facilitating a quicker implementation. Simulation results show significant improvements not only in video quality, but also in radio node's utilization. One important characteristic of our design is the capability to give service to mobile devices connected simultaneously to more than one radio node of the same technology.

We have designed in Chapter 4 three multipath load balancing algorithms (QBALAN, DALBA and GEL) that deal with the problem of how to split traffic in the uplink such that the traffic load remains at the desired load. While the load balancing problem has received significant attention in recent years, to the best of our knowledge, this is one of the few works that considers a scenario where mobile devices with multiple network interfaces are connected simultaneously to multiple radio nodes. Our results show significant improvements not only in splitting error, but also in PSNR and power consumption.

Finally, in Chapter 5, we have demonstrated that we can improve not only the overall network capacity, but also the throughput at the cell edges (throughput fairness) by devising the use of mobile devices with multiple network interfaces that act as full duplex relays. First, we have proposed an objective optimization problem, and then we have described a detailed family of greedy algorithms that find a low complexity set of sub-optimal relay selection solutions.

It is also worth mentioning that some of the practical benefits of the research presented in this thesis can be analyzed not only from the network operator perspective, but also from the user perspective. The development of new techniques that allow us to increase the network capacity and its reliability, represent some benefits for the network operators by enabling them to either serve more users or to increase the available bandwidth of the current ones. Similarly from the user perspective, it benefits them by having the possibility of obtaining higher data rates and more reliable communications.

6.2 Future Work

The future directions can be summarized in three general areas, namely the Joint Slice Manager (JSM) framework, traffic load balancing and relaying D2D communication. These

are discussed separately in this section.

6.2.1 JSM Framework

Generalization to multi-RAT systems will envision a system with potentially multiple access technologies coexisting to provide content to users. The difficulty in generalizing to multi-RAT systems lie in (i) the management of different access technologies with their associated resource access/contention mechanisms, and (ii) the global optimization of delivery scenarios for multiple potential interfaces for each user.

In the JSM framework we have used a simple user satisfaction metric P_w (equation (3.3)) to derive flow migration decisions. It will be interesting to extend our studies by including additional mobile device-based metrics fed back to the control plane (e.g. buffer occupancy, loss rate, etc.) to enhance this long-term metric. Investigation of how much knowledge can be gained from more information about video contents, through Deep Packet Inspection (DPI) can also be a good avenue of research.

6.2.2 Load Balancing

So far, we have tested our proposed algorithms independently of the JSM framework, it will be important to incorporate our load balancing solutions into the JSM framework and verify its performance. Although our algorithms were designed for uplink scenarios, and can easily be extended to downlink, it will be helpful to extend the simulations to both scenarios simultaneously. Furthermore, another important aspect will be to explore the best way to define/calculate the desired load K_n^m, which was assumed to be given in our experiments.

6.2.3 D2D Communication

In Chapter 5, we have shown the possible gains in capacity and power consumption that can be obtained when using D2D relays (one relay, two hops). It will be important to extend our study to the more general case with the possibility to have a higher number of hops (> 2), in which mobile devices are arriving and departing the system, as well as users are moving within the system (handover). Another aspect to analyse in detail is the possible battery drain of some mobile devices that are used most of the time as relays. This may require the design of pricing models that can help to have a fairer use of resources.

References

[1] ITU-R, "Future technology trends of terrestrial IMT systems," Report M.3020, International Telecommunication Union, Geneva, November 2014.

[2] Cisco Systems Inc., "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2015-2020," February 2016.

[3] M. Y. Arslan, K. Sundaresan, and S. Rangarajan, "Software-defined networking in cellular radio access networks: potential and challenges," *IEEE Communications Magazine*, vol. 53, pp. 150–156, January 2015.

[4] R. Wang, H. Hu, and X. Yang, "Potentials and Challenges of C-RAN Supporting Multi-RATs Toward 5G Mobile Networks," *IEEE Access*, vol. 2, pp. 1187–1195, October 2014.

[5] C.-C. Lin, H.-H. Chin, and D.-J. Deng, "Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System," *IEEE Systems Journal*, vol. 8, pp. 225–234, March 2014.

[6] D. Kaspar, K. Evensen, A. Hansen, P. Engelstad, P. Halvorsen, and C. Griwodz, "An analysis of the heterogeneity and IP packet reordering over multiple wireless networks," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 637–642, July 2009.

[7] P. Mach, Z. Becvar, and T. Vanek, "In-Band Device-to-Device Communication in OFDMA Cellular Networks: A Survey and Challenges," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 1885–1922, Fourth quarter 2015.

[8] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1617–1634, August 2014.

[9] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, pp. 14–76, January 2015.

[10] N. A. Jagadeesan and B. Krishnamachari, "Software-Defined Networking Paradigms in Wireless Networks: A Survey," *ACM Computing Surveys*, vol. 47, pp. 1–11, January 2015.

[11] C. Cox, *An introduction to LTE : LTE, LTE-advanced, SAE, and 4G mobile communications*. John Wiley & Sons Ltd, March 2012.

[12] L. Li, Z. Mao, and J. Rexford, "Toward Software-Defined Cellular Networks," *IEEE European Workshop on Software Defined Networking (EWSDN)*, pp. 7–12, October 2012.

[13] L. Li, Z. Mao, and J. Rexford, "CellSDN: Software-Defined Cellular Networks," tech. rep., TR-922-12, Princeton University, Department of Computer Science, April 2012.

[14] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: concept, survey, and research directions," *IEEE Communications Magazine*, vol. 53, pp. 126–133, November 2015.

[15] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for introducing innovation into wireless mobile networks," *ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures (VISA)*, pp. 25–32, September 2010.

[16] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 1333–1346, September 2012.

[17] R. Mahindra, R. Kokku, H. Zhang, and S. Rangarajan, "MESA: Farsighted flow management for video delivery in broadband wireless networks," *IEEE Third International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–10, January 2011.

[18] G. Bhanage, I. Seskar, R. Mahindra, and D. Raychaudhuri, "Virtual basestation: architecture for an open shared WiMAX framework," *ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures (VISA)*, pp. 1–8, September 2010.

[19] K.-K. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazemian, and N. McKeown, "The Stanford OpenRoads deployment," *ACM workshop on Experimental evaluation and characterization (WINTECH)*, pp. 59–66, September 2009.

[20] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: empowering research in mobile networks," *ACM SIGCOMM Computer Communication Review*, vol. 40, pp. 125–126, January 2010.

[21] K. Nakauchi and Y. Shoji, "WiFi Network Virtualization to Control the Connectivity of a Target Service," *IEEE Transactions on Network and Service Management*, vol. 12, pp. 308–319, June 2015.

[22] C. Wang and M. Zink, "QoS featured wireless virtualization based on 802.11 hardware," *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pp. 801–805, September 2012.

[23] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, "Virtual network functions orchestration in wireless networks," *11th International Conference on Network and Service Management (CNSM)*, pp. 108–116, November 2015.

[24] K. Nakauchi, Y. Shoji, and N. Nishinaga, "Airtime-based resource control in wireless LANs for wireless network virtualization," *Fourth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 166–169, July 2012.

[25] S. Costanzo, L. Galluccio, G. Morabito, and S. Palazzo, "Software Defined Wireless Networks: Unbridling SDNs," *IEEE European Workshop on Software Defined Networking (EWSDN)*, pp. 1–6, October 2012.

[26] P. Lv, X. Wang, and M. Xu, "Virtual access network embedding in wireless mesh networks," *ACM Ad Hoc Networks*, vol. 10, pp. 1362–1378, September 2012.

[27] R. Matos, S. Sargento, K. Hummel, A. Hess, K. Tutschku, and H. Meer, "Context-based wireless mesh networks: a case for network virtualization," *Springer Telecommunication Systems*, vol. 51, pp. 259–272, December 2012.

[28] T. Chen, H. Zhang, X. Chen, and O. Tirkkonen, "SoftMobile: control evolution for future heterogeneous mobile networks," *IEEE Wireless Communications*, vol. 21, pp. 70–78, December 2014.

[29] D. Raychaudhuri and M. Gerla, "New Architectures and Disruptive Technologies for the Future Internet: The Wireless," tech. rep., Mobile and Sensor Network Perspective, Report of NSF WMPG Workshop, August 2005.

[30] W. W. Group, "Technical Document on Wireless Virtualization," tech. rep., GDD-06-17 GENI: Global Environment for Network Innovations, September 2006.

[31] V.-G. Nguyen, T.-X. Do, and Y. Kim, "SDN and Virtualization-Based LTE Mobile Network Architectures: A Comprehensive Survey," *Wireless Personal Communications*, vol. 86, pp. 1401–143, August 2015.

[32] C. Liang and F. R. Yu, "Wireless virtualization for next generation mobile cellular networks," *IEEE Wireless Communications*, vol. 22, pp. 61–69, February 2015.

[33] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "CellSlice: Cellular wireless resource slicing for active RAN sharing," *IEEE Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–10, January 2013.

[34] D. Klein and M. Jarschel, "An openflow extension for the omnet++ inet framework," *6th International Workshop on OMNeT++*, March 2013.

[35] V. D. Philip, Y. Gourhant, and D. Zeghlache, *e-Infrastructure and e-Services for Developing Countries*, ch. OpenFlow as an Architecture for e-Node B Virtualization, pp. 49-63, November, 2011.

[36] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A network virtualization layer," *OpenFlow Switch Consortium, Tech. Rep*, pp. 1–13, October 2009.

[37] S. Costanzo, D. Xenakis, N. Passas, and L. Merakos, "OpeNB: A framework for virtualizing base stations in LTE networks," *IEEE International Conference on Communications (ICC)*, pp. 3148–3153, June 2014.

[38] J. Kempf, B. Johansson, S. Pettersson, H. Lning, and T. Nilsson, "Moving the mobile Evolved Packet Core to the cloud," *8th International IEEE Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 784–791, October 2012.

[39] Q. Liu and L. N. Bhuyan, "fAHRW+: Fairness-aware and locality-enhanced scheduling for multi-server systems," *20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 63–70, December 2014.

[40] K.-K. Yap, *Using All Networks Around Us.* PhD thesis, Stanford University, California, USA, 2013.

[41] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri, "SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, December 2010.

[42] L. Gharai, C. Perkins, and T. Lehman, "Packet reordering, high speed networks and transport protocol performance," *IEEE International Conference on Computer Communications and Networks (ICCCN)*, pp. 73–78, October 2004.

[43] S. Tinta, A. Mohr, and J. Wong, "Characterizing End-to-End Packet Reordering with UDP Traffic," *IEEE Symposium on Computers and Communications*, pp. 321–324, July 2009.

[44] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang, "Experience in Measuring Internet Backbone Traffic Variability: Models, Metrics, Measurements and Meaning," *in Proceedings of the International Teletraffic Congress (ITC-18)*, pp. 379–388, December 2003.

[45] M. Przybylski, B. Belter, and A. Binczewski, "Shall we worry about packet reordering," *Computational Methods in Science and Technology*, vol. 11, pp. 141–146, January 2005.

[46] S. Spirou, "Packet Reordering Effects on the Subjective Quality of Broadband Digital Television," *IEEE Tenth International Symposium on Consumer Electronics (ISCE)*, pp. 1–6, September 2006.

[47] G. Detal, C. Paasch, S. Van Der Linden, P. MéRindol, G. Avoine, and O. Bonaventure, "Revisiting Flow-based Load Balancing: Stateless Path Selection in Data Center Networks," *Computer Networks*, vol. 57, pp. 1204–1216, April 2013.

[48] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On Load Distribution over Multipath Networks," *IEEE Communications Surveys Tutorials*, vol. 14, pp. 662–680, March 2012.

[49] A. L. Ramaboli, O. E. Falowo, and A. H. Chan, "Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues," *Elsevier Journal of Network and Computer Applications*, vol. 35, pp. 1674 – 1690, November 2012.

[50] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic Load Balancing Without Packet Reordering," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 51–62, March 2007.

[51] J.-O. Kim, P. Davis, T. Ueda, and S. Obana, "Splitting downlink multimedia traffic over WiMAX and WiFi heterogeneous links based on airtime-balance," *Wireless Communications and Mobile Computing*, vol. 12, pp. 598–614, July 2010.

[52] F. Zhong, C. K. Yeo, and B. S. Lee, "Adaptive load balancing algorithm for multiple homing mobile nodes," *Elsevier Journal of Network and Computer Applications*, vol. 35, pp. 316 – 327, January 2012.

[53] L. Shi, B. Liu, C. Sun, Z. Yin, L. Bhuyan, and H. Chao, "Load-Balancing Multipath Switching System with Flow Slice," *IEEE Transactions on Computers*, vol. 61, pp. 350–365, March 2012.

[54] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "Effective Delay-Controlled Load Distribution over Multipath Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 1730–1741, October 2011.

[55] M. Li, H. Nishiyama, N. Kato, K. Mizutani, O. Akashi, and A. Takahara, "On the fast-convergence of delay-based load balancing over multipaths for dynamic traffic environments," *Wireless Communications Signal Processing*, pp. 1–6, October 2013.

[56] K. Hashiguchi, Y. Kon, M. Hasegawa, K. Ishizu, and H. Harada, "Traffic allocation control using support vector machine in heterogeneous wireless link aggregation," *IEEE Consumer Communications and Networking Conference*, pp. 653–657, January 2011.

[57] J. Wu, J. Yang, Y. Shang, B. Cheng, and J. Chen, "SPMLD: Sub-packet based multipath load distribution for real-time multimedia traffic," *Journal of Communications and Networks*, vol. 16, pp. 548–558, October 2014.

[58] J. Lee, N. Dinh, G. Hwang, J. K. Choi, and C. Han, "Power-Efficient Load Distribution for Multihomed Services With Sleep Mode Over Heterogeneous Wireless Access Networks," *IEEE Transactions on Vehicular Technology*, vol. 63, pp. 1843–1854, May 2014.

[59] J. Wu, C. Yuen, N. M. Cheung, and J. Chen, "Delay-Constrained High Definition Video Transmission in Heterogeneous Wireless Networks with Multi-Homed Terminals," *IEEE Transactions on Mobile Computing*, vol. 15, pp. 641–655, March 2016.

[60] J. Wu, C. Yuen, B. Cheng, Y. Shang, and J. Chen, "Goodput-Aware Load Distribution for Real-Time Traffic over Multipath Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, pp. 2286–2299, August 2015.

[61] A. Asadi, Q. Wang, and V. Mancuso, "A Survey on Device-to-Device Communication in Cellular Networks," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1801–1819, Fourth quarter 2014.

[62] R. Alkurd, R. M. Shubair, and I. Abualhaol, "Survey on device-to-device communications: Challenges and design issues," *New Circuits and Systems Conference (NEWCAS)*, pp. 361–364, June 2014.

[63] D. Karvounas, A. Georgakopoulos, K. Tsagkaris, V. Stavroulaki, and P. Demestichas, "Smart management of D2D constructs: an experiment-based approach," *IEEE Communications Magazine*, vol. 52, pp. 82–89, April 2014.

[64] S. Andreev, O. Galinina, A. Pyattaev, K. Johnsson, and Y. Koucheryavy, "Analyzing Assisted Offloading of Cellular User Sessions onto D2D Links in Unlicensed Bands," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 67–80, January 2015.

[65] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multihop device-to-device communications," *IEEE Communications Magazine*, vol. 52, pp. 56–65, April 2014.

[66] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A New Approach for Spectrum Sharing in Device-to-Device Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 1139–1151, June 2014.

[67] X. Lin, J. G. Andrews, and A. Ghosh, "A comprehensive framework for device-to-device communications in cellular networks," *arXiv preprint*, pp. 1305–4219, May 2013.

[68] M. Hasan and E. Hossain, "Distributed Resource Allocation for Relay-Aided Device-to-Device Communication Under Channel Uncertainties: A Stable Matching Approach," *IEEE Transactions on Communications*, vol. 63, pp. 3882–3897, October 2015.

[69] K. Vanganuru, S. Ferrante, and G. Sternberg, "System capacity and coverage of a cellular network with D2D mobile relays," *Military Communications Conference (MILCOM)*, pp. 1–6, October 2012.

[70] N. Mastronarde, V. Patel, J. Xu, L. Liu, and M. van der Schaar, "To Relay or Not to Relay: Learning Device-to-Device Relaying Strategies in Cellular Networks," *IEEE Transactions on Mobile Computing*, vol. PP, pp. 1–19, August 2015.

[71] G. Kollias, F. Adelantado, K. Ramantas, and C. Verikoukis, "CORE: A Clustering Optimization Algorithm for Resource Efficiency in LTE-A Networks," *IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, December 2015.

[72] M. M. Miao, J. Sun, and S. X. Shao, "A Cross-Layer Relay Selection Algorithm for D2D Communication System," *IEEE Wireless Communication and Sensor Network (WCSN)*, pp. 448–453, December 2014.

[73] C. Zhengwen, Z. Su, and S. Shixiang, "Research on relay selection in device-to-device communications based on maximum capacity," *IEEE Information Science, Electronics and Electrical Engineering (ISEEE)*, vol. 3, pp. 1429–1434, April 2014.

[74] S. Xu and K. S. Kwak, "Effective Interference Coordination for D2D Underlaying LTE Networks," *IEEE Vehicular Technology Conference (VTC)*, pp. 1–5, May 2014.

[75] Y. Xu, Y. Liu, and D. Li, "Resource management for interference mitigation in device-to-device communication," *IET Communications*, vol. 9, pp. 1199–1207, June 2015.

[76] V. Venkatasubramanian, F. S. Moya, and K. Pawlak, "Centralized and decentralized multi-cell D2D resource allocation using flexible UL/DL TDD," *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 305–310, March 2015.

[77] B. Chen, J. Zheng, and Y. Zhang, "A time division scheduling resource allocation algorithm for D2D communication in cellular networks," *IEEE International Conference on Communications (ICC)*, pp. 5422–5428, June 2015.

[78] C. Yang, J. Han, and X. Xu, "How D2D communication influences energy efficiency of small cell network with sleep scheme," pp. 1690–1695, September 2014.

[79] Y. Zhao, Y. Li, H. Zhang, N. Ge, and J. Lu, "Fundamental Tradeoffs on Energy-Aware D2D Communication Underlaying Cellular Networks: A Dynamic Graph Approach," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 864–882, April 2016.

[80] M. Wang and Z. Yan, "Security in D2D Communications: A Review," *IEEE Trustcom/BigDataSE/ISPA*, vol. 1, pp. 1199–1204, August 2015.

[81] S. Mukherjee, *Analytical Modeling of Heterogeneous Cellular Networks: Geometry, Coverage, and Capacity*. Cambridge University Press; 1st edition, February 2014.

[82] J. Rodriguez, *Fundamentals of 5G Mobile Networks*. Wiley, April 2015.

[83] R. Arpaci-Dusseau and A. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces v0.91*. Arpaci-Dusseau Books, LLC, 2015.

[84] 3rd Generation Partnership Project, "3GPP specification TS 36.104: E-UTRA Base Station (BS) radio transmission and reception," June 2009.

[85] D. Bertsekas, *Nonlinear Programming: 2nd Edition*. Athena Scientific, Belmont, MA., 1999.

[86] B. Radojevic and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments," *Proceedings of the 34th International Convention MIPRO*, pp. 416–420, May 2011.

[87] A. Alheid, A. Doufexi, and D. Kaleshi, "A study on MPTCP for tolerating packet reordering and path heterogeneity in wireless networks," *Wireless Days (WD)*, pp. 1–7, March 2016.

[88] A. Jayasumana, N. Piratla, T. Banka, A. Bare, and R. Whitner, "Improved Packet Reordering Metrics," RFC 5236, June 2008.

[89] B. Ye, A. Jayasumana, and N. Piratla, "On Monitoring of End-to-End Packet Reordering over the Internet," *International Conference on Networking and Services (ICNS)*, pp. 3–3, July 2006.

[90] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid-A framework for video transmission and quality evaluation," *In Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pp. 255–272, September 2003.

[91] *e-Handbook of Statistical Methods: Single Exponential Smoothing*. NIST/SEMATECH at the National Institute of Standards and Technology, April 2012.

[92] W. Zhang and J. He, "Modeling End-to-End Delay Using Pareto Distribution," *Second International Conference on Internet Monitoring and Protection (ICIMP)*, pp. 21–21, July 2007.

[93] M. Caramia and P. Dell'Olmo, *Multi-objective Management in Freight Logistics Increasing Capacity, Service Level and Safety with Optimization Algorithms*. Springer, 2008.

[94] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: A bibliography review," *Journal of Global Optimization*, vol. 5, pp. 291–306, February 1994.

[95] A. A. Pessoa, M. Poss, and M. C. Roboredo, "Solving bilevel combinatorial optimization as bilinear min-max optimization via a branch-and-cut algorithm," *Annals of XLV Brazilian Symposium of Operations Research*, vol. 2, pp. 2497–2508, September 2013.

[96] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of Operations Research*, vol. 153, pp. 235–256, April 2007.

[97] J. Kleinberg and E. Tardos, *Algorithm Design*. Pearson Education, 2006.

[98] G. Attiya and Y. Hamam, "Two phase algorithm for load balancing in heterogeneous distributed systems," *IEEE 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, pp. 434–439, February 2004.

[99] S. Tomar, "Converting video formats with ffmpeg," *Linux journal*, vol. 2006, p. 10, June 2006.

[100] L. Lam, K. Su, C. Chan, and X. Liu, "Modeling of round trip time over the internet," *Asian Control Conference*, pp. 292–297, August 2009.

[101] M. Shakir, H. Tabassum, and M.-S. Alouini, "Analytical Bounds on the Area Spectral Efficiency of Uplink Heterogeneous Networks Over Generalized Fading Channels," *IEEE Transactions on Vehicular Technology*, vol. 63, pp. 2306–2318, June 2014.

[102] 5G PPP Pre-standardisation Working Group, "Issues Paper," October 2015.

[103] NTT DOCOMO INC, "5G Radio Access: Requirements, Concept and technologies," July 2014.

[104] Ericsson, "White paper: 5G Energy Performance," April 2015.

[105] H. Shi, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, "Fairness in Wireless Networks:Issues, Measures and Challenges," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 5–24, First quarter 2014.

[106] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-Device Communication in LTE-Advanced Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 1923–1940, Fourth quarter 2015.

[107] D. Burghal and A. F. Molisch, "Efficient Channel State Information Acquisition for Device-to-Device Networks," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 965–979, February 2016.

[108] H. Tang, Z. Ding, and B. Levy, "Enabling D2D Communications Through Neighbor Discovery in LTE Cellular Networks," *IEEE Transactions on Signal Processing*, vol. 62, pp. 5157–5170, October 2014.

[109] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *IEEE Communications Magazine*, vol. 50, pp. 170–177, March 2012.

[110] Y. Li, L. Zhang, X. Tan, and B. Cao, "An advanced spectrum allocation algorithm for the across-cell D2D communication in LTE network with higher throughput," *China Communications*, vol. 13, pp. 30–37, April 2016.

[111] S. Haykin, *Communication Systems*. Wiley Publishing, 5th ed., March 2009.

[112] J. G. Andrews, R. K. Ganti, M. Haenggi, N. Jindal, and S. Weber, "A primer on spatial modeling and analysis in wireless networks," *IEEE Communications Magazine*, vol. 48, pp. 156–163, November 2010.

[113] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," tech. rep., Digital Equipment Corporation, September 1984.