Mobility-Aware Virtual Network Embedding Techniques for Next-Generation Mobile Networks

Chochlidakis, Georgios

Awarding institution:
King's College London

University of London
King's College London
Department of Informatics

Mobility-Aware Virtual Network Embedding Techniques for Next-Generation Mobile Networks

Georgios Chochlidakis

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Telecommunications of King's College London,
Informatics Department 30 September 2017

Abstract

Network virtualisation has become one of the most prominent solutions for sustainability towards the dramatic increase of data demand in next-generation mobile networks. In addition, apart from increasing the overall infrastructure utilisation, it also greatly improves the manageability, the scalability and the robustness of the network. In order to allow multiple virtual networks to coexist in the same substrate network, the need for efficient network sharing techniques is imperative.

The main purpose of this work is to provide a holistic optimization framework for virtual network embedding solutions, where the actual user mobility effect is explicitly considered. First, the main focus is given on the study of the mobility effect and the impact of the mobility management techniques on the end-to-end communication of the mobile user. A hybrid-distributed mobility management scheme is proposed and compared against the latest mobility management schemes. Then, an optimisation framework for efficient mobility-aware virtual network embedding is proposed and evaluated by comparison with other works from the literature. Moving deeper in the area of virtual network embedding, the focus is given on minimizing the end-to-end delay and providing service differentiation, allowing in this way delay sensitive services to use the formed virtual networks with the minimum possible delay, as opposed to other more elastic services that use the same substrate network. The last part of this work is the study and the analysis of the stochastic nature of the virtual network embedding parameters and the proposal of an optimisation framework for adjustable-robustness virtual network embedding.

Driven by the benefits from virtualising the network and its functions, research as well as industry are expected to exploit in a greater degree than today the merits of this concept. The co-existence of multiple tenants not only will greatly change the network industry from a business perspective, but also will emphasise the need for more efficient and flexible network sharing techniques. This work belongs to the initial efforts to embrace and adopt the virtualisation concept in the next-generation wireless networks.

Acknowledgements

I would like to express my ultimate appreciation to:

- My supervisor, Dr Vasilis Friderikos, for his support and his precious guidance during this wonderful collaboration of the past four years.

- My second supervisor Prof Hamid Aghvami for his inspiring mentorship.

- All my colleagues from KCL CTR department for this wonderful journey during the years of my PhD. Special thanks to (alphabetically) Christoforos Vlachos, Ehsan Ghoreishi, Fahad Ausaf, Gao Zheng, Kostas Antonakoglou, Omar Al Kadri and Yaqub Alwan.

- The EU Multi-Partner Initial Training Network (MITN) Marie Curie project *CROSSFIRE* for giving me the opportunity to pursue this PhD, as well as all the project members for their wonderful assistance and collaboration.

- My parents, Stelios and Eirini, my sister Danai, and all my friends and family members for their ultimate love and support.

Dedication

I dedicate this work, with all my heart, to my loving parents to whom I owe everything, to all my teachers and to my dearest friends.

'The unexamined life is not worth living'

Socrates
Plato's Apology (38a5-6)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Next Generation (5G) wireless and mobile communication is above all expected to provide ultra-high data rates over wireless in the range of Gbps to fulfil the ever increasing user demand on data consumption. The main perhaps breakthrough of 5G networks will more importantly be about providing greater reliability and supporting Quality of Experience (QoE) in a personalized manner.

As the 5G era is approaching, it is expected for research and innovation to develop along both revolutionary as well as evolutionary paths. The stance to be taken hereafter is that an evolution in terms of physical layer enhancements is foreseen in order to provide increased data rates (increasing the overall capacity), whereas a revolutionary step is required in terms of network orchestration and management in order to provide consistency and efficient utilisation of the available resources at a minimum cost.

In the roadmap towards 5G networks, a forecasted evolution in terms of technologies is envisioned for supporting Gbps wireless transmission, whereas a revolution would be required from the current modus operandi in the ways network orchestration and resource management is performed in these complex, hierarchical, heterogeneous

and highly autonomous wireless networks.

Hence, even if wireless transmission throughput keeps increasing in the same trend as today, as it is expected, along with the steady increase of the hardware capabilities, a ground-breaking rethinking of the overall way networks are designed, managed and shared will essentially be required in order to create the new generation of wireless networks that will bring an actual innovation in the ways humanity and machines interconnects and communicates at scales ever imaginable.

1.1 Motivation and Objectives

The key challenge for mobile and wireless networks for the decade to come will undoubtedly be to support the anticipated thousand-fold mobile traffic increase, while at the same time efficiently support the ever increasing diverse set of requirements from different applications. With 4G technologies (LTE-A) only recently having been fully deployed (as of 2017), research and innovation efforts have fully commenced on the development of the next generation wireless systems. Generally referred to as 5G, it will need to encompass breakthrough technologies and architectures, which will be able to match the unprecedented rise in quantity and heterogeneity of wireless traffic. This has repercussions on both access and core networks.

The 3G technology will be remembered as the first mobile network which was largely IP enabled, i.e. packet-switching technologies have been designed into the network. That was important for the exponential uptake of data-driven applications, which eventually would drive the smart phone revolution. On the downside, 3G will be remembered as performing relatively poorly in the Radio Access Technology (RAT) and Radio Access Network (RAN) sides contrary to all promises made on the ease of using a single-frequency network.

A positive side-effect of the radio network management becoming so complex, however, was the emergence of self-organizing networking (SON) techniques which would pave the way for proper management of heterogeneous networks in 4G and now 5G networks. 3G was also the first system to introduce more advanced multi-antenna systems, such as space time coding, among others, and prove it would work in a commercial setting.

4G learned from the lessons of designing and rolling out 3G. Notably, the core was made lighter and flatter and was entirely packet-switched; the RAN enjoyed advanced concepts such as Coordinated Multipoint (CoMP) reception and carrier aggregation (CA); more SON features mainly related to RAN management appeared; and a much simpler yet powerful and scalable air interface was introduced in form of Orthogonal Frequency-Division Multiple Access (OFDMA). 4G engineers tried less to come even closer to the Shannon bounds, rather than building a cost efficient system able to meet capacity needs of the emerging smart-phone revolution. Techniques such as CoMP for example seem to be complex for efficient implementation in real networks.

Two major constituents in the design were the availability of more spectrum (leading to CA protocols) and denser networks (leading to the first highly heterogeneous cellular networks). On the other hand, 4G networks can be considered as "islands" where sharing and innovation on the control and management plane are very much limited since all functions are based on proprietary hardware and software. This is a significant limiting factor to ensure network sustainability and integration of various different heterogeneous networks.

1.2 Contribution

With above lessons in mind, the community is currently gauging the design requirements for 5G networks, where the emerging issue of 'softwarisation' of the end-to-end communication that includes both Software-defined Networking (SDN) and Network Function Virtualisation (NFV), as well as the main emerging mobility management techniques.

This work will explore the merits of virtualising the network, in the area of mobile communications. For this reason, firstly the focus is given on the study of the state-of-the-art mobility management techniques and, thus, a hybrid distributed mobility management scheme is proposed. Combining an efficient way to manage the mobility effect as well as virtualising the network, the next step is to explore the area of virtual network embedding.

A new mobility aware virtual network embedding algorithm is then proposed and evaluated against the state of the art. The rest of this work presents a deeper study of the virtual network embedding area with the proposal of a solution that focuses on minimising the latency for multiple tenants that share a physical infrastructure.

The last part takes into consideration the stochastic nature of the parameters that are considered in the mathematical formulations of the previous solutions and thus, it proposes a robust virtual network embedding solution that gives a flexible framework for multi-tenancy with adjusted robustness.

In conclusion, the main contribution of the outcome of this work can be summarised as the effort to bring together the area of mobile network optimisation and sharing along with the concept of network softwarisation, and all the merits that this can bring. This work explores the techniques that can introduce a more efficient and flexible sharing of the network by multiple tenants that additionally may nearly optimise the overall performance in terms of different metrics in scope. Those techniques

can be seen as network orchestration concepts for efficient sharing and operation of the available network resources that primarily are adapted to the case of mobile networks, taking into account the effect of the mobility of the users.

1.3 Publications

1. G. Chochlidakis and V. Friderikos, "Hybrid Distributed Mobility Management for Next- Generation Wireless Networks", in *IEEE International Conference on the Network of the Future (NoF14), Paris, France*, December 2014.

2. G. Chochlidakis and V. Friderikos, "Mobility Aware Virtual Network Embedding", in *IEEE International Conference on Communications - Communications QoS, Reliability and Modelling Symposium (ICC15 CQRM), London, United Kingdom*, June 2015 pp. 58655871.

3. G. Chochlidakis and V. Friderikos, "Robust Virtual Network Embedding for Mobile Networks", in *IEEE 26th International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): Mobile and Wireless Networks (IEEE PIMRC2015 Mobile & Wireless), Hong Kong, P.R. China*, August 2015, pp. 18671871.

4. G. Chochlidakis and V. Friderikos, "Low Latency Virtual Network Embedding for Mobile Networks", in *IEEE International Conference on Communications - Communications QoS, Reliability and Modelling Symposium (ICC16 CQRM), Kuala Lumpur, Malaysia*, May. 2016.

5. V. Friderikos, G. Chochlidakis , H. Aghvami and M. Dohler, "Challenges of 5G Networking in Access and Core Networks", in *IGI GLOBAL Handbook of Research on Redesigning the Future of Internet Architectures*, September 2015.

6. C. Vlachos, G. Chochlidakis, J. Heide, and V. Friderikos, "Network Coded Compression-based Caching for Device-to-Device Communications" in *IEEE European Conference on Networks and Communications (EuCNC)*, June 2016.

7. G. Chochlidakis and V. Friderikos, "Mobility Aware Virtual Network Embedding", *IEEE Transactions on Mobile Computing* , June 2016, DOI: 10.1109

\TMC.2016.2591525 , pp. 1343 - 1356.

Chapter 2

Background Theory

2.1 Towards 5G Core Network Revolution

In 5G networks, the design of the core network is of fundamental importance since the support of ultra-low latency applications and ubiquitous seamless mobility support require strong support from the core/backbone network. This includes dynamic application-aware per-flow and/or per-class routing decision making, advanced multi-homing approaches, and distributed mobility support under growing heterogeneity.

Indeed, the increased heterogeneity of wireless access and the need for high speed ubiquitous/seamless access to a variety of different Internet applications calls for efficient mobility management schemes at the IP layer. Furthermore, the introduction of programmability at the network level is changing the landscape of networking and even though such techniques have been mainly considered in data centres, extensions for wireless/mobile networks is becoming the next frontier, attracting significant research efforts worldwide.

Mobility management has been considered as a functionality to support vertical han-

dovers [1] [2] but there has been little effort to optimize its performance, especially for small cells, HetNets and split-architectures where the notion of cell is disappearing. Hence, efficient solutions for mobility aware network slicing/sharing and virtualisation are completely open areas of research. The aim is thus to shed further light on the trade-offs between host and session mobility, optimized routing in virtualised infrastructures and opportunistic adaptation to full or hybrid decentralisation of the mobility functionalities based on network and traffic conditions.

2.1.1 From Centralized to Distributed Mobility Management in 5G

Over the last few years, the number of the mobile Internet users has tended to increase steadily and the data traffic follows an exponential increase. In addition, according to predictions an explosive growth of data demand is going to take place over the next few years. However, due to the current flat rate model that has been in use in mobile services, mobile operators have been turned into 'dumb pipes' for the application providers, watching in this way the average revenue per user (ARPU) decreasing in a fast rate.

In order to increase the margin and increase their profit, mobile operators have had to find solutions and techniques in order to increase the utilisation of the limited physical resources as well as to make the traffic delivery more efficient. Among others, main emphasis has been given in IP mobility management as a way to improve the QoS for end-service and the total performance of the core and access network. Below, there is a description of the main all-IP mobility management schemes that have been proposed so far.

The Internet Engineering Task Force (IETF) standards organisation proposed Mobile IP in order to support IP hosts mobility. The main concept is to give to the

MN two different addresses: the home address (HoA), which is the fixed address to identify it and the care of address (CoA), which is used in order to track the MNs location (current subnet). In Mobile IPv4 (MIPv4) [3], a network entity called the home agent (HA) is responsible for mapping the HoA and the CoA and forwarding flows towards and from the MN. In particular, when a Correspondent Node (CN) is about to send packets to a MN, it will actually try to reach the home address of the MN. Then, locally this flow will be tunnelled through the HA to the MN or a foreign agent with direct link to the MN, by IP-in-IP encapsulation.

In Mobile IPv6 (MIPv6) [4], the mapping of the HoA and the CoA can be done, also, by each CN as long as it has stored to its cache the binding updates for the Mobile Nodes (MNs) with which they communicate. This enhanced feature of MIPv6 avoids the suboptimal (triangle) routing via the HA, because CNs can send packets directly to the MNs, using IPv6 routing header option, if they have recent entries for their statuses. In the case that a CN has no entry for an MN, the flow will be forwarded to the MN through the HA, but when the MN receives the encapsulated packets, it will send back to the HA and/or the CN a Binding Update (BU) about its current CoA, using either extension headers or piggybacking on data packets Fig. 2.2a.

In cases with high mobility, the fast changes of the point of attachment of MNs can decrease the QoS, mainly due to signaling overhead and delay. For this reason, IETF proposed Fast Handovers for Mobile IPv6 [5], which manages to decrease the triggering time of the handover procedure. More specifically, when a MN detects that it is about to migrate to a new Access Router (AR) domain, as long as it has cached information for this domain, it prepares the binding update signalling and the flow forwarding in order to minimize the handover delay. In this way, the MN manages to move, while having an open session, within different AR domains, without being affected by the handover procedure.

Hierarchical Mobile IPv6 (HMIPv6) [6], is another mobility support scheme pro-

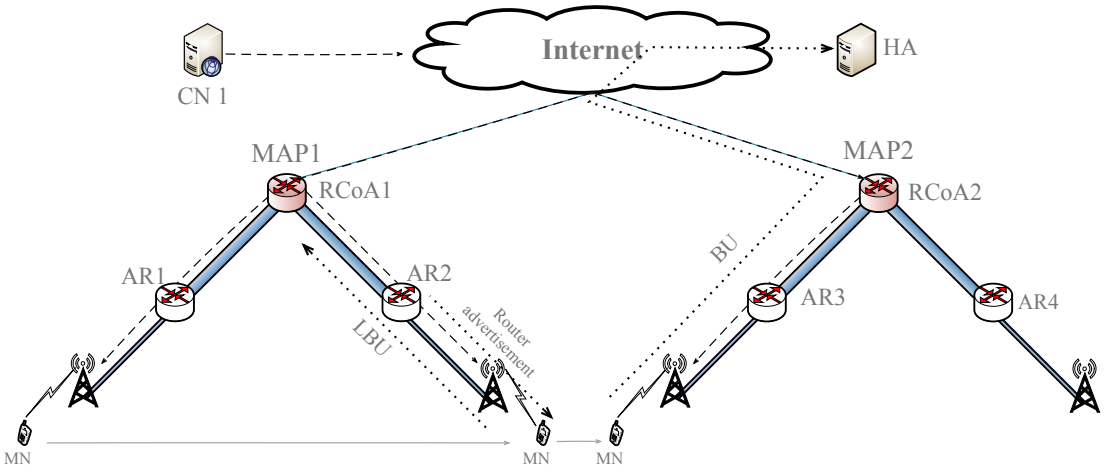

Figure 2.1: Mobility Management in Hierarchical MIPv6.

posed by IETF as an extension to MIPv6, mainly suitable for scenarios where a lot of handovers take place and the MNs are located far away from their home domains. According to HMIPv6 Fig. 2.1, the mobility of the users is handled locally by network entities called Mobility Anchor Points (MAPs), which serve certain AR domains and act like local HAs. In this way, most of the handovers are handled locally in an efficient and fast way. On the other hand, when an MN migrates to a different MAP domain, MIPv6 handles the mobility support as it was described above. More precisely, when a MN is located inside a MAP region, it obtains a Regional Care of Address (RCoA) and the HA and the CNs are update their entry. From now on, every flow towards the MN will be routed through the MAP, which serves this area using the on-Link Care of Address (LCoA). While the MN moves within the same MAP area, its MAP is responsible of tunnelling the flow to the AR which serves the MN, avoiding in this way high signalling and making the handover procedure fast. In the more rare cases of global handovers (between different MAP domains), the BU has to be sent back to the HA and/or CN of the MN in order to update the RCoA and to forward the flow through the new MAP.

In order to decrease the signaling overhead and to avoid the requirement of mobility software at the MN, Proxy Mobile IPv6 (PMIPv6) [7] was proposed. PMIPv6 Fig. 2.1b is a network-based mobility solution, where in contrast to the previous mo-

bility management schemes there is no need to install any-mobility related software on the MN, while the mobility management is assigned to specific network entities. More precisely, Mobility Access Gateway (MAG) is a network entity, which runs on the AR, and it is responsible for tracking MNs location and creating a tunnel with the other basic network entity, Local Mobility Anchor (LMA). LMA, which is an enhanced version of MIPv4s HA, is mainly responsible for ensuring the reachability of the MNs address, while it moves within a PMIPv6 domain. When a MN is located inside a PMIPv6 domain, MAG obtains MNs profile and then it sends a Proxy Binding Update (PBU) to LMA, which after an authentication process sends back a Proxy Binding Acknowledgment (PBA) and sets up a route for the MNs home network prefix using the tunnel with the MAG. After receiving the PBA, MAG is able to emulate the MNs home network by sending a Router Advertisement (RA) message to MN in order for it to configure its home prefix accordingly. At this moment, all data will be forwarded via this MAG-LMA tunnel, saving bandwidth from the MN by excluding it from mobility-related signalling (so, the MN is unaware of the mobility support procedure).

All of the above mentioned mobility management schemes share the common feature that they are strongly centralized and for this reason it will be refered as Centralized Mobility Management (CMM) schemes. As summarized in [8], the main disadvantage of CMM is the suboptimal or triangle routing, which means that all the flows have to be routed through potentially unnecessarily longer paths via a centralized mobility anchor. In this way, mobility anchors become points of congestion causing high delay and degradation of the QoS. In addition, the centralized anchors can also become single points of failure affecting in a greater degree the robustness of the whole network. Moreover, CMM schemes lack dynamic mobility support as well as scalability because of the fact that they turn out to be a bottleneck for the whole network, especially when MNs tend to increase. For this reason, core networks are dimensioned to support peak data traffic. Finally, in CMM there is a waste of

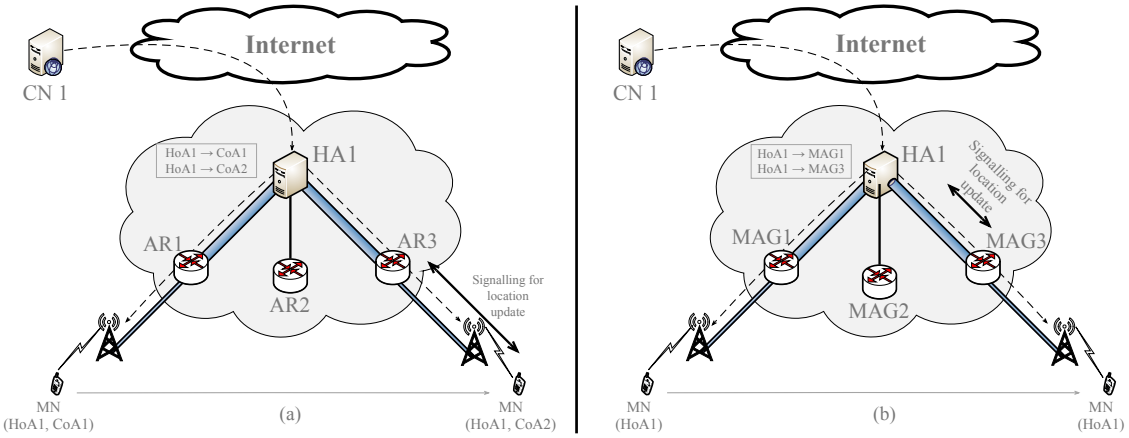

Figure 2.2: Mobility Management in (a) MIPv6 and in (b) PMIPv6

network resources because even in cases where seamless handover is not important (Internet browsing), there is tracking of the MN and 'blind' IP mobility support.

All these drawbacks of CMM schemes motivated IETFs working group to propose Distributed Mobility Management (DMM) scheme [9], a mobility management scheme, where the mobility function is distributed at the edge of the core network. In [10], the main requirements that DMM had to meet, and the motivations were defined. According to [11] the main classes of solutions for distribution of the mobility management are the client, the network and the routing based solutions.

In the client-based approach, the anchoring is distributed at the edge of the core network and the MN uses additional IP address at each visited AR. In this way, for each new flow the MN will use the locally anchored address, but each time it migrates to another AR it keeps reachability for the previous IP address of the domain where the flow was set up. In order to do so, the MN has to bind the addresses of the active sessions with the local address of its current domain allowing the data to be forwarded from the previous HA to the new one. This approach needs software intelligence from the side of the MN because it has to update the entries of the current and previous addresses, use the right one to start new sessions, track the addresses which need mobility support and take care of the binding, signalling and tunnelling.

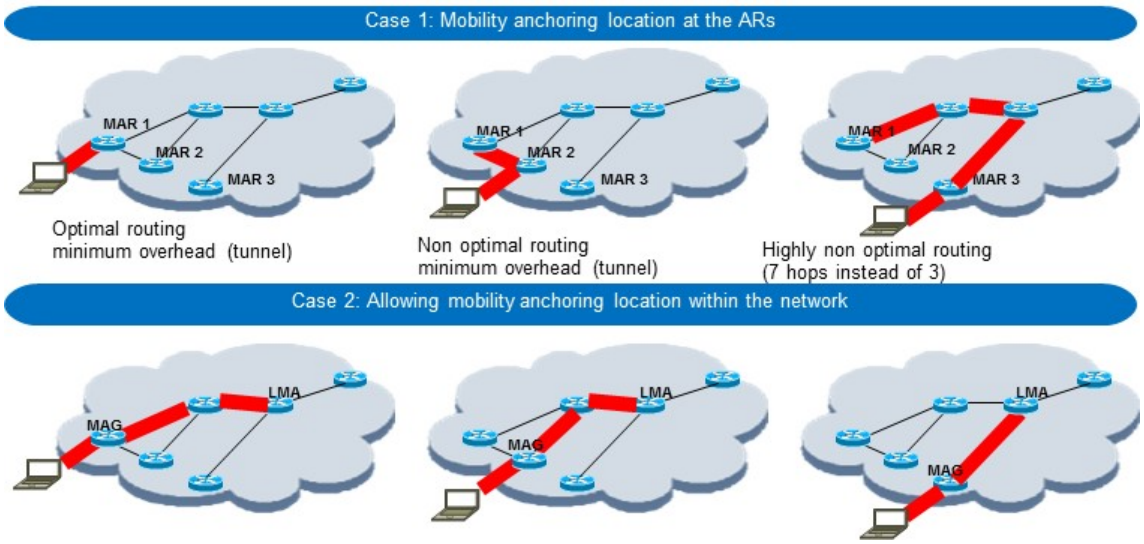

Figure 2.3: Comparison of routing and tunneling between DMM (Case 1) and CMM (Case 2) mobility support.

Regarding the network-based approach, there are two subclasses: the fully distributed and the partially distributed solution, depending on whether or not the control plane is decoupled from the data plane. According to the former one, both the data and the control plane are distributed at the edge of the network and the network entities (LMA and MAG in PMIPv6 approach) are responsible for data forwarding and mobility related signalling. In the partially distributed solution only the data plane is distributed while the control plane is managed by network entities (e.g. 3GPPs MME), which are located hierarchically higher across the network.

In Fig 2.3 the two different schemes are visualized by depicting a comparison of routing and tunnelling between DMM (shown as Case 1) and CMM (shown as Case 2). For the case of DMM when the MN is connected to MAR 1 the routing path is optimal and the tunnelling cost is minimal since it only take place between the MAR 1 and the AR where the MN is connected to. On the other hand, in the case of a handover the routing path becomes sub-optimal since the flow has to be tunnelled to the new MAR (MAR 2) via MAR 1. As can be seen in the figure for more handovers the sub-optimality of the routing path can further deteriorate and in that specific setting the number of hops, when connected to MAR 3, will be 7 instead of the optimal path that requires 3 hops. On the other hand, for the

case of CMM (in the figure PMIPv6 is considered) the routing cost, even in the case of multiple handovers, can remain close to optimal (depending on the location of the LMA) but there is a significant tunnelling cost since all flows need to be encapsulated between the LMA towards the MN. Fig 2.4 focuses on the issue of tunnelling cost between DMM and CMM approaches when taking into account cell residence times and session duration times. Both of these can be considered as independent random variables that have a direct effect on the performance of the different mobility support schemes. The figure depicts the cases where during session duration time there are zero (P0) up to three (P3) handovers. As can be seen in the figure CMM always results in tunnelling overhead whereas in the case of DMM tunnelling overhead only take place when there is a handover. Therefore, in the cases where the session duration time is significantly less than the cell residence time DMM can entail significant less overhead compared to CMM schemes.

Therefore, it becomes evident that the performance of DMM schemes are very much topology dependent since there can be instances where a more centralized approach might perform better especially in terms of routing. These design trade-offs need to be carefully taken into account for supporting seamless mobility under DMM solutions.

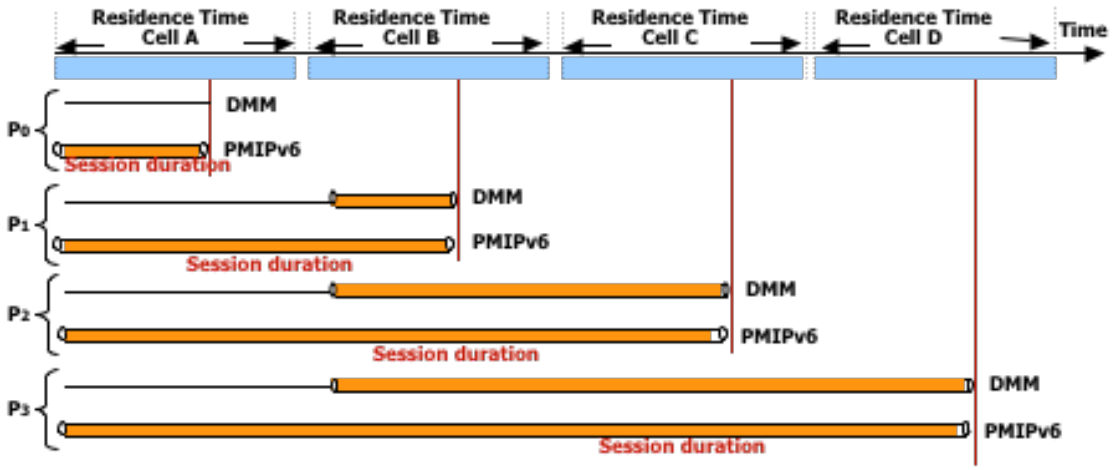

Figure 2.4: Tunneling cost of DMM and CMM (PMIPv6) for different instances of cell residence time and session duration.

Finally, in the routing-based approach [12], the routers are connected in mesh-like structure and not in a hierarchical one (i.e. core, aggregation and access). The main difference is that during a handover, the flow will not be tunneled from the former AR to the new AR, but each time an optimal route will be deployed via routing updates. However, this implementation has some disadvantages comparing to the other two. The main are the limitations by the routing convergence of the handover delay and the scalability issues due to potential storms of routing updates.

In current 3GPP Release 12, although DMM is not yet deployed, there have already been some remarkable efforts in terms of traffic relieving and dynamic mobility management. Two of the most remarkable of them are Selected IP Traffic Offloading (SIPTO) and Local IP Address (LIPA) (3GPP TS 23.401, 2011). SIPTOs concept is to allow traffic offloading at a network node close to the UEs point of attachment to the access network by selecting a set of gateways, which are located geographically close to it. LIPA allows a UE to connect to other IP-capable entities in its local network without the involvement of the rest of the network.

Several other works towards the improvement of current 3GPP mobility management have been proposed the last few years. In [13], [14] there is a proposal for distribution of the data plane in 3GPP Evolved Packet Core (EPC) networks. Particularly, the author studies the potential gain from the relocation of GWs in order to optimize the routing of the flows to the users and to get rid of non-optimal routing. The authors in [15] motivated by the disadvantages of centralized mobility management and by the lack of mobility support of SIPTO and LIPA, they present an evolved 3GPP architecture that supports fully distributed mobility management, which is also compatible with current solutions. Their approach introduces a new mobility support entity, the distributed gateway, which is located at the edge of the network and it is responsible of supporting the mobility and forwarding the data during handovers.

In [16], the authors propose efficient local mobility management schemes for 3GPP LTE-A networks, where instead of dealing with path switching at the core network for each handover they introduce the idea of local traffic forwarding chain construction in order to use the existing Internet backhaul and the local path between the local anchor femtocell and the next femtocell for active sessions. The results show that the proposed schemes manage to reduce the signalling cost and relieve the processing burden of mobile core networks, where femtocells are deployed. They can also enable fast session recovery after a handover allowing in this way to adapt to the self-deployment nature of the femtocells.

Finally, the authors in [17], firstly provide the main aspects and challenges of mobility management in mobile networks with deployment of femtocells classified according to the cell identification, the access control, the cell search and selection, and the handover decision and execution. Then, they list the main algorithms and techniques from the literature, according to the class they belong to, mainly focusing on the handover decision phase.

In conclusion, future mobile networks, in order to address the limitation of physical resources and to increase the capacity to deal with high future demand, are expected to make extended use of femtocells. One of the important issues of the deployment of smaller cells is the increased number of handovers and the need for efficient and reliable mobility support. For this reason, a distributed mobility management approach is strongly believed that is going to be considered for deployment from next generation networks. As stated above, such deployment will allow the mobile operators to move from centralized architectures to more flexible distributed architectures, where the mobile function will move towards the edge of the network as well as anywhere across the network, as it will be extensively shown in the next chapter, depending on desired QoS, types of delivered services and overall performance requirements.

2.2 Network Virtualisation via SDN, NFV & Virtual Network Embedding

Broadly speaking, virtualisation is the process of creating virtual entities, which have the same characteristics with physical resources and emulate their function. This is by far not a new concept since the idea of virtualisation was firstly introduced in computer systems in the early 60s through multi-programming techniques. Since then, and under the same concept, different resources, like memories, CPUs etc. have been virtualised and used in order to increase the utilisation, the flexibility and the total efficiency of computer systems.

2.2.1 Network virtualisation in mobile networks

Currently, virtualisation can be classified into three main categories, which are by themselves areas that attracted significant research: storage virtualisation, which refers to creation of virtual memories using resources from a single or multiple physical memories, server virtualisation, where different virtual machines can be created from one or more physical machines and network virtualisation, the technique of forming virtual networks (VNs) by combining physical resources. An example of the latter is shown in Fig 2.5 below, where two different clients negotiate with an ISP (physical infrastructure) for the creation of two virtual networks. Assuming an SDN enabled infrastructure, the mobile operator or ISP needs to define the topology (this is called virtual network embedding), which must have the requirements as specified by the client.

Enablers towards this direction will be both software-defined networking techniques in which control and forwarding planes are logically decoupled as well as Network Function virtualisation (NFV) where different mobile network functions can be de-

coupled from a specific hardware which in current networks are mainly proprietary (ETSI, 2013). Therefore, broadly speaking, NFV can be considered as the idea of implementing network functions from software-specific hardware to software-based applications (hardware agnostic) on widely used off-the-shelf systems. Still it can be deemed as rather early to foresee the actual direction in terms of impact that SDN and NFV will have on the infrastructure and operation of the network even though key players have already intensified their efforts in that scope area. It is worth pointing out that there are more than 200 member companies including operators within the ETSI NFV initiative. For example the Internet Multimedia Subsystem (IMS) might quickly move from an all-in-one black box to hardware agnostic virtualised functions which might also be implemented in the cloud. Despite that, network virtualisation is emerging as an important architectural item for emerging wireless network, hence it will be detailed furthermore hereafter.

As already eluded above, over the last few years, mobile operators have to deal with the problem of exponential traffic demand and to come up with solutions in order to improve their QoS and their overall sustainability. Taking into consideration that physical resources are limited and expensive, the increase of their utilisation is a high-priority problem. For this reason, it comes as no surprise that network virtualisation becomes more and more a prominent solution [18].

The main advantages of network virtualisation are the fact that it increases the flexibility and the utilisation of the resources and thus it is energy efficient. Thanks to network virtualisation, it is much easier for a mobile operator to redesign dynamically the network by adding and removing network elements as well as to recover much faster and reliably from network failures. In addition, network virtualisation gives the chance to more operators to enter the market by leasing resources and setting up virtual networks, without owning large and expensive infrastructures. This leads to a more competitive market with better quality of service and better choices

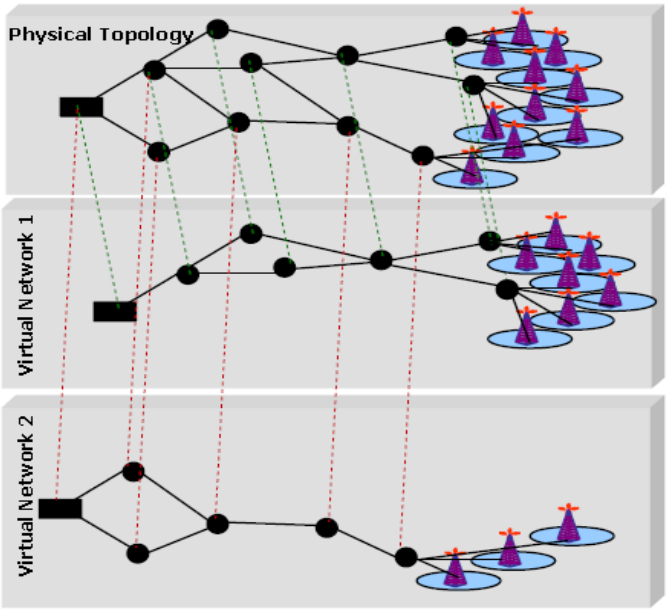

Figure 2.5: A scenario of physical mobile wireless network infrastructure (belonging for example to an ISP) and the creation of two Virtual Networks representing two different requests by different clients.

for the end-user.

The main plan of network virtualisation in mobile networks consists of three basic roles. Firstly, the infrastructure providers own, manage and virtualise the physical resources. The virtual network providers act like brokers between the infrastructure providers and the network operators by combining the physical resources in order to form the virtual networks. Finally, the virtual network operators are responsible of running the virtual networks and provide service to the end-users.

The virtualisation techniques in current mobile networks can be classified into virtualisation of physical infrastructure (i.e. nodes, links etc.) and virtualisation of the air-interface (access control and spectrum). Regarding spectrum virtualisation, the main problem that has to be addressed is how radio physical resources can be partitioned and shared among different network providers. In [19] the authors provide a preview on the state-of-the-art on RAN sharing techniques and then they present their proposal, Network Virtualisation Substrate (NVS), a slice scheduler and a flow scheduling framework. Their approach aims to provide isolation between shared resources, customisation in order to achieve dynamic allocation and

improved utilisation of the resources. In [20] the authors present CellSlice, which is an innovative system for slicing wireless physical resources, enabling efficient RAN sharing. CellSlice is a gateway-level solution, which remotely controls the scheduling decisions without the modification of the BS schedulers, making it easier to be adopted. The results show that the performance of CellSlice is close to NVSs, being access-technology independent, hence can be considered suitable for various networks. The authors in [21] perform a study and evaluation of different sharing options, from simple to more complex methods. Then they compare, via a simulation testbed, capacity sharing, spectrum sharing and virtualised spectrum sharing under different scenarios in order to conclude that capacity sharing, a generalisation of traditional roaming, is the best performing as well as the simplest option while spectrum sharing is least effective. In [22], the authors propose a partial resource reservation, which is a flexible active RAN sharing technique allowing each operator to have a minimum percentage of the physical resources and also having access to a common resource pool with a first-come-first-serve scheme. Their proposal, adapted for LTE networks, manages to address both the scheduler and admission control aspects. After system-level simulation, they show that, in comparison to full reservation schemes, their proposal can dynamically and efficiently allocate the physical resources among different operators according to the demand and traffic priorities and in this way to increase the spectrum utilisation.

Regarding the virtualisation of the core network, a major problem that attracted significant research attention over the last few years is finding efficient (or optimal if possible in some instances) ways to map a virtual network over a physical infrastructure. Virtual network embedding deals with the allocation of virtual resources both in nodes and links and it can be divided in virtual node mapping and virtual link mapping. The authors in [23] provide the landscape of the research work that has been done so far on the area of algorithms and techniques, which address the virtual network embedding problem. Then, they create a taxonomy of major algorithmic

efforts which exist in the literature. One clear message which is becoming now apparent is that more effort should be placed on the design of distributed virtual network embedding algorithms, since centralized ones have not only the problem of single-point of failure but they might increase overall signalling load from the access to the core network in areas which are particularly prone to congestion. A distributed approach might be more effective to "hide" their operation across the network since decision making might take place locally decreasing in that sense overall signalling load in the network. In addition, energy efficiency and security are aspects that should be kept in mind in future research.

This is an important issue since with a greater degree of programmability in the network the vulnerabilities will only tend to increase in the network, hence special attention should be placed in protecting a software-defined network from sophisticated malicious attacks. Network control and decision making in 5G networks should be able to handle a wide heterogeneity of requirements from current, emerging and future applications. In terms of only delay for example, these should range from millisecond (tactile Internet) to hundreds of milliseconds (mobility, resource allocation) and hours (network slicing, topology configuration). Hence, in this emerging environment orchestration of the control plane is becoming significantly important due to the high heterogeneity of the time scales in terms of decision making. Therefore, it might needs to be devised dedicated scheduling and control plane optimisation algorithms to enable efficient "smart" decision making using an ever increasing information set from the network hence creating networks that are able to adapt efficiently to internal (i.e., failures) and external (i.e., traffic, mobility) events.

2.2.2 Software Defined Networking

The need for scalable, dynamic and resource efficient next generation networks has made Software Defined Networking (SDN) [24] the dominant architecture that ex-

plicitly can exploit the benefits that network virtualisation offers. SDN decouples the control plane from the data forwarding plane and abstracts the network function. In this way, it offers the freedom to dynamically configure the forwarding logic.

The network intelligence is logically centralized in a software-based SDN controller (or more, depending on the scale of the network). The SDN controller has full knowledge of the state of the physical infrastructure and its responsibility is to updated timely network policies according to flow activities. Moreover, application programming interfaces (APIs) are supported to link the application and the control layer in order to facilitate the network management (Fig. 2.6). Acceleration on developments of the SDN frontier is expected to increase as various collective efforts mature such as the Open Networking Foundation (ONF) [1], the OpenDaylight [2] and the OpenCompute [3].

Regarding the enabling of SDN, the OpenFlow [25] protocol can be used as a standard communications interface between the control layer and the forwarding layers. In particular, it provides access to the forwarding plane of physical and virtual network devices (i.e. switches, routers). The programmable OpenFlow routers and switches follow the policies that are determined by the SDN/OpenFlow controller. Within this remit some other protocols could be utilized such as for example *NETCONF*, which provides mechanisms for installing, manipulating and deleting set of defined configurations at SDN enabled network devices [26].

Flowvisor [27] is an innovative network virtualisation approach that acts like a hypervisor between software and hardware on a PC using OpenFlow as a hardware abstraction layer to sit logically between control and forwarding paths on a network device. The same researchers who proposed FlowVisor in [28] describe a way to build a testbed that is embedded in the network and evaluates its performance. The

[1] www.opennetworking.org
[2] www.opendaylight.org
[3] www.opencompute.org

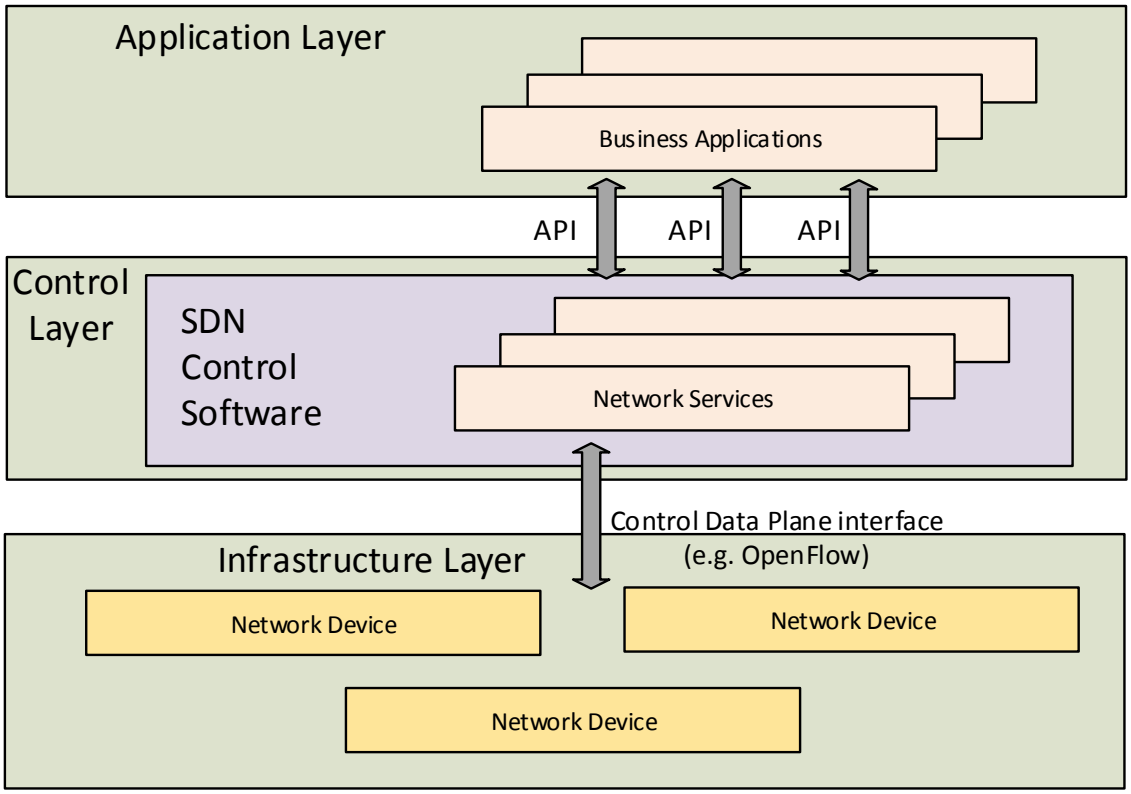

Figure 2.6: SDN architecture overview

main advantages of using FlowVisor in a sharing infrastructure (like the one in the scenario that will be investigated) is that it provides transparency, strong isolation and dynamic slice definition.

Another interesting SDN approach that can also be implemented in the hereafter presented scenario, is given in [29]. The authors believe that slice isolation should be provided at the language level and to this end, they propose a slice abstraction that facilitates the isolation between network programs. They firstly define a simple programming abstraction for defining slices and then they describe algorithms for compiling slices to OpenFlow switches. Then they present the performance of their proposal. For a more detailed treatment on current and emerging SDN approaches, a notable work which surveys the state-of-the-art in traffic engineering for SDNs is given in [30].

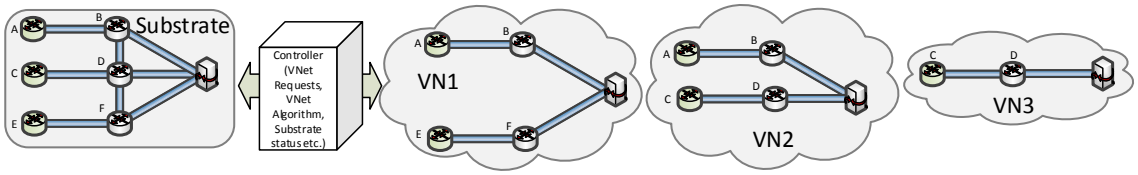

Figure 2.7: Forming virtual networks on the top of a core network infrastructure under a predefined strategy

2.2.3 Network Function Virtualisation

Network Function Virtualisation (NFV) [31] proposes the installation of the mobile network functions as software instances on data-centres, network nodes and on the end users' premises. This aims to limit the hardware dependence, decrease the maintenance costs, energy [32] and capital investment and increase the network scalability and robustness. NFV is a highly promising solution for any data plane packet processing and control plane function in wired or wireless mobile network infrastructures.

A combination of both SDN and NFV implementation can be seen in [33] where the authors argue that depending on network performance in terms of load and delay, there can be areas of the network topology where NFV deployment might be more beneficial and other areas where SDN deployment with decoupled data and control plane performs better. Then a function placement problem is proposed that minimizes the transport network load overhead with respect to different parameters.

NFV bears the potential of flexibly locating network functionalities in a vendor agnostic networking hardware and also there is the potential of combining network functions in on-demand adaptive way, hence creating a link with SDN. Although NFV bears plenty of advantages and may be a step in the right direction, it might be insufficient [34]. The authors believe that it is very important that a programmable and dynamic realisation of per-flow control that manages flows across different functions chains should be adopted for next-generation networks. Such an approach can allow service providers to quickly create and deploy new revenue-generating services.

2.3 Publications

1. V. Friderikos, G. Chochlidakis , H. Aghvami and M. Dohler, "Challenges of 5G Networking in Access and Core Networks", in *IGI GLOBAL Handbook of Research on Redesigning the Future of Internet Architectures*, Sep. 2015.

Chapter 3

Hybrid DMM for Next-Generation Wireless Networks

One of key challenges for emerging and future wireless networks will be the support of seamless distributed IP mobility management to support a plethora of different applications. In this chapter, an optimisation problem for provisioning efficient deployment of centralized Mobility Agents (MAs) is formulated, as well as a realistic model is developed for the Distributed Mobility Management (DMM) scheme. These are subsequently compared in terms of various key characteristics, such as routing cost, delay and topology dependence. Then, an innovative *Hybrid Distributed Mobility Management* (HDMM) scheme is presented, that provides improved network performance in terms of handover support for delay sensitive flows, compared to fully DMM schemes, in which their performance can be strongly topology-dependent. The proposed scheme combines the centralized and distributed mobility support, depending on the network's topology characteristics. A wide set of numerical investigations reveal the advantages of the DMM scheme over the centralized scheme for different network cases and detail the reasons why future networks tend to decentralize mobility management functionalities. Simulation results, also, show that

the proposed HDMM scheme can significantly improve the network's performance and the achieved QoS of the end-users, allowing seamless mobility support for delay intolerant over-the-top services (e.g. VoIP).

3.1 Introduction

As the number of mobile Internet users tend to increase steadily and a dramatic growth of data traffic demand is expected to continue taking place into the next few years [35], all-IP based networks and IP based mobility support are the de-facto solutions, as they can offer to the users seamless mobility between heterogeneous wireless networks, without interrupting their service. In order to address the mobility support problem, many different mobility management schemes have been proposed so far, being mainly divided into two different categories: host-based and network-based mobility support solutions. In host-based, the Mobile Nodes (MNs) actively take part in the mobility management procedure, while in the network-based, certain network elements are deployed which are entirely responsible for the mobility support procedure.

As already has been detailed in-depth in the previous chapter, the Internet Engineering Task Force (IETF) standards organisation proposed Mobile IPv4 (MIPv4) [3] & Mobile IPv6 (MIPv6) [4] as host-based mobility support solutions for all-IP networks, where two IP addresses are assigned to the MN: the Home Address (HoA), which is the fixed address to identify the MN and the Care of Address (CoA), which indicates the current position (IP subnet) of the MN. The network entity called Home Agent (HA) handles the mapping of HoA and CoA and it is the node, where all the traffic towards and from the MN flows through.

Hierarchical Mobile IPv6 (HMIPv6) [6] is another mobility support solution proposed by IETF, as an extension to MIPv6, in order to improve network performance,

in cases where frequent handovers take place and there is a long distance between the MNs and their home domains. HMIPv6 introduces the Mobility Anchor Point (MAP), which handles the mobility inside local domains, acting like a local HA, while mobility between different domains is handled by MIPv6.

In order to eliminate the signalling overhead and avoid the client's IP mobility software installation, which is used in the host-based mobility management schemes, a network-based protocol for the mobility support has been proposed by IETF called Proxy Mobile IPv6 (PMIPv6) [7]. PMIPv6 uses network entities in order to handle the mobility, on behalf of the MN and, in this way, it overcomes the need to add any software at the mobile users, excluding them from participation in any mobility-related signalling. When a MN is located within a PMIPv6 domain, the serving network gives it a unique home network prefix, although it has no awareness of this procedure, considering the PMIPv6 domain as the home network. Then all the traffic towards the MN is tunnelled from the Local Mobility Anchor (LMA), an enhanced version of the HA in MIPv6, to the Mobility Access Gateway (MAG), which runs on the Access Router (AR), detecting the mobility of the MN and triggering the mobility-related signalling with the LMA.

However, the aforementioned mobility management solutions are strongly centralized, leading to some potential disadvantages that have been well documented in the literature, which might impact the overall performance of network. Firstly, there might be suboptimal routing of the flows, which means that in order for the path to include a mobility anchor, flows might follow unnecessarily longer paths. In addition, the anchor points might, also, become points of congestion, which can lead to higher latency, with detrimental effects to the Quality of Service (QoS). Moreover, when mobility anchoring is taking place within the network, there are issues with network robustness, since the failure of certain anchor points can lead to losing mobility support for a high number of MNs

For these reasons, IETF's working group proposed Distributed Mobility Management (DMM) [9], a decentralized solution with specific requirements [10], where the main concept is the distribution of the mobility anchors by locating them at the ARs of the network. In this way, each flow is being optimally routed, while the handovers are handled by tunnelling the flow to the next AR. Such a solution would be especially attractive for cases, where the call session time is less than the cell residence time, resulting in a very small number of handovers. But this might not be always the case, especially with the current trend of reducing cell radius in order to increase the capacity and address the problem of increased data demand and limited available bandwidth.

In this chapter, an optimisation framework for the optimal number, location and selection of the Mobility Agents (MAs) across an access/core network is presented, where a centralized mobility management scheme is applied, by formulating it into a mathematical programming problem. Then, a model of the operation of the DMM scheme is developed and its performance is evaluated by comparing it with the derived optimized centralized mobility support scheme. Based on the above mentioned comparison, Hybrid-DMM scheme is proposed, where delay-sensitive flows are handled by mobility anchors within the network, so that in case of handovers their performance is not deteriorate, compared to a pure DMM support, where the handover delay might be highly dependent on network topology. The main concept is that, when the delay for some specific flows exceeds a threshold, these flows anchor to mobility agents, which are placed on hierarchically higher nodes across the network. Finally, the performance of the proposed scheme is compared with the so far presented schemes proposed by IETF, and the results show that there is a significant improvement gain on the achieved QoS for the end-users.

3.2 Selected Related Works

In this section, the main related to this chapter works from the literature (some of them have been mentioned previously in Chapter 2) are presented.

The authors in [36] present a framework on the optimal micro-mobility management in broadband access networks. Firstly, they analyse the impact of the MAs's location within the access network on the total routing and mobility overhead cost. Then, they formulate the problem of the optimal number and location of the MAs and, based on the simulations results, they achieve a significant improvement in the overall network performance. However, in their analysis, the flows are taken into account as total demands by the ARs and not individual flows and, also, the only possible handovers occurring are single-hop handovers.

In [11] Zuniga et al. give an extensive description of the DMM's framework, as it has been modified by the IETF and the 3GPP so far. First, the exact motivation of DMM is explained by listing the current deployed mobility management schemes' main drawbacks. Then, there is a review of the solutions for distribution of the mobility management that IETF and 3GPP standards organisations have proposed. The authors conclude by presenting some possible solutions for the evolution of the 3GPP's Evolved Packet Core (EPC) and they end up pointing out that DMM is a suitable solution for the future mobile networks.

The authors in [37] propose a network-based DMM scheme, where the logical function of the LMA splits and the mobility routing is located jointly at each AR with a mobility client function. The main objective of their work is to optimize the routing of the handover flows as well as the new flows, in order to improve the overall network performance. The results from their simulations confirm that their proposed scheme achieves better packet-delivery cost, tunnelling cost and total cost than previously proposed schemes (D-PMIP), but has increased signalling cost.

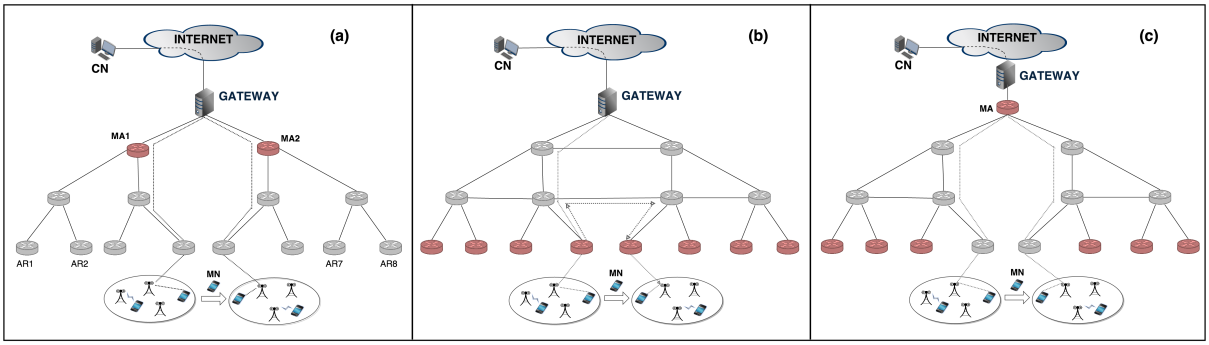

Figure 3.1: (a) Centralized scheme, (b) DMM scheme and (c) HDMM scheme

In [38], T.X. Do and Y. Kim focus on vehicular scenarios and flat architectures and they propose *D-NEMO*, a distributed mobility management scheme, where all of the ARs have the mobility management functions of mobility anchors and PMIPv6's MAG and where a proxy router is responsible for handling the registration of the MNs. The numerical analysis shows that the proposed scheme achieves better performance in terms of handover latency, in comparison to existing mobility management schemes.

Li Yi et al. in [39] present a study on the Internet flows duration, extracting some very useful results, based on real data and then they show how the duration of flows can affect the performance of the DMM scheme. Afterwards, by highlighting that only 3% of the Internet flows have a very long duration, they perform a comparison between the performance of the centralized and the distributed mobility management scheme. By this comparison they concluded that in general the DMM can be more efficient way to handle localized mobility. They, also, recommended that if the HoA is classified according to the application type, then the few flows with very long session durations could be handled by a centralized mobility management scheme.

3.3 System Model

3.3.1 Centralized mobility management scheme

For the first part of the numerical investigations, as presented hereafter, a model of an access/core wireless network is created, and the focus is on utilizing mobility management, where anchoring is taking place within the network, i.e. centralized schemes. The aim is to determine the optimal number, location and selection by the ARs of the MAs in the network. The type of networks, which are used, closely resemble wireless networks. For this reason, random planar tree-like graphs, varying from highly interconnected to sparse structures, are implemented, in order to test the topology dependency. The main goal is to simulate, as realistically as possible, the existent core networks of cellular networks. The network in Fig. 3.1a is an example of a binary tree (the simulations' worst-case topology), while the network in Fig. 3.1b is a tree-like network with dense interconnections between the intermediate the nodes. As the number of the interconnections increases, the routing options for the flows increase as well, offering a better potential routing cost.

A mathematical programming based solution approach is adopted for optimizing the process of mobility anchoring in wireless networks. To this end, the network can be modelled as a directed graph $G = (V, E)$, where V represents the set of nodes and E is the set of links. Let $F : E \rightarrow \mathbb{N}$ be a function defining the cost of each link and $c : E \rightarrow \mathbb{N}$ be a function, which defines the capacity of each link. They are considered K different accepted flows in the network. Without loss of generality, the data traffic originates from the source node *g=1* and flows through the MAs which serve the destination ARs. Each flow $k \in K$ can be expressed as a demand d_k which originates from the source node towards one particular AR of the set T of ARs (the more realistic networking case scenario is assumed, where flows are unsplittable). Let q denote the number of flows per AR. Every link between

two routers has a constant weight, so different paths across the network have a different impact in terms of delay or routing cost. Since all of the traffic is routed through the MAs, it is obvious that the number and the location of them will affect the total cost dramatically. More specifically, the MAs can be nodes of congestion leading to performance deterioration in terms of QoS in the network and also they severely affect the performance of handovers in terms of latency. Let C_m represent the capacity of the node, where MA m is located.

In this scenario, the shortest paths are pre-calculated from the source node $g = 1$ through every possible MA m towards an AR j, which is the destination node for the flow k. Let P express the set of those paths. Considering that each commodity can use only one path, let $p_{km} \in P$ be the path for the k_{th} flow, calculated from the source node to its destination AR j through the MA m. In order to express the total cost which is produced by the data traffic, it is considered that the path p_{km} from the source node g to an AR j consists of two parts: the path from the source node to the MA m and the path from the MA to the destination AR j, where an overhead is added (expressed as the percentage o), in order to represent the encapsulation.

Taking into consideration that each link has a specific cost, the routing cost for each flow is proportional to the total number of hops. It is obvious that as the number of deployed MAs (L) is higher, there are more available paths and the total routing cost can be potentially lower. In addition, their location plays a significant role in each different case, depending on the given demand distribution at the ARs.

In order to capture the mobility of the users, a handover matrix H is considered, where $h_{kj} \in (0, 1)$ is the probability that flow k moves to the AR j (with $h_{kk} = 0$.

Regarding the assumption of the mobility as a concept throughout this work, this is expressed by a set of probabilities of the flows to be switched to a different domain due to an occurring handover. Hence, this could be translated to the proportion of the sessions that are handled by handovers due to users mobility and varies

depending on the demographics, the geography of the served area, the structure of the covered area and other parameters. The effect on the access network is out of scope of this work since it is only studied how the routing of the active flows can be affected by the probability of it to be forwarded to a new domain.

In this scenario, and without loss of generality, one and two-hop handovers can occur, the handover matrix w is defined, where $w_{kjl} \in (0, 1)$ is the probability that the flow k moves to AR l through the intermediate AR j. Note that in the following formulation, all of the flows with the same destination AR j are anchored to the same MA. Also, when referring to the set of ARs T, without loss of generality, this refers to the set of destination nodes which is created by taking a single flow for each ARs (e.g. $T = \{k_i, k_j, ...k_n\}$ where k_i's destination is AR_1, k_j's is AR_2 etc.).

First, the following decision variables are defined:

$$M_m = \begin{cases} 1, \text{ if an MA is located at node } m \\ 0, \text{ otherwise} \end{cases} \tag{3.1}$$

$$x_{km} = \begin{cases} 1, \text{ if flow } k \text{ is anchored to the MA } m \\ 0, \text{ otherwise} \end{cases} \tag{3.2}$$

$$G_{kjm} = \begin{cases} 1, \text{ if flow } k \text{ and AR } j \text{ are supported by MA } m \\ 0, \text{ otherwise} \end{cases} \tag{3.3}$$

The total routing cost Ξ can be written as follows:

$$\Xi = \sum_{k \in K} \sum_{m \in J} d_k p_{km} x_{km} \tag{3.4}$$

When a flow k moves to an AR j, then the flow is routed through the MA that supports the flows of the new AR. If this MA is different than the previous one (inter-MA handover), then an extra cost Z_{kj} is defined, which is added to the total

mobility cost:

$$Z_{kj} = \alpha X_{kj} + \omega \tag{3.5}$$

where X_{kj} is the distance between the two ARs, α is a weight variable and ω is a constant, which represents the signalling cost to the home agent through the Internet. The total mobility cost Ψ, due to the total handovers, can be written as follows:

$$\sum_{k\in K}\sum_{m\in J}\sum_{j\in T}\left\{h_{kj}\left(d_k x_{jm} p_{jm} + Z_{kj}\left(x_{jm} - G_{kjm}\right)\right) + \sum_{l\in T} w_{kjl}\left(d_k x_{lm} p_{lm} + Z_{jl}\left(x_{lm} - G_{jlm}\right)\right)\right\} \tag{3.6}$$

Finally, the total cost is the summation of the routing cost and the mobility cost: $\Xi + \Psi$.

Below, the problem is expressed in a mathematical programming setting:

$$\text{minimize} \quad (\Xi + \Psi)$$

subject to

$$\sum_{m \in J} M_m \leq L \tag{3.7a}$$

$$x_{km} \leq M_m \qquad \forall k \in K \ , \forall m \in J \tag{3.7b}$$

$$\sum_{k \in K} x_{km} d_k \leq C_m M_m \qquad \forall m \in J \tag{3.7c}$$

$$\sum_{k \in K} d_k - \sum_{k \in K} \sum_{m \in J} x_{km} d_k \leq 0 \tag{3.7d}$$

$$A \left(\sum_{k \in T} x_{km} - q \right) \leq M_m - 1 \quad \forall k \in K \ , \forall m \in J \tag{3.7e}$$

$$G_{kjm} \leq x_{km} \quad \forall k \in K, \quad \forall j \in T, \forall m \in J \tag{3.7f}$$

$$x_{km} + x_{jm} - G_{kjm} \leq 1 \quad \forall k \in K, \forall j \in T, \forall m \in J \tag{3.7g}$$

$$\sum_{m \in J} x_{km} = 1 \qquad \forall k \in K \tag{3.7h}$$

$$\sum_{j \in T} \sum_{m \in J} x_{jm} = q \tag{3.7i}$$

where constraint (3.7a) sets the maximum number of MAs that will be deployed, (3.7b) ensures that if flow k will be anchored to an MA m, this MA has to be deployed, (3.7c) is a capacity constraint in order not to exceed the maximum capacity of MAs m, (3.7d) ensures that all of the demands have to be satisfied, (3.7e) ensures that all of the flows with the same destination AR have to be anchored to the same MA m, (3.7f) and (3.7g) make sure that if and only if both ARs' domains are served by the same MA, decision variable G has to be 1. Equality constraint (3.7h) makes sure that each flow will be anchored to only one MA and (3.7i) sets the number of flows per AR. Note that $A \in \mathbb{N}$ can be a relatively big natural number (e.g. $A = 2q$). As stated in the previous section, the proposed mathematical programming setting is inspired by [36], which has been augmented to consider multiple flows and sessions, that can experience multiple handovers (one-hop and two-hop).

3.3.2 Distributed Mobility Management

For the DMM scheme scenario, the mobility management function is distributed at the ARs of the network. As a result, in the first place, all of the flows are routed through the shortest paths. The routing cost R_k for each flow k is:

$$R_k = d_k p_{gj} \tag{3.8}$$

where d_k is the data demand of the flow and p_{gj} is the shortest path from the source node g to the destination AR j.

Regarding the handover procedure, when a flow migrates from an AR j to another AR i, the traffic is tunnelled to the new AR i (Fig. 3.1b), with an added overhead o. In this scenario, the possible handovers that can occur are, also, one-hop and two-hop handovers. The mobility of the users is expressed by the handover matrix H, where $h_{kj} \in (0, 1), h_{kk} = 0$ represents the probability that a flow k moves to an AR j, whether the new AR is one or two hops away. The cost caused by the mobility of the users for one flow k is:

$$H_k = o h_{kj} d_k p_{ji} \tag{3.9}$$

where p_{ji} is the shortest path from the AR j to the AR i and o the overhead, expressed as the ratio of the packet length of the tunnelled flow to the initial one. Finally, the total cost for all of the flows is calculated as follow:

$$C = \sum_{k \in K} (R_k + H_k) \tag{3.10}$$

3.3.3 Hybrid Distributed Mobility Management

According to the results, which are presented analytically in the next section, the aforementioned mobility management schemes and especially the DMM scheme (for

instance in heterogeneous networks, where there can be no direct connectivity between different access points) can suffer from strong topology dependence. This means that, although DMM is a promising mobility management solution for next-generation wireless networks, there can exist areas across the network where the migrating flows suffer from sub-optimal routing and strong latency.This can affect the QoS to the end users, especially when using delay-sensitive applications (*i.e.* VoIP).

Motivated by this, a hybrid scheme is proposed, where the mobility function is distributed at the ARs, except for certain areas, where handover procedures cause high latency due to the underlying network topology characteristics. In these areas of the network, the mobility of the flows is supported by mobility agents which are located hierarchically higher across the network, as seen in Fig. 3.1c. The number and location of those MAs can be found by solving the optimisation framework that is presented in section 3.3.1 for those flows.

As shown in algorithm 1, the total cost of every flow is initially calculated , considering that DMM scheme is adopted and a threshold γ is defined, above which the distributed mobility support switches to a centralized mobility management. Afterwards, the flows which will switch from DMM support to centralized are grouped. Then, the optimisation framework is solved, as it was described in 3.3.1, for these flows only, in order to find the optimal location and selection of MAs which will serve these flows. (in Fig. 3.1c this occurs at the two ARs in the middle of the topology). Then, the mobility support for these flows is handled by MAs (in Fig. 3.1c the MA has been located at the source node). The cost for the flows that are supported by DMM scheme can be computed as in section 3.3.2, while for those which are supported by the centralized anchors can be calculated as in section 3.3.1.

Algorithm 1: HYBRID–DMM SCHEME

Step 1 Calculate the cost of every flow, using DMM
Step 2 Find the flows for which the total cost is greater than a predefined threshold γ
Step 3 Solve the integer programming problem, as described in 3.3.1, for those flows only
Step 4 Anchor those flows to the mobility agents found by the optimisation framework, using CMM. The rest will be supported by DMM scheme

3.4 Numerical Evaluation

An important aspect, which has to be considered when implementing a mobility management scheme, is how the physical topology of the network can affect the overall network performance. For this reason, the centralized mobility management scheme is compared to the DMM scheme, as it is described by IETF in [9]. In terms of topology dependence, two different types of network topologies are simulated; a sparse tree-like network (which incorporates the binary tree as the worst case in terms of available connectivity) and a highly interconnected tree-like network (Figures 3.1a,b accordingly). The optimisation framework was solved using MATLAB's optimisation toolbox.

The networks used in these simulations consist of 256 nodes, the weights of the links are randomly assigned using a uniform distribution with integer values that range from 8 to 12 and the capacities of the nodes are set to at least 20% of the total demand. Regarding the centralized scheme, two MAs are deployed, similar to the network shown in Fig. 3.1a and $\omega = 30$, $a = 1.1$. The sparse network is a topology similar to Fig 3.1a and the dense network similar to Fig 3.1b. There are $q = 10$ flows per AR and the demands are randomly assigned with a uniform distribution ranging from 10 to 15 units.

As Fig. 3.2 shows, the DMM scheme can be affected in a greater degree by topological characteristics of connectivity in the network (25% increase in the cost for the sparse

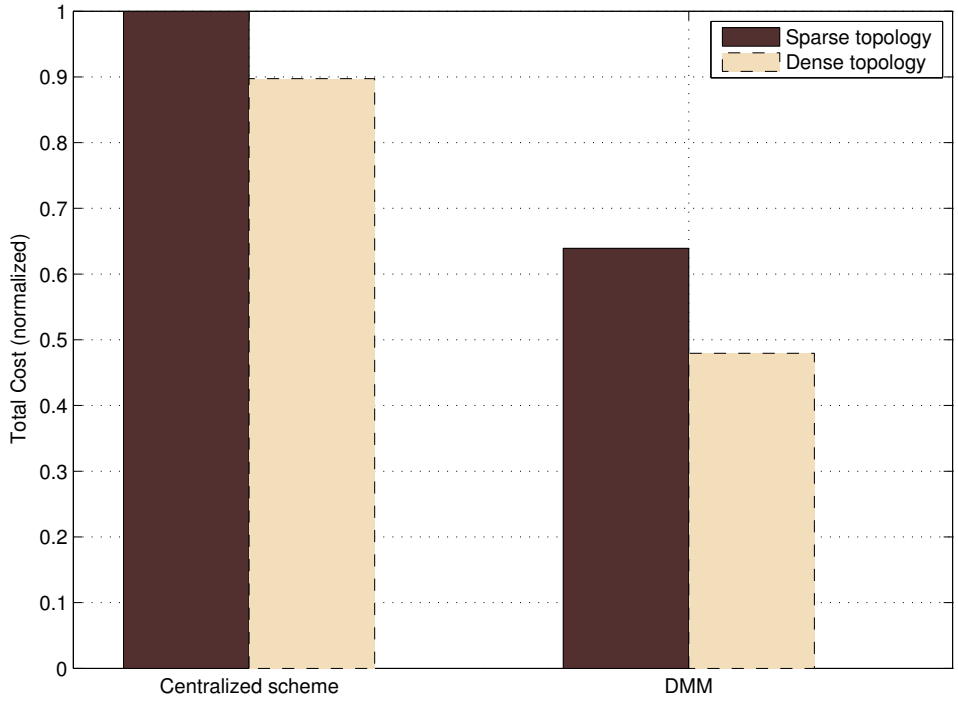

Figure 3.2: Total cost for different topologies

topology), than the centralized mobility management scheme (11% increase in cost for sparse network). The reason is that, although the DMM scheme offers optimal routing, while flows are tunnelled to a neighbour AR, during the handover procedure, depending on the topology, they can be routed through unnecessarily longer paths, which relates to the connectivity of the network and the distance between the ARs.

In the simulations, a traffic model is considered where users can have sessions amortized in terms of number of handovers that can take place during the session time. In that respect, the baseline assumption is that sessions last for one or two handovers. In Fig. 3.3 the total cost for centralized and DMM schemes in different mobility cases is presented. Each case is described from a pair of probabilities; the first one is the probability of one-hop handovers and the second one is the probability of two-hop handovers (e.g. the pair (0.7,0.4) in the x-axis means that the average probability for one-hop handovers is 70% and the average probability for two-hop handovers is

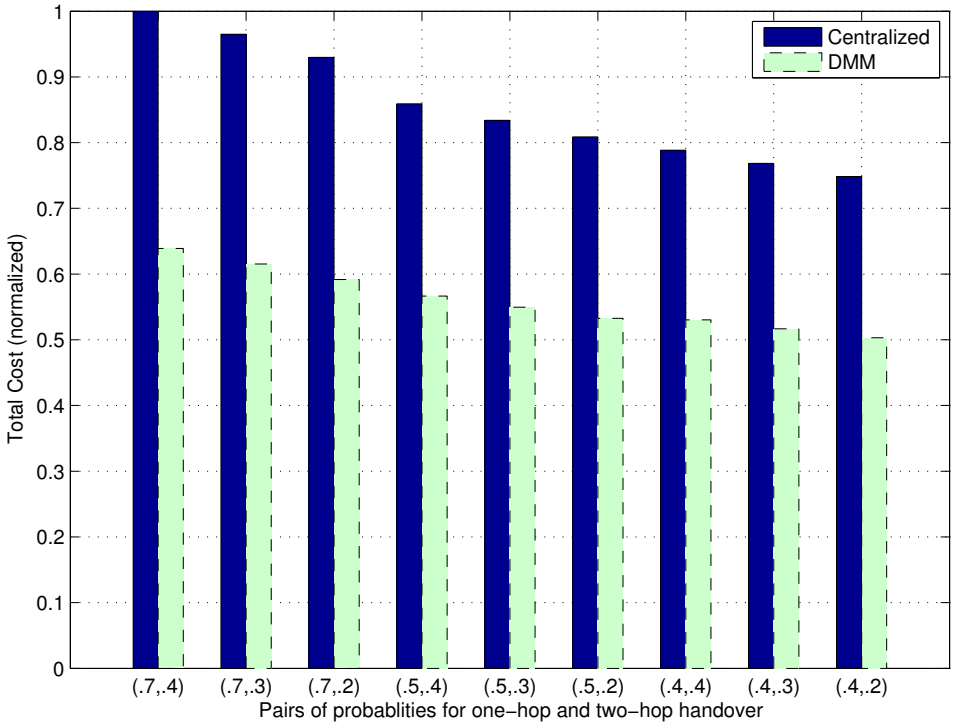

Figure 3.3: Total cost for different mobility scenarios

40%, for flows that have already had an one-hop handover). The simulation results confirm that the DMM outperforms the centralized mobility management scheme in terms of total cost in all evaluated cases, having up to 35% less total cost for a high mobility scenario.

Furthermore, although DMM performs better than the centralized scheme, as already shown, the focus is given not only on the overall performance, but, also, on the QoS for each particular flow. Fig. 3.4 presents the total cost per group of flows with the same destination AR (in this figure, the network has 511 nodes and there are depicted 256 different ARs, numbering from left to right, as in Fig 3.1a). The results show that there are areas across the network where particular flows experience a total cost of up to 30% more than the network's average. In addition, as it is shown, there are destinations nodes (ARs) where the flows destined to them appear to have a total cost greater than the depicted threshold γ (87%).

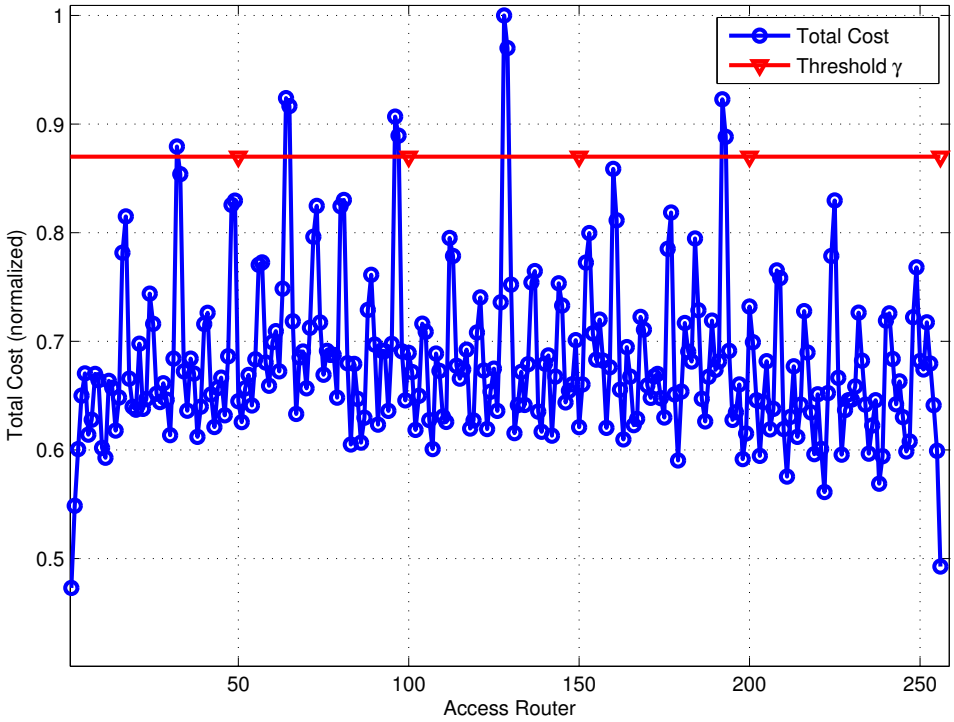

Figure 3.4: Total cost for each AR domain in DMM scheme

The motivation behind the Hybrid Distributed Mobility Management (HDMM) scheme is the seamless mobility for delay sensitive applications. The main goal is to improve the QoS of those flows, that have increased total cost, greater than a specific threshold γ, due to the network topology characteristics, as shown in Fig. 3.4. In this simulation the threshold is set as $\gamma = 87\%$ of maximum total cost. To this end, the proposed HDMM scheme is implemented by utilizing the mobility support for those flows with cost greater than the average to the 2 MAs located at hierarchically higher in the network (the nodes which have been found with the optimisation framework of 3.3.1). The performance of the DMM and the HDMM scheme is presented in Fig. 3.5. As it is shown, the gain of the proposed HDMM scheme for the aforementioned flows can be up to 17%, depending on the probabilities of the occuring handovers (i.e. last case in Fig. 3.5).

For the rest of this work, the HDMM [40] scheme is not going to be deployed, since

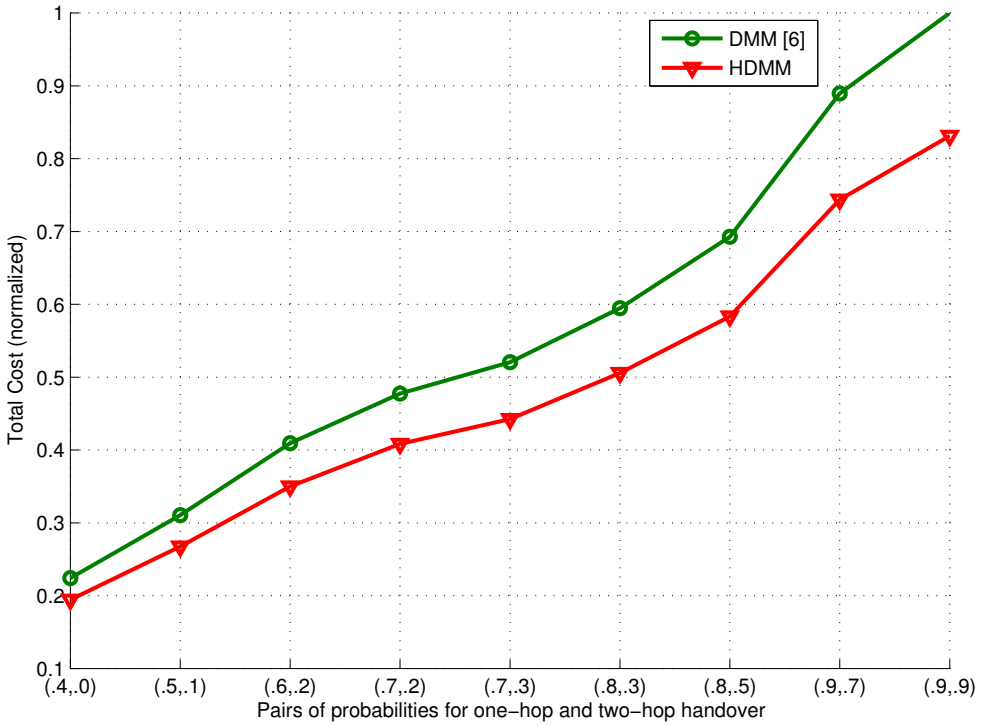

Figure 3.5: Total cost of flows supported by DMM and Hybrid-DMM

the main focus will be given on the area of Network Optimisation and Sharing and particularly on the field of Virtual Network Embedding (VNE); the DMM scheme is going to be used instead. The reason behind that is, primarily, because the DMM scheme is already an active IETF standard and it is currently widely accepted. Moreover, although there are cases where the HDMM scheme indeed outperforms DMM, this applies when specific criteria are satisfied, as it was shown in this section, hence, its use and implementation goes beyond the scope of the next chapters.

3.5 Publications

1. G. Chochlidakis and V. Friderikos, "Hybrid Distributed Mobility Management for Next- Generation Wireless Networks", in *IEEE International Conference on the Network of the Future (NoF14), Paris, France*, Dec. 2014.

Chapter 4

Mobility Aware Virtual Network Embedding

Over the last few years, network virtualisation has become one of the most promising solutions for sustainability towards the ongoing increase of data demand in mobile networks. Within that context, the virtual network embedding problem has recently been studied extensively and many different solutions have been proposed; but mainly these studies have focused on wired networks. The main purpose of this chapter is to provide an optimisation framework for optimal virtual network embedding, including a heuristic algorithm with low computational complexity, by explicitly considering the effect of supporting the actual user mobility, assuming the emerging Distributed Mobility Management (DMM) scheme as well as a traditional Centralized Mobility Management (CMM) scheme. In addition to that, service differentiation is introduced, giving higher priority to time-critical over-the-top (OTT) services compared to more traditional elastic Internet applications. The performance of the proposed framework is compared to mobility agnostic greedy algorithms as well as virtual network embedding algorithms from the literature. Numerical investigations reveal that the effect of user mobility has a significant impact on the design

of virtual networks. Additionally, the mobility aware scheme can provide tangible gains in the overall performance compared with the previous proposed schemes that do not take explicitly into account the effect of user mobility.

4.1 Introduction

Mobile network operators worldwide are witnessing a dramatic increase of data demand due to the introduction of smartphones and the plethora of Internet applications that they can support. At the same time, their revenues are reaching a plateau due to a flat charging model, turning them into 'dumb pipes' for the application providers [41]. For this reason, and in parallel to what has been the case in wired networks and data-centres, the virtualisation of resources, in order to achieve efficient network sharing, has been considered as a promising solution for next-generation networks, including the scope of 5G. It is worth pointing out that some form of passive (non-adaptive) infrastructure sharing already exists within the cellular operators and as it has been observed, by the end of 2015, 90% of mobile operators will have explored this avenue in some form[1].

Clearly, this trend is expected to further continue in the future, but sharing will inevitably have to move deeper into the network. This will give the flexibility of more dynamic sharing via network virtualisation techniques, allowing for multi-tenancy at different network elements within the core and wireless access network. The most significant advantages that network virtualisation carries are the increase of utilisation of the available physical resources, the energy efficiency, the flexibility and scalability that can offer and finally the prospect of sustainability for mobile operators [42][43].

In network virtualisation, the physical resources (e.g. nodes and links) are in essence

[1]Mobile Network Sharing Report Developments, www.reportlinker.com

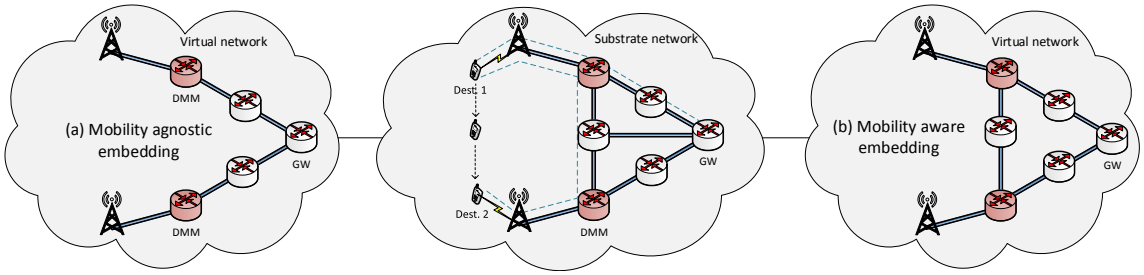

Figure 4.1: (a) Mobility agnostic and (b) mobility aware embedding algorithm

virtualised into isolated slices, in order to form virtual networks, respectively to virtual network requests, in a procedure known in the literature as virtual network embedding or mapping. As already stated, the problem of virtual network embedding has been studied extensively and a plethora of different approaches have been proposed so far. A meticulous review of the research progress so far in the scope area of virtual network embedding is presented in [23] where the authors outline the most important optimisation and heuristic algorithms. Then, they categorize the algorithms along three main dimensions: static versus dynamic, centralized versus distributed and concise versus redundant solutions.

An important issue that has to be taken into account from the outset while developing a virtual network embedding algorithm for mobile networks is the actual effect of users mobility. The way that the mobility support is implemented (i.e. the mobility management scheme that is deployed) affects the core network's congestion, the overall network performance and the availability of the resources. For this reason, mobility subsequently affects the virtual network embedding procedure.

Since the proposal of Mobile IPv6 (MIPv6) [3] by the Internet Engineering Task Force (IETF), various all-IP mobility management schemes have been standardized so far. The modern trend is to move from centralisation to the distribution of the mobility function at the edge of the core network. To this end, after setting the requirements [10], IETF proposed the Distributed Mobility Management scheme [9], where each edge router can potentially become a mobility anchor for the users. According to DMM, when a handover occurs, the active flows are tunnelled directly

from the old to the new edge router that is connected to the new base-station. Hereafter, it will be assumed that mobility management is handled in a distributed manner using a DMM compliant solution, where DMM mobility anchor points are located onto the edge routers.

In this chapter a core mobile network physical infrastructure is considered that has to be shared by multiple tenants after different virtual networks are efficiently set up, according to virtual network requests. The requests are classified into different classes depending on the type of the service that has to be delivered. A linear integer mathematical programming formulation is developed, which also takes into account the effect of user mobility on the design of the virtual networks. The main contribution of this work is that it reveals and examines in a conscientious manner the interaction of physical network topology and traffic tunnelling at the mobility anchors on the efficient construction of virtual networks. To this end, the proposed integer programming formulation considers, in an explicit way, the user mobility and the effect of that mobility on the available network resources. Also, the proposed scheme performs a prioritisation among multiple tenants with potential differentiation in terms of QoS but the model can be easily extended to differentiate charging business model and/or Quality of Experience (QoE).

In Fig. 4.1 a simple example of the outcome of a mobility agnostic and mobility aware [44] virtual embedding algorithm respectively is presented. Firstly, a substrate network topology is considered, where a virtual network has to be formed in order to serve a class of flows, a fraction of which moves between the two edge routers due to expected user mobility, as depicted in the figure. The handover of the flow is handled by DMM scheme and the intermediate path between the two edge routers is deployed for the data forwarding. Then, as shown in Fig. 5.1a, given the gateway and the destination edge router as defined by the virtual network request, the mobility agnostic algorithm will not inevitably include in the formed virtual

network the shortest routing paths that are important for the efficient handover procedure substrate tunnelling path. In this way, the DMM tunnelling will have to be routed through a path consisted of 5 intermediate physical nodes instead of the shortest one consisted of 3 nodes.

On the other hand, the proposed mobility aware algorithm takes explicitly into consideration not only the source and destination nodes but also the DMM tunnelling path for the fraction of the flows that would require mobility support. Hence, the algorithm will firstly map the two shortest paths that connect the gateway node with the base-stations (for flows that start from there) and it will also include the intermediate path as shown in Fig. 5.1b. In this way, the DMM tunnelling path, connecting the previous DMM edge router with the new one, will be improved, consistently in this case, as shown in the example, of 3 nodes.

The key difference between the two approaches is that an embedding algorithm that explicitly utilizes mobility information will map the required paths that will be used for the data tunnelling to the next mobility anchor point where the flow will migrate. Considering the users mobility effect, such a mobility aware algorithm will create virtual networks that can support migrating flows in a significant more efficient manner.

4.2 Selected works from the literature

4.2.1 Related virtual network embedding algorithms

Different approaches have been proposed so far addressing the virtual network embedding problem. In this chapter selected related works from the literature are presented, some of them having been already mentioned in Chapter 2. Firstly, the authors in [45] formulate the virtual embedding problem as a mixed integer pro-

gram and then propose virtual embedding algorithms by introducing a coordination between node and link mapping phases. More specifically, they relax the integer constraints to get a linear programming setting by using deterministic and randomized rounding techniques and so they manage to achieve polynomial-time albeit suboptimal solutions for virtual network embedding. The simulation results show that their algorithms outperforms the existing approaches in terms of acceptance ratio, revenue and provisioning cost.

The authors in [46] propose an linear integer programming formulation that solves the online virtual network embedding problem. Their solution aims to minimize the total resource consumption and to achieve load balancing at the network. To this end, they introduce three cost functions: one that minimizes the total load on each virtual network, one that minimizes the number of links that are mapped and selects nodes with higher available resources and one that includes the demanded capacity by the virtual network requests in the objective function. The simulation results show that their proposal outperforms in general the different compared heuristic approaches.

A virtual network embedding distributed protocol, named *MADE* suitable for mobile environments, where the nodes are not static and the substrate network is dynamic is presented in [47]. The authors apply the path splitting and migration techniques in order to optimize the utilisation of the available physical resources. The simulation results show the efficiency of this protocol in terms of acceptance and completion of the requested virtual networks.

In [48] a distributed algorithm is presented, which performs load balancing and virtual network embedding over a substrate network. The algorithm makes use of a proposed mapping protocol that enables the communication and the exchange of messages between the substrate network's nodes in order to ensure distributed negotiation and synchronisation for the virtual network set up. The performance

results show that in the distributed mapping approach the number of messages exchanged can have an important impact on the overall performance but, on the other hand, compared to the centralized approach, their proposed algorithm can reduce the time delay and process multiple parallel virtual network requests.

The authors in [49] develop a scalable embedding algorithm named *VNE-AC*, based on ant colony meta-heuristic. The algorithm aims to minimize the allocated physical resources of the physical substrate network for each request in order to minimize the reject rate and to maximize the provider's revenue. Results from the simulations show that the proposed algorithm achieves better overall performance in comparison to related algorithms from the literature.

In [50] an alternative approach s presented for the virtual embedding problem, which focuses on rethinking the design of the substrate network per se in order to enable less complex algorithms and increase the utilisation of the resources without the restriction of the problem space. In order to achieve flexibility of the substrate network, path splitting and migration as well as customized node-mapping algorithms are used. The simulations results show that the proposed solution is competitive and it manages to increase the resources' utilisation.

In [51], the authors focus on solving the problem of virtual network embedding as the substrate network evolves. To this end, they present an integer programming formulation of this problem which aims to minimize the upgrading cost of the virtual network with respect to node resource and path delay constraints. Because of the problem's complexity the authors develop a heuristic algorithm and they present its efficiency through simulations.

A different approach is presented in [52] where the main purpose of the proposed virtual embedding algorithm is to deploy virtual networks according to the client's requirements in a reliability context with bandwidth feasibility and reliability calculations.

Finally, the authors of [53] focus on two categories of the virtual network assignment problem: virtual network assignment without configuration and with configuration. They try to provide heuristic algorithms and optimisation strategies in order to increase the efficiency in the resource utilisation and to be able to deal with the demands real-time. One of the proposed algorithms will be presented extensively in the next section and will be used in order to compare this proposal.

In all the above previous research works the effect of mobility has not been considered and, as shown in the sequel, this is an important parameter that needs to be taken into account for creating more efficient virtual networks.

4.3 System Model

In order to evaluate the performance of the proposed algorithm, it is compared to different approaches under the same assumptions. Firstly a substrate core network is mathematically formulated. The aim is to form virtual networks under requests by combining the physical resources, under specific constraints. Then the performance of the virtual networks is evaluated under the same traffic model and mobility of the users.

4.3.1 Implementation of virtual network embedding algorithms

In order for the virtual network requests to be served, it is assumed that a central controller, as discussed in section 2.2.2, exists, being responsible for slicing the physical resources and providing isolation between them. In particular, the controller is handled by one of the hereafter algorithms (Fig. 2.7). This means that the embedding algorithm creates a plan/strategy for virtualisation of the resources and

for their assignment to the different requests. Each algorithm has different strategy and logic regarding the most efficient mapping, depending on the parameters that it takes into consideration.

After the decision for the assignment of the resources, the controller is responsible for communicating with the network elements and forming the slices. In this way, the virtual networks are created and they are ready to serve the flows which they are responsible to handle. Each slice is totally isolated from the others and has no awareness of the existence of the rest slices.

At this point it has to be clarified that the MN is considered to have limited network functionality and intelligence in terms of network optimisation decisions. Hence, those network orchestration decisions are taken by the aforementioned centralised controller. In this way, there is no need to assume that every mobile node will have enough computational and memory resources to carry out those network-related functions. In addition, considering the limited battery capacity of commercial mobile terminals, the low network intelligence prevents the unnecessary energy consumption due to high processing load.

Regarding the architecture, for the simulation topology, a single gateway and several destination edge routers is assumed. Each virtual network request is defined by demands from the source gateway to the end edge routers. Hence, the formed virtual networks have a substrate topology dependence and should include the same source and destination nodes.

As for the mobility management function, since DMM scheme is deployed, every single destination edge router acts as a mobility management anchor point. Hence, it is assumed that by applying SDN and NFV deployment, the mobility function is virtualised by co-locating it at the edge of the core network (i.e. the edge routers).

4.3.2 Optimal mobility aware virtual network embedding

In this section,the proposed mathematical programming setting for optimal, in terms of routing cost, mobility aware virtual embedding is detailed. In order to provide a formal model for the virtual network embedding process, and in order to develop the optimisation framework, firstly the physical substrate network is modelled, following the requests for virtual networks, the objective function and the problem constraints.

Substrate network

The substrate network can be modelled as an undirected planar tree-like graph $G = (V, E)$, where the set V represents the set of nodes and the set E depicts the set of links. Let $B : E \rightarrow \mathbb{R}_{>0}$ be the cost of each link and $C : V \rightarrow \mathbb{R}_{>0}$ the capacity of each node. The gateway is located at the root of the graph, while the leaves of the tree represent the destination edge routers. The intermediate nodes are the routers of the core network. The algorithm objective is to assign nodes and links according to the virtual network requests in a way that it will minimizes the total weighted routing cost.

Virtual network requests

The proposed algorithm offers a multi-tenant prioritisation. This can be applied in a case where the infrastructure owner provides a tenant differentiation according to desirable QoS requirements. This, also, applies to a scenario where a prioritisation in favour of one type of service over another one with lower priority is required. For this reason this feature of the proposed algorithm is going to be referred to as the tenant or type of service classification/differentiation.

Let Q represent the set of sets of the virtual network requests for all tenants. More-

over, let U represent the set of the different tenants. Then, let $Q^u \in U$ represent the set of virtual network requests for a tenant $u \in U$. The set Q can be described as follows: $Q = Q^{u_1} \bigcup Q^{u_2} \bigcup ... \bigcup Q^{u_{|U|}}$.

In order to define the set of virtual network requests $Q^u \in Q$, each request $q \in Q^u$ is described by a set of classes of unsplittable flows $k \in K^{uq}$ that have to be satisfied and routed from the root gateway to the edge routers $j \in T$ by the formed virtual networks. Also, each class of flows $k \in K^{uq}$ corresponds to a total demand d_k^{uq}.

Hence the virtual network request $q \in Q^u$ can be represented as a graph $G' = (V', E')$ that has to be mapped on the graph $G = (V, E)$. The graph $G' = (V', E')$ is consists of virtual paths $\pi_{kp} \in P$ that connect the virtual gateway with the virtual edge routers. Each class of flow of a virtual network request can be considered as a demand for resource allocation across the core network.

Hereafter, it is explicitly considered a per-class allocation of the available network resources allowing for scalability and assuming that requested traffic demand incorporate sufficient slack values, so that traffic variations (congestion episodes) during the duration of the virtual network are taken into account.

In order to capture the actual mobility of the users, a handover matrix $H_{K \times K}$ is defined, the elements $h_{kj} \in (0, 1)$ represent the probability of the flow $k \in K^{uq}$ to migrate to another edge router $j \in T$ (note that each edge router j can be described by a flow $k \in K^{uq}$). When DMM scheme is in use, there is an additional need to assign paths that connect edge routers or network elements where DMM anchoring is taking place.

Regarding the assumption of the mobility as a concept throughout this work, this is expressed by a set of probabilities of the flows to be switched to a different domain due to an occurring handover. Hence, this could be translated to the proportion of the sessions that are handled by handovers, due to user mobility and varies,

depending on the demographics, the geography of the served area, the structure of the residential area and other parameters.

In this work and without loss of generality it is assumed that DMM anchoring takes place at the edge router and for this reason, for each edge router of class of flow $k \in K^{uq}$ towards an edge router $j \in T$, the set of substrate paths $r_{kji} \in R$ is considered. In the same way, each tunnelled class of flows will be routed through one of the available paths.

However, the optimisation algorithm can be adapted for different other mobility management solutions, since mobility cost depends on the mobility management scheme that is used. In conclusion, the algorithm, taking into account the expected probabilities of handovers, will also map paths for the virtual networks for more efficient data forwarding.

Problem variables

Based on the above setting, the goal is to find the optimal selection of routing paths in order to minimize the total routing cost and at the same time to achieve tenant differentiation. Below, there is a summary of the variables that are used for the formulation of the integer mathematical program and the introduction of the decision variables:

- $G = (V, E)$: undirected planar graph
- π_{kp}: set of substrate paths from source to edge router k (expressed in routing cost)
- r_{kji}: set of substrate paths from the edge router defined by flow $k \in K^{uq}$ to edge router $j \in T$ (expressed in routing cost)
- U: set of tenants

- Q^u: set of virtual network requests of tenant $u \in U$

- K^{uq}: set of classes of flows that have to be served by virtual network of request $q \in Q^u$

- d_k^{uq}: set of demands for class of flows $k \in K^{uq}$ of tenant $u \in U$ and virtual network request $q \in Q^u$

- $z_{kp}^n = \begin{cases} 1, \text{ if node } n \in \pi_{kp} \\ 0, \text{ otherwise} \end{cases}$

- $l_{kji}^n = \begin{cases} 1, \text{ if node } n \in r_{kji} \\ 0, \text{ otherwise.} \end{cases}$

Then, the following problem binary variables are defined:

$$x_{kp}^{uq} = \begin{cases} 1, \text{if } \pi_{kp} \text{ is assigned to k } \in K^{uq} \\ 0, \text{ otherwise} \end{cases} \tag{4.1}$$

$$y_{kji}^{uq} = \begin{cases} 1, \text{if } r_{kji} \text{ is assigned to k } \in K^{uq} \\ 0, \text{ otherwise.} \end{cases} \tag{4.2}$$

Objective function and problem's constraints

The total routing cost ϕ can be written as:

$$\phi = \sum_{u \in U} \sum_{q \in Q^u} \sum_{k \in K^{uq}} \sum_{p \in P} d_k^{uq} \pi_{kp} x_{kp}^{uq} \tag{4.3}$$

and expresses the cumulative routing cost of the aggregated set of flows that will use the formed virtual networks with the expected demands. The total mobility cost M

can be written as:

$$M = \sum_{u \in U} \sum_{q \in Q^u} \sum_{k \in K^{uq}} \sum_{j \in T} \sum_{i \in I} h_{kj} d_k^{uq} r_{kji} y_{kji}^{uq} \tag{4.4}$$

and it is the routing cost of the forwarded traffic caused by the mobility of the users.

The total cost T is the summation of the routing cost and the mobility cost:

$$T = \phi + M \tag{4.5}$$

and the total cost T_u for the tenant $u \in U$:

$$\sum_{q \in Q^u} \sum_{k \in K^{uq}} \Big(\sum_{p \in P} d_k^{u_1 q} \pi_{kp} x_{kp}^{u_1 q} + \sum_{j \in T} \sum_{i \in I} h_{kj} d_k^{u_1 q} r_{kji} y_{kji}^{u_1 q} \Big) \tag{4.6}$$

Based on the above definitions the mathematical program can be formulated as follows:

$$\textbf{minimize } T \tag{4.7}$$

subject to

$$\sum_{u \in U} \sum_{q \in Q^u} \sum_{k \in K^{uq}} \Big(\sum_{p \in P} d_k^{uq} x_{kp}^{uq} z_{kp}^n + \sum_{j \in J} \sum_{i \in I} h_{kj} d_k^{uq} y_{kji}^{uq} l_{kji}^n \Big) \leq C_n \ , \forall n \in V \tag{4.8}$$

$$w_n T_{u_n} \leq w_{n+1} T_{u_{n+1}} \ , \forall u, q, k, p, i \tag{4.9}$$

$$\sum_{j \in J} \sum_{i \in I} y_{kji}^{uq} = \mathbb{1}(h_{kj}) \ , \forall \ u, q, k \tag{4.10}$$

$$\sum_{p \in P} x_{kp}^{uq} = 1 \ , \forall \ u, q, k \tag{4.11}$$

$$x_{kp}^{uq}, y_{kji}^{uq} \in \{0, 1\}, \ \forall \ u, q, k, p, i \tag{4.12}$$

where constraint (4.8) makes sure that the capacity of each node is not violated.

Constraint (4.9) ensures the classification of the two types of service by introducing the weighted constraint. Moreover, constraint (4.10) guarantees that if there is a handover between two ARs (defined by matrix h, then one and only one path will be used. Note that $\mathbb{I}$ is an indicator function defined as follows:

$$\mathbb{I}(n) = \begin{cases} 1, \text{ if } n \neq 0 \\ 0, \text{ if } n = 0. \end{cases} \tag{4.13}$$

Finally, constraint (4.11) ensures that among all the alternative paths which connect the source node with an edge router k only one will be used and constraint (4.12) makes sure that the decision variables will be boolean. Since integer linear programming is NP-complete [54], problem instances related to large networks might be intractable and so low complexity heuristic methods are useful for such instances. In the next subsections, heuristic approaches from the literature and proposed by us are presented in order to evaluate the performance of the proposed scheme.

4.3.3 Greedy mobility agnostic heuristic algorithm

For the first heuristic approach of the above problem, a generalized greedy algorithm is implemented, as the heuristic algorithm presented in [50] (*Algorithm 1 - Greedy Node Mapping Algorithm*), by augmenting it to multiple traffic classes.

The algorithm firstly sorts the demands of every virtual network request of each tenant. Then, it handles firstly the high-priority one in the following way: the flows with the higher aggregate demands utilize the lower-cost routing paths, until a threshold on the congestion level of these paths is reached, and then the flows with the lower aggregate demands follow. After the high-priority tenant has been assigned with routing paths which form the requested virtual network, the algorithm serves the second tenant, which has lower priority.

Algorithm 2: Greedy Mobility Agnostic Heuristic Algorithm

INPUTS: Graph $G = (V, E)$, number of tenants U, virtual network requests Q, classes of flows K, set of paths P, set of demands d, set of capacities C
OUTPUTS: Set of mapped substrate paths for every virtual network request

```
1: for u = 1 to U do
2:    sort (descending) demands d^uq ∀q ∈ Q, k ∈ K
3:    for q = 1 to Q do
4:       for k = 1 to K do
5:          p = 1
6:          repeat
7:             if path π_kp not congested then
8:                map path π_kp to request q
9:                decrease available capacity of path π_kp
10:               flag = 1
11:            end if
12:            p = p + 1
13:         until flag = 1 or p > P
14:         flag = 0
15:      end for
16:   end for
17: end for
```

The steps of this algorithm are presented in Alg. 2. Note that the mobility agnostic greedy algorithm embeds networks where the mobility is not taken into account. In this way, the tunnelling of the flows from the previous to the new DMM edge routers will use the mapped paths, which connect the gateway with the edge nodes (as it has been already explained at the exampled shown in Fig. 5.1).

4.3.4 Greedy mobility aware heuristic algorithm

After being implemented, the greedy node mapping algorithm as was proposed in [53] for this case, is extended in order to be able to take into account the mobility effect.

In particular, the algorithm is like Alg. 2 but also assigns intermediate paths for the

data forwarding between the edge routers, during the handover procedure. Hence, the algorithm, also, sorts the available pre-calculated paths that connect the edge routers and it assigns them to the virtual network requests, supporting firstly the high demanded ones. In addition, like the previous two presented algorithms, this algorithm also performs tenant prioritisation. The steps of this algorithm are summarized in Alg. 3.

4.3.5 Basic virtual network assignment mobility agnostic algorithm

After having implemented two greedy heuristic approaches, the focus is given on implementing another virtual network algorithm from the literature which takes into account the node and link stresses. In particular, the main aim is to compare the proposed solution to *Algorithm 1 - Basic VN Assignment Algorithm* that is presented in [53].

In order to adapt this algorithm for the current scenario, it is assumed that the algorithm maps the virtual source (gateway) to the physical one and the virtual sink nodes to the physical edge routers.

Then, it maps paths which have the minimum Π upon the i^{th} arrival of a virtual network request, as it is defined below:

$$\Lambda(\pi_{kp}) = \frac{\sum_{e \in p_{kp}} \frac{1}{S_{lmax}(i)+\delta_L - S_L(i,e)}}{S_{nmax(i)+\delta_N - S_N(kp,i)}}, \tag{4.14}$$

where e denotes a link of a path p_{kp}, S_{lmax} and S_{nmax} the current maximum link and node stress respectively and $\delta_L = \delta_N = 1$ a relatively small number to avoid dividing by zero.

Regarding the notion of stress, for a substrate node n the stress $S_N(n)$ is defined as

Algorithm 3: Greedy Mobility Aware Heuristic Algorithm

INPUTS: Graph $G = (V, E)$, types of services U, virtual network requests Q, classes of flows K, set of paths P and R, set of demands d, set of node capacities C and handover matrix $H_K \times K$
OUTPUTS: Set of mapped substrate paths for every virtual network request

for $u = 1$ to U **do**
  sort (descending) demands d^{uq} $\forall q \in Q, k \in K$
  for $q = 1$ to Q **do**
    for $k = 1$ to K **do**
      $p = 1$
      repeat
        if path π_{kp} not congested **then**
          map path π_{kp} to request q
          decrease available capacity of path π_{kp}
          $flag_1 = 1$
          for $j = 1$ to K **do**
            $i = 1$
            repeat
              if path r_{kji} not congested **and** $h_{kj} \neq 0$ **then**
                map path r_{kji} to request q
                decrease available capacity of r_{kji}
                $flag_2 = 1$
              end if
              $i = i + 1$
            until $flag_2 = 1$ **or** $i > R$
            $flag_2 = 0$
          end for
        end if
        $p = p + 1$
      until $flag_1 = 1$ **or** $p > P$
      $flag_1 = 0$
    end for
  end for
end for

Algorithm 4: Basic VN Assignment Mobility Agnostic Algorithm

INPUTS: Graph $G = (V, E)$, number of tenants U, virtual network requests Q, classes of flows K, set of paths P, set of demands d, set of capacities C
OUTPUTS: Set of mapped substrate paths for every virtual network request

```
1: for u = 1 to U do
2:   for q = 1 to Q do
3:     for k = 1 to K do
4:       p = 2
5:       repeat
6:         if path π_kp not congested and Λ(π_kp) < Λ(π_k(p−1)) then
7:           map path π_kp to request q
8:           decrease available capacity of path π_kp
9:           flag = 1
10:        end if
11:        p = p + 1
12:      until flag = 1 or p > P
13:      flag = 0
14:    end for
15:  end for
16: end for
```

the number of the virtual nodes that are assigned to it. So, the stress $S_N(kp, i)$ of a path p_{kp} is the summation of the stresses of all of the nodes that belong to this path. For a substrate link e the stress $S_L(e)$ is the number of virtual links that have been mapped to it.

This algorithm aims to perform load balancing across the network by attempting to minimize the stress of the nodes and the links. A *one-to-one* mapping is assumed; a virtual network element is mapped to a single substrate network element. The exact steps of this algorithm can be seen in Alg. 4.

4.3.6 Randomized mobility agnostic heuristic algorithm

Lastly, a virtual network embedding algorithm which assigns paths in a completely randomized way is implemented. This means that for every request, regardless the

Algorithm 5: Randomized Mobility Agnostic Heuristic Algorithm

INPUTS: Graph $G = (V, E)$, number of tenants U, virtual network requests Q, classes of flows K, set of paths P, set of demands d, set of capacities C
OUTPUTS: Set of mapped substrate paths for every virtual network request

```
1: for u = 1 to U do
2:   for q = 1 to Q do
3:     for k = 1 to K do
4:       repeat
5:         p = randi(1 : P)
6:         if path π_kp not congested then
7:           map path π_kp to request q
8:           decrease available capacity of path π_kp
9:           flag = 1
10:        end if
11:      until flag = 1
12:      flag = 0
13:    end for
14:  end for
15: end for
```

class U, the algorithm forms the network by choosing randomly routing paths among the set of the available paths. This approach does not perform any optimisation strategy and it does not take the users mobility effect into account. The detailed steps can be seen in Alg. 5. In addition it doesn't perform any kind of tenants classification.

Is has to be noted that if the capacity of the substrate network is much higher than the total demand of the flows, then this algorithm, because of the fact that it uses no intelligence, is much faster to be solved and there is a high probability of avoiding the rejection of the virtual network requests.

Table 4.1: Simulation Parameters

Parameter	**Value**
Network nodes (V)	31
Types of services (U)	2
Requests per service (Q)	2
Flows per request (K)	16
Alternative paths (P)	5
Alternative paths (R)	5
Flow demand (d)	1
Node capacity ((C)	100
Probability of one-hop handover (s)	70%
Probability of two-hop handover (w)	40%

4.4 Performance Evaluation

In this section the performance of the above presented virtual network embedding algorithms is evaluated. The main purpose is to observe the effect of aggregated end-user mobility and how each one of the presented algorithms performs. Then, the performance of the proposed virtual network embedding mobility aware algorithm is compared to the rest of the previously presented algorithms.

4.4.1 Distributed Mobility Management scheme

At the first part of the simulations the performance of the virtual network embedding algorithms in cases where the available physical resources are sufficient to afford all of the requested traffic is evaluated . The parameters of the simulation scenario can be seen in the provided simulation parameters table. The simulation assumptions comply with an average LTE-A network's metrics; the assumed number of nodes, which is the main bottleneck for the algorithm's computational time, is close to an average MNO core network's size. In order to solve the proposed integer programming optimisation algorithm, MATLAB's optimisation toolbox was used. The rest of the algorithms were implemented using MATLAB as well.

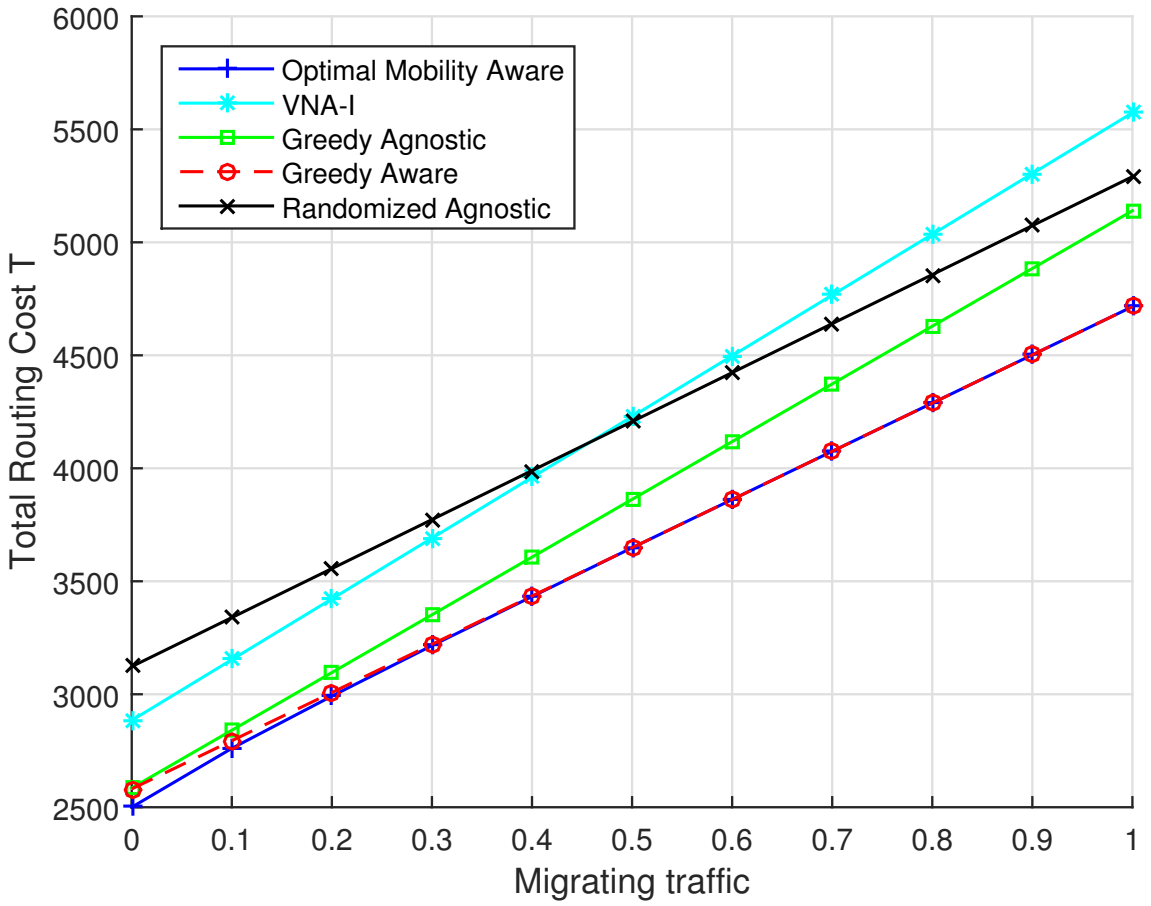

Figure 4.2: Total cost as the mobility increases

Total routing cost T

In Fig. 4.2 the overall performance of the aforementioned algorithms as the total ratio of migrating traffic due to mobility to the total traffic increases can be seen. In particular, there is a comparison of the total cost as it was calculated in the objective function (4.6) of the optimisation framework.

The two mobility aware algorithms outperform the rest of the algorithms in this low congestion scenario, achieving the same total cost. It has to be noted that the greedy mobility agnostic algorithm has significantly lower performance, up to 10% and the VNA-I along with the randomized agnostic algorithm achieve the worst total cost. As it was expected, since the network is not heavy loaded, the greedy

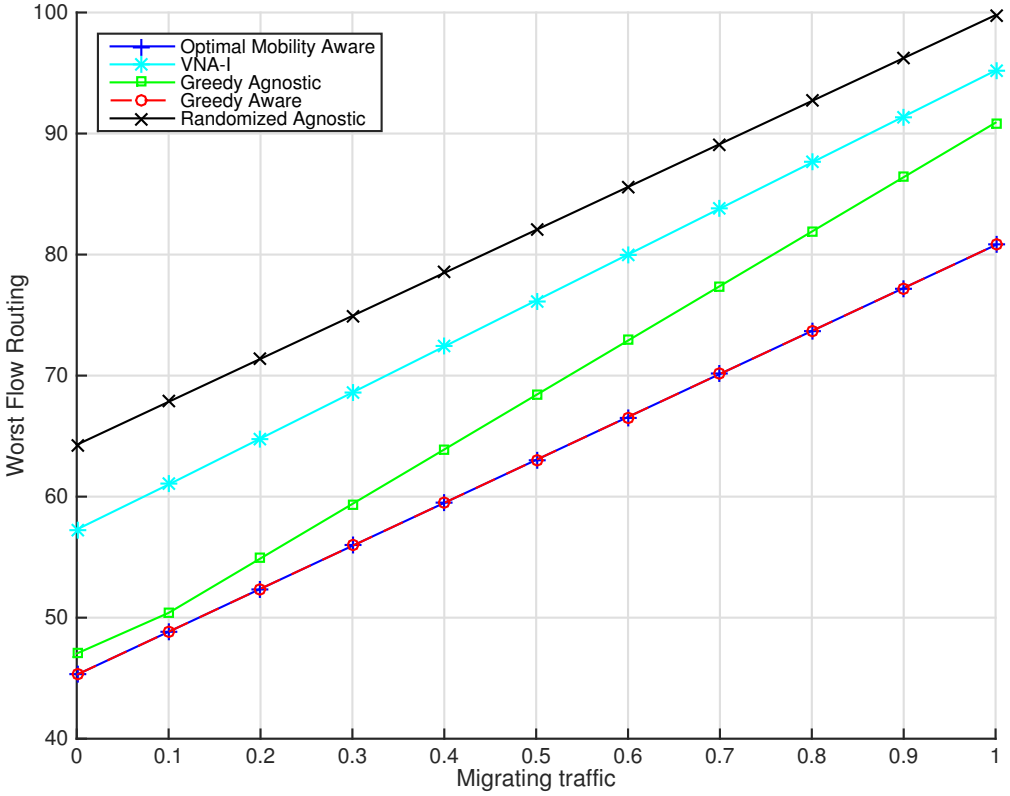

Figure 4.3: Worst flow cost as the mobility increases

algorithm manages to embed firstly the best paths and for this reason it has the same performance with the optimisation algorithm that provides optimal solutions.

In Fig. 4.3 there is a presentation of the worst routing cost of all the flows against the percentage of the mobility for each one of the algorithms. A linear behaviour can be observed as with respect to the total cost. In this scenario as well, the mobility aware algorithms have the least worst case performance, whereas the randomized algorithm along with the VNA-I achieve the worst behaviour, more than 20% worse than the optimal solution.

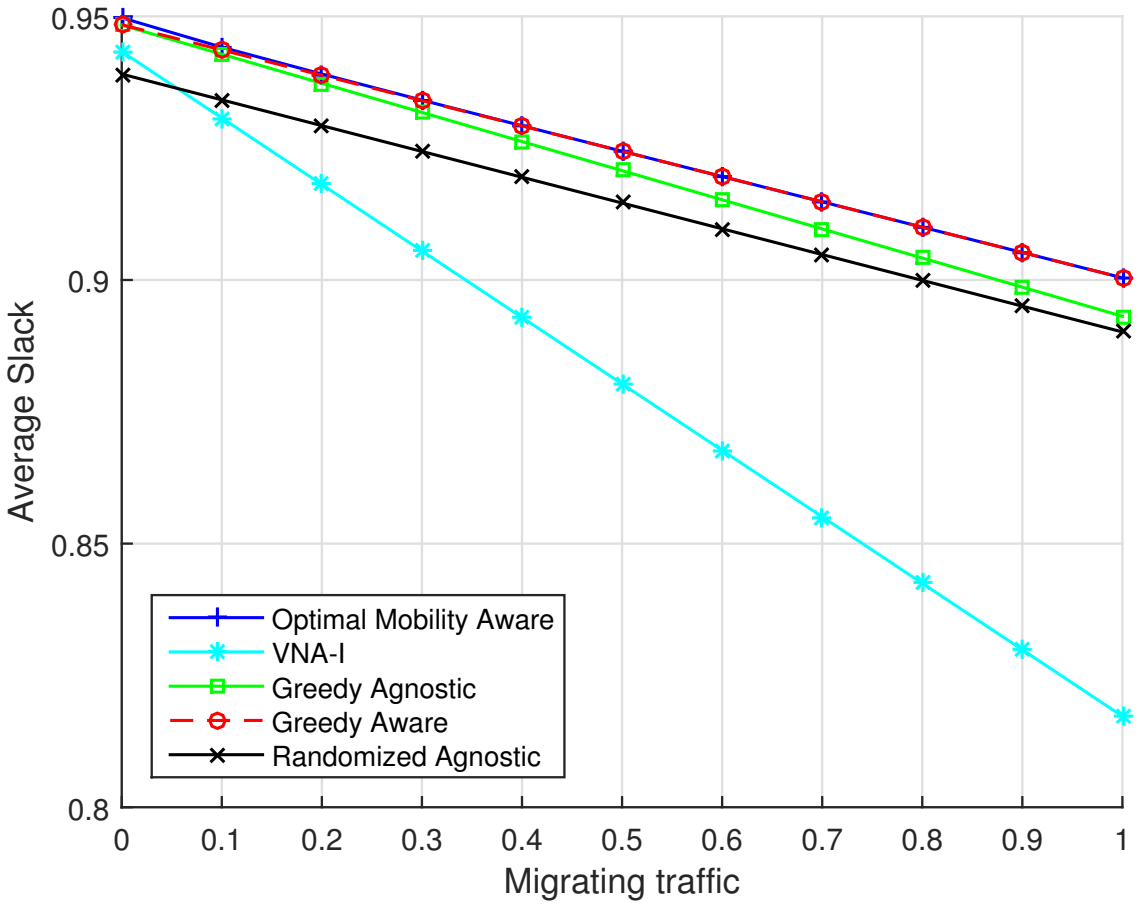

Figure 4.4: Average slack

Available node capacity slack and node stress

An important metric for network operators that liaise their physical network infrastructure is the remaining available capacity after the virtual network embedding has been performed; this relates to the additional traffic that the network can accommodate. To this end, Fig. 4.4 depicts the average node capacity slack in the network.

The mobility aware algorithms achieve the biggest average available slack and the rest of the algorithms achieve almost the same performance except for the VNA-I where the average node capacity decreases linearly with the mobility percentage. The same behaviour is observed also for the average node stress as shown in Fig. 4.5.

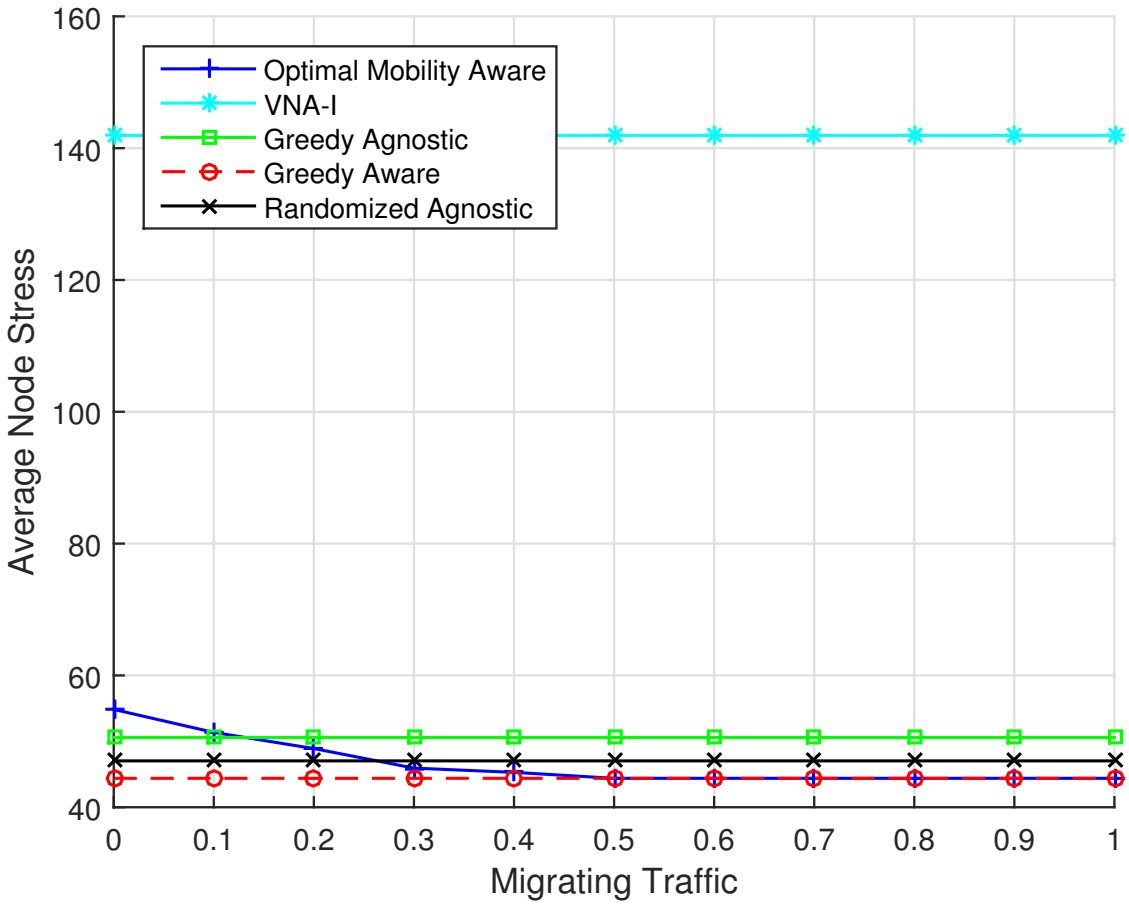

Figure 4.5: Average node stress

4.4.2 Centralized Mobility Management scheme

After evaluating the performance of the presented algorithms, the main focus is given on studying the effect of the mobility management scheme. As already explained, in the above system model a distributed mobility management scheme is assumed. Hence, the mobility function is placed at the very edge of the network on every edge router.

Without altering the algorithms' mapping strategy, the effect of a centralised mobility management scheme is now considered. This means that the mobility function, i.e., mobility anchoring, is placed hierarchically higher in the network. In this case and in contrast to the distributed mobility management scheme, less mobility an-

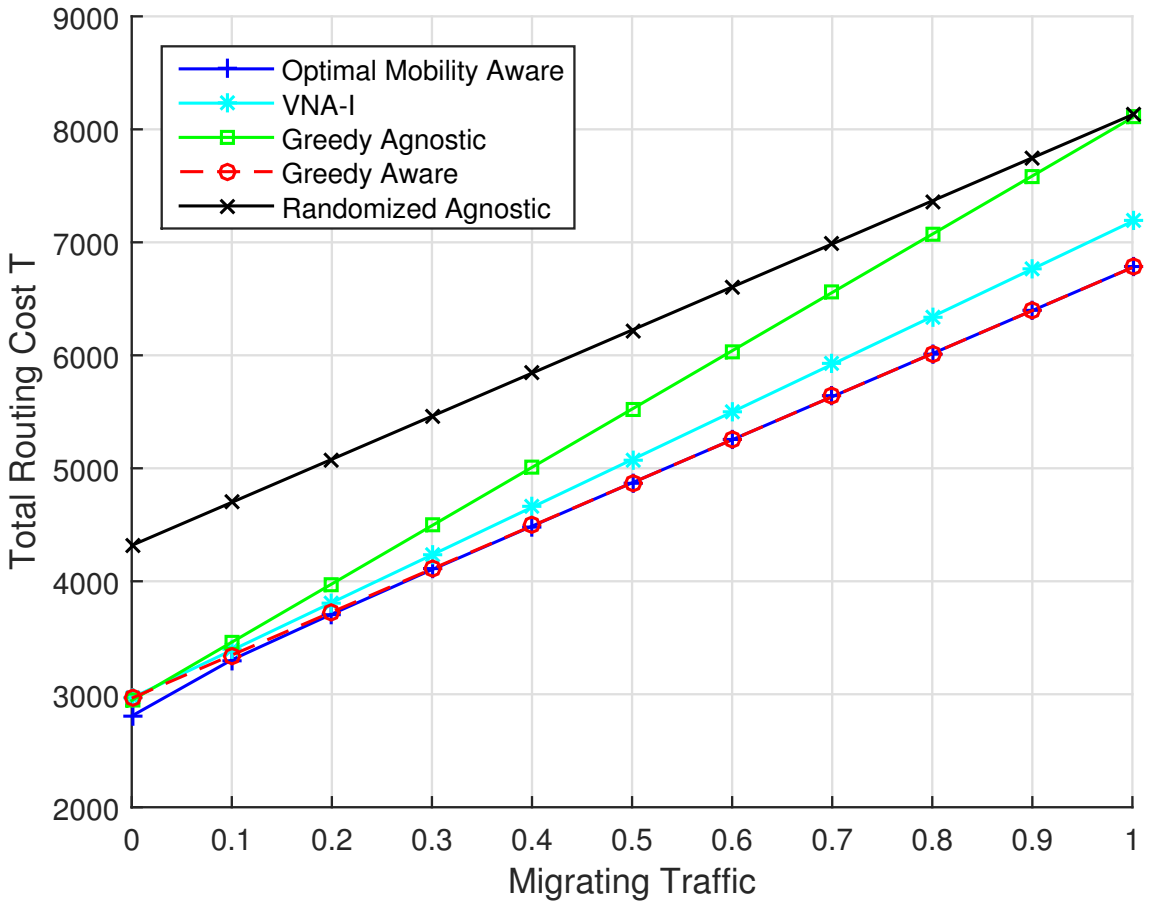

Figure 4.6: Total cost as the mobility increases

chor points serve the scope network domain. Every flow has to be directed through a mobility anchor point and during a handover it will be anchored at this point.

Without loss of generality for the centralized mobility management it is considered that a Proxy Mobile IP (PMIPv6) solution is used. PMIPv6 is a network-based mobility solution, where in contrast to the previous mobility management schemes there is no need to install any-mobility related software on the mobile equipment, while the mobility management is assigned to specific network entities. More precisely, Mobility Access Gateway (MAG) is a network entity that is responsible for tracking mobile's location and creating a tunnel with the other basic network entity, Local Mobility Anchor (LMA). LMA, which is an enhanced version of MIPv4s home agent, is mainly responsible for ensuring the reachability of the MNs address, while

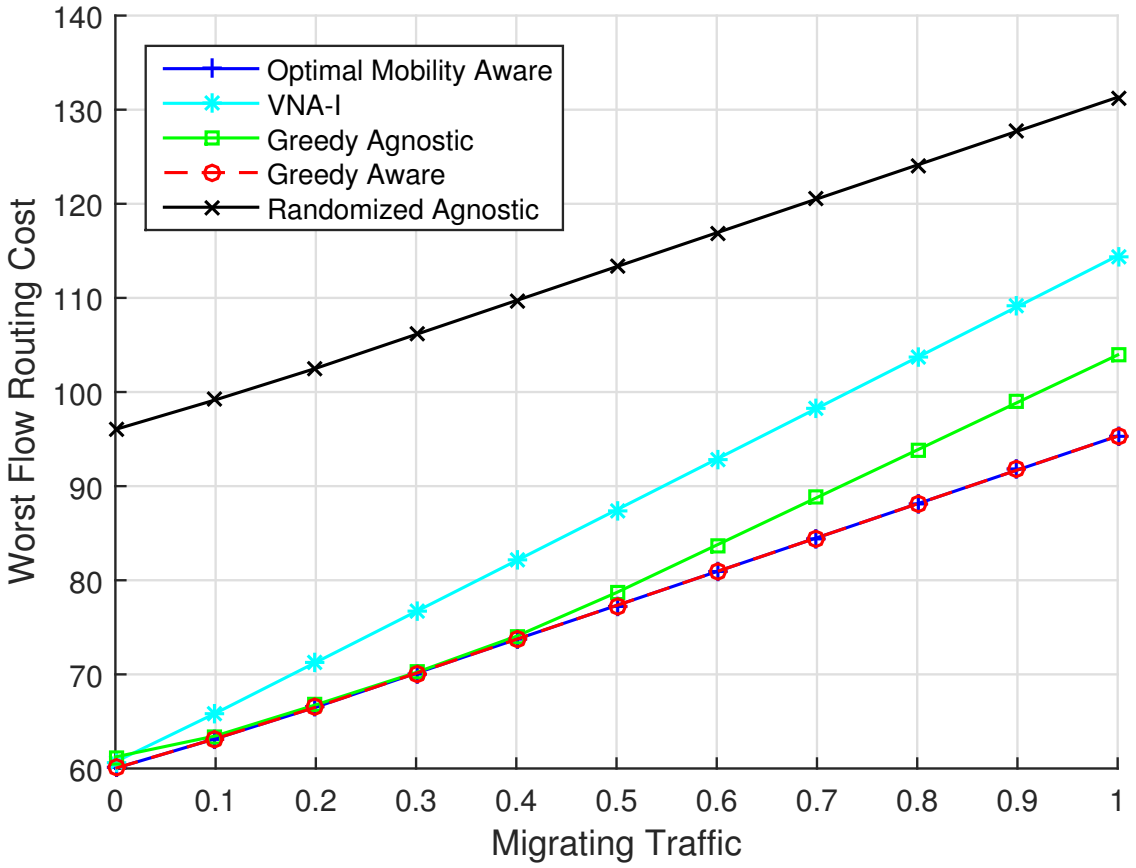

Figure 4.7: Worst flow cost as the mobility increases

it moves within a PMIPv6 domain.

When a MN is located inside a PMIPv6 domain, MAG obtains MNs profile and then it sends a Proxy Binding Update (PBU) to LMA, which after an authentication process sends back a Proxy Binding Acknowledgment (PBA) and sets up a route for the MNs home network prefix using the tunnel with the MAG. After receiving the PBA, MAG is able to emulate the MNs home network can send a Router Advertisement (RA) message to MN in order for it to configure its home prefix accordingly. At this moment, all data will be forwarded via this MAG-LMA tunnel, saving bandwidth from the MN by excluding it from mobility-related signalling (so, the MN is unaware of the mobility support procedure).

For the same topology as in the previous simulations 4 mobility anchor points that

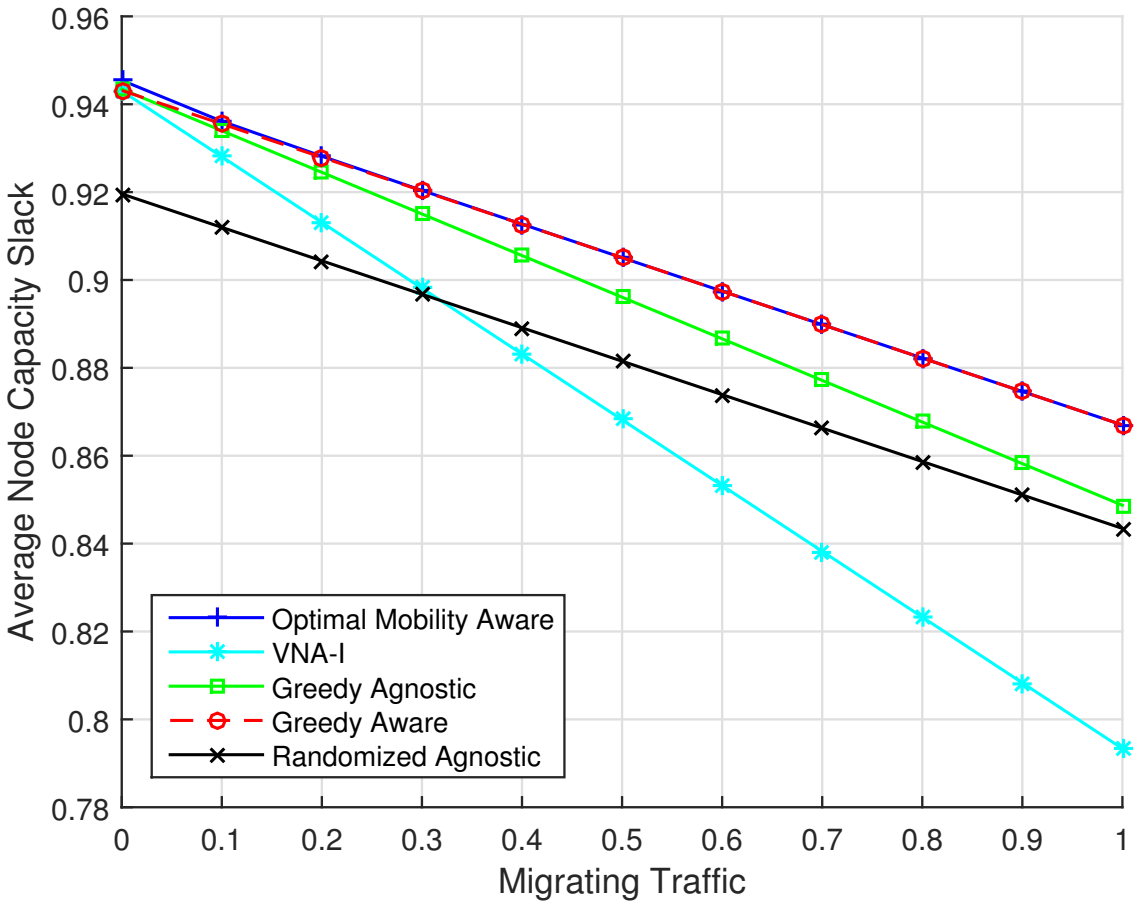

Figure 4.8: Average slack

each serve 4 destination edge routers are co-located. Every path π_{kp} has to include the mobility anchor point that serves the edge router k and every path r_{kji} is now the path that connects the mobility anchor point with the new edge router j where the mobile equipment has migrated to.

Routing cost

In the scenario where CMM is in use, the total cost as shown in Fig. 4.6 and the worst routing of a single flow as shown in Fig. 4.7, follow the same trend with the previous scenario where DMM was used. However, a 25% increase of the cost is observed and that is expected since now the centralised mobility management directs the flows through non-optimal paths. Hence, the network is becoming more loaded at these

parts of the network and the embedding of virtual networks results in an increased use of resources.

Average node capacity slack and average node stress.

The same behaviour holds for the performance in terms of average and minimum node capacity slack as well as the average node stress where the randomized algorithm and the VNA-I are outperformed by the rest of the algorithms. Again, the mobility aware algorithms achieve the best performance compared to the rest of the solutions. The results can be seen in Fig. 4.8 (average slack), Fig. 4.9 (minimum slack) and Fig. 4.10 (average node stress).

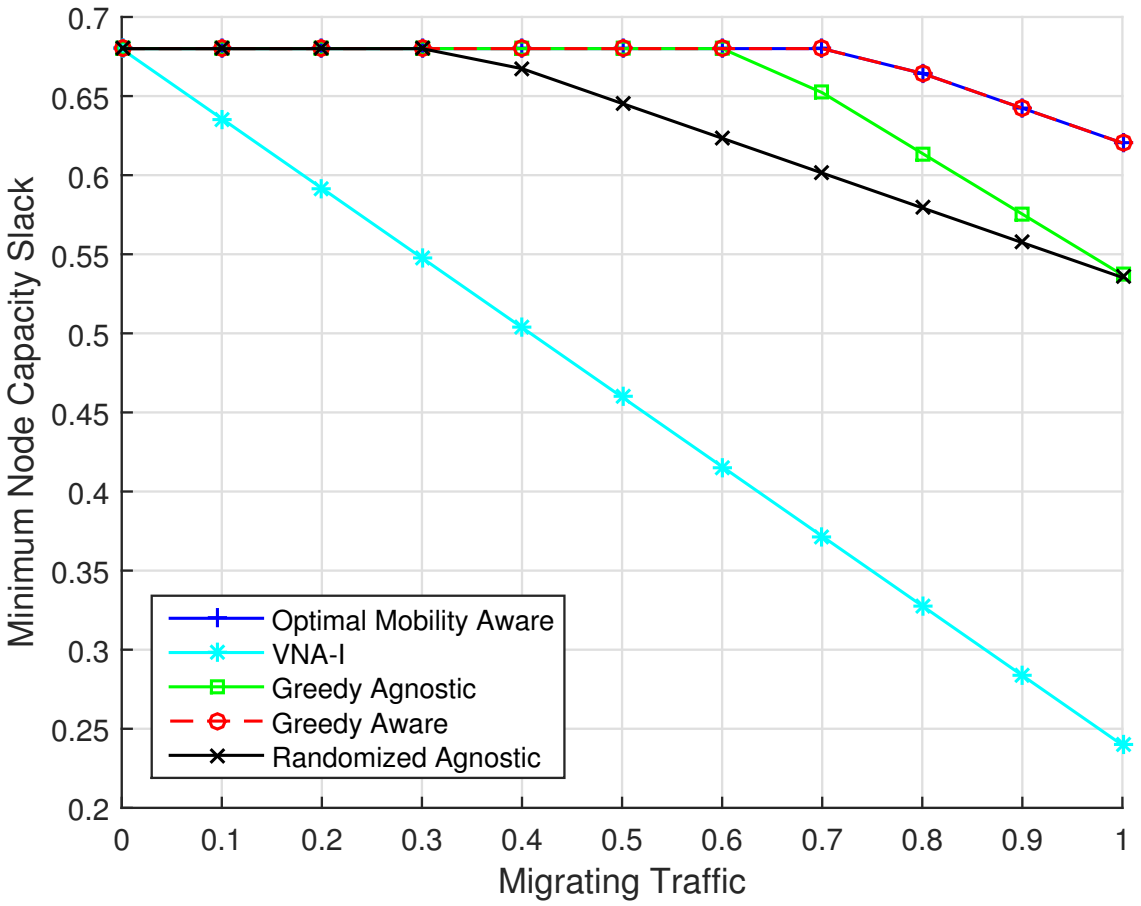

Figure 4.9: Minimum node slack

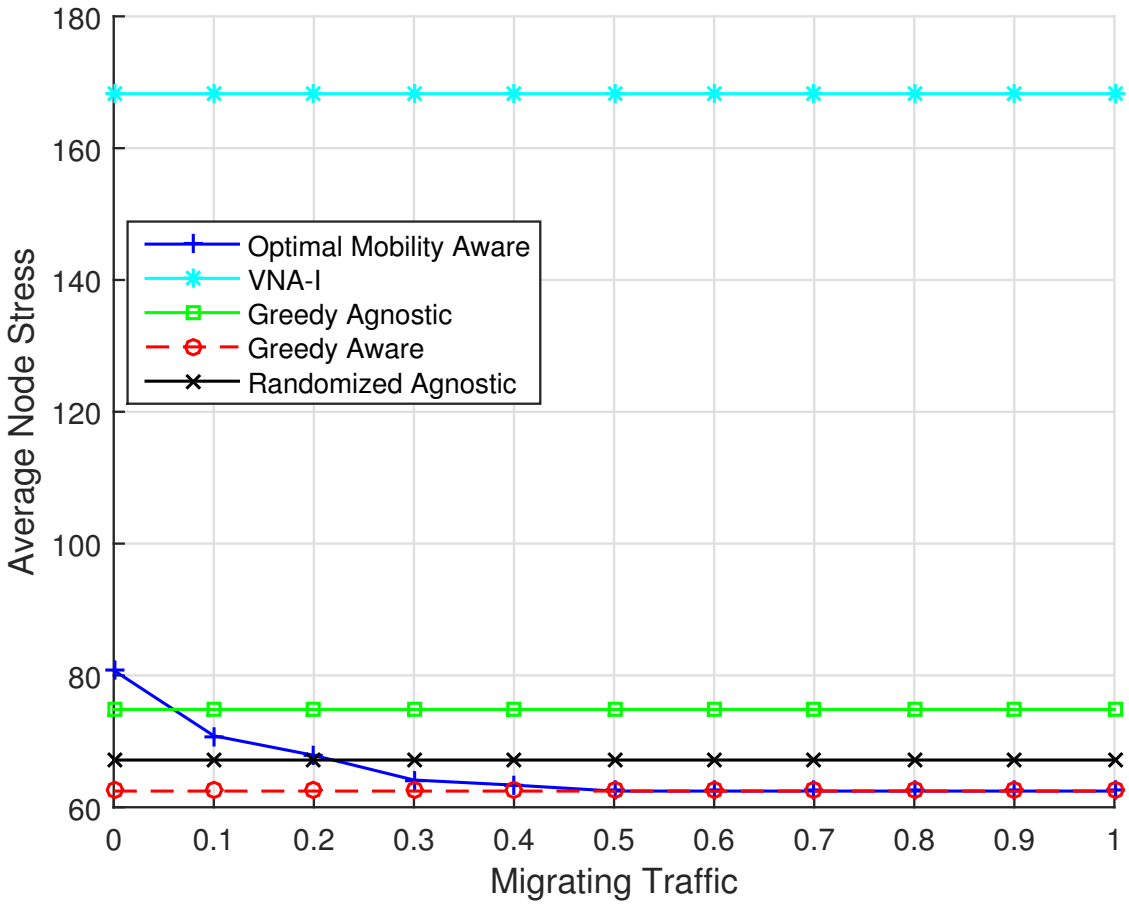

Figure 4.10: Average node stress

4.4.3 Complexity and optimality gap

As already been eluded above, the performance of the mobility aware optimisation framework and the mobility aware greedy heuristic algorithm is the same for a low congestion scenario. The reason is that both of the algorithms load the shortest paths that can hold all the demanded traffic.

However, in terms of complexity, since the optimisation algorithm is a Mixed-Integer Linear Programming (MILP), it can be solved in non-polynomial time as it is a well known NP-hard problem. Due to its intractable nature, when the network's topology grows, the proposed optimisation algorithm cannot be used. However, from the previously presented results it can be seen that the proposed greedy mobility

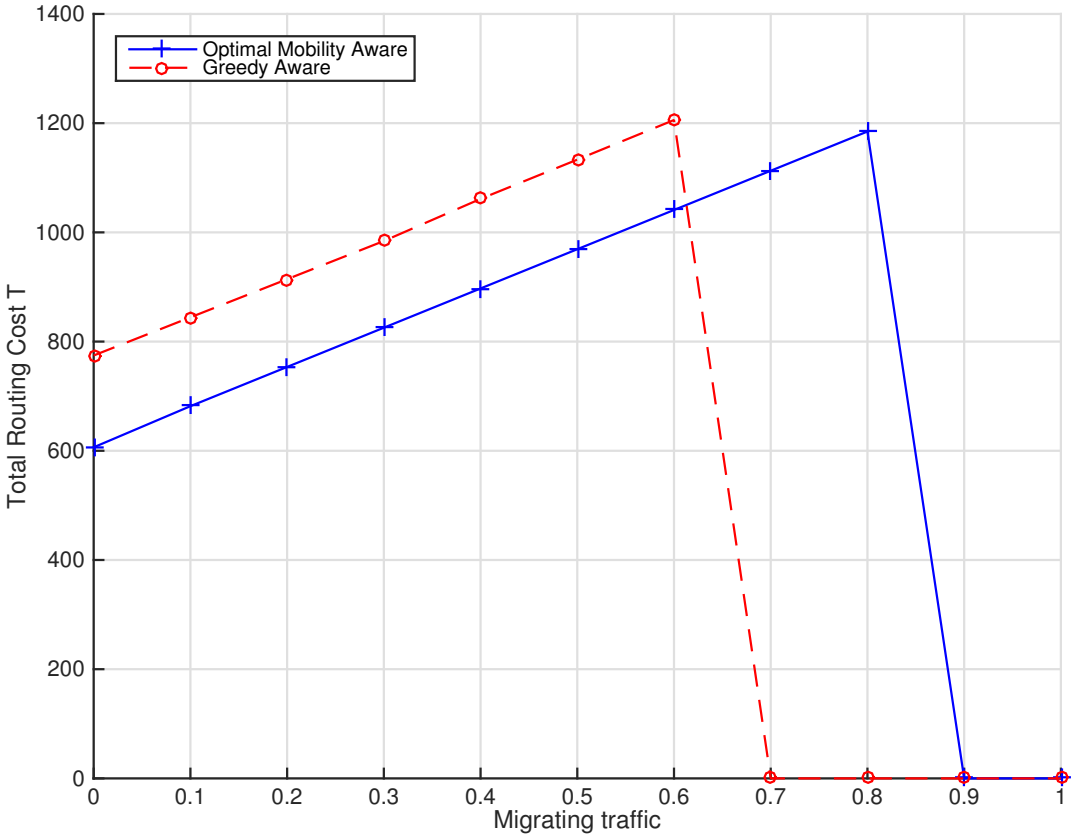

Figure 4.11: Optimality gap between optimisation and greedy algorithm.

aware algorithm can be used with a very competitive performance for efficient virtual network embedding.

As is is evident, there is an inherent trade-off between low computational complexity and optimality gap. The embedding problem as it was expressed in this paper could be reformed into a bin-packing problem while the mobility aware greedy algorithm is a form of a next-fit algorithm. According to [55] the optimality gap of the heuristic algorithm can be up to $\times 2$ worst than the optimal solution. For this scenario, the optimality gap depends highly on the selection of the routing paths.

As a last part of the simulation based analysis, the performance of the two mobility aware algorithms in a high traffic scenario where the topology as well as the available resources may increase the optimality gap, is compared to the greedy algorithm and

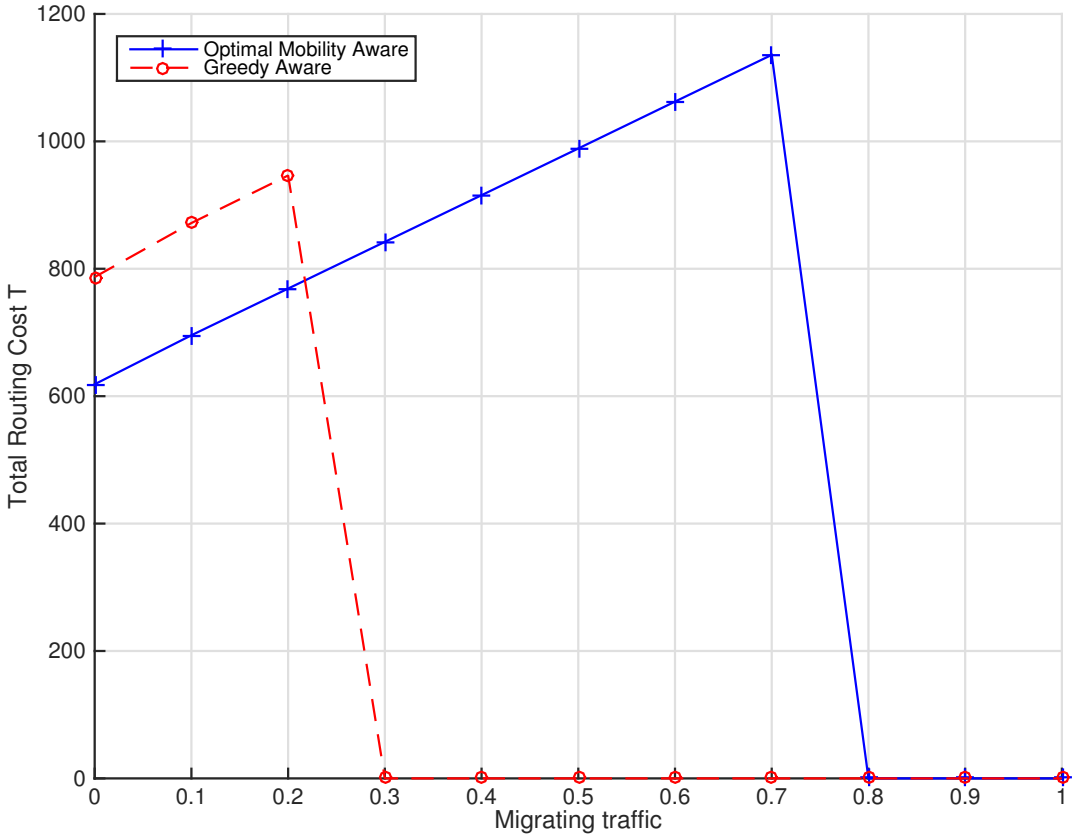

Figure 4.12: Optimality gap between optimisation and greedy algorithm.

the optimisation scheme.

In order to outline that the optimisation algorithm potentially outperforms the greedy mobility aware algorithm, a high congestion episode scenario is assumed, where the trivial embedding of the shortest paths in a greedy way does not overlap with the optimal solution.

In this scenario the average node capacity value is set to 10 and the resources' utilisation is set to near 1. As Fig. 4.11 shows, the greedy algorithm has 33% worse performance in terms of total cost T as the migrating traffic due to mobility effect increases. Also, as the congestion increases furthermore due to the migration of the traffic from the mobility effect, the greedy algorithm is no longer able to map the virtual network requests (the zero values mean that no virtual network request was

satisfied).

In Fig. 4.12 the average node capacity is slightly decreased. This resulted in higher network infrastructure utilisation and heavier traffic load. In that case it can be observed that the greedy algorithm fails to embed the requested networks at the point of less than half user traffic in comparison to the optimisation algorithm.

Depending on the network topology and the network utilisation the greedy mobility aware algorithm can potentially be outperformed by the optimisation embedding algorithm. However, as it was shown in the previous scenario, when the demand is very low then the greedy strategy maps the virtual networks in the optimal way.

4.5 Publications

1. G. Chochlidakis and V. Friderikos, "Mobility Aware Virtual Network Embedding", in *IEEE International Conference on Communications - Communications QoS, Reliability and Modelling Symposium (ICC15 CQRM), London, United Kingdom*, Jun. 2015 pp. 58655871.

2. G. Chochlidakis and V. Friderikos, "Mobility Aware Virtual Network Embedding", *IEEE Transactions on Mobile Computing* , June 2016, DOI: 10.1109 \TMC.2016.2591525 , pp. 1343 - 1356.

Chapter 5

Low Latency Virtual Network Embedding

Network virtualisation has become one of the most prominent solutions that can efficiently deal with the dramatic increase of data demand in mobile networks. In order to allow multiple virtual networks to coexist in the same substrate network, the need for development of efficient virtual network embedding algorithms and techniques is imperative. The main purpose of this chapter is to provide an optimisation framework for virtual network embedding that minimizes the end-to-end delay. The proposed scheme takes into account the actual user mobility effect in order to allow efficient mapping for mobile networks. In addition, it provides service differentiation allowing delay sensitive services to use the formed virtual networks with the minimum possible delay in comparison to elastic services. The performance of the proposed algorithm is evaluated and compared to existing optimal shortest path virtual network embedding algorithms. The numerical results reveal that the proposed algorithm converges to an optimal solution and its performance can achieve significant improvement in comparison to shortest path optimisation algorithms.

5.1 Introduction

During the last years, mainly due to the best-effort and drop-tail based packet processing in IP networks, it is a widely accepted tenet of modern network economics that staying ahead of demand can only take place via a suitable network over-provisioning. This orthodox way of network operation is starting to raise real signs of concerns for two intertwined reasons. The first, and well documented one, is the steadily, almost exponential, traffic increase that network operators are facing. This firm increase of data traffic is leading to unsustainable network operation especially because revenues from mobile users is reaching a plateau [41]. Secondly, it is now widely accepted that we are at the dawn of new and emerging types of network traffic related to Machine-to-Machine (M2M) communications and haptic-based applications that will require very low latency communications in the near future. Among all, Network Virtualisation (NV) is emerging as a key technology to ease the pressure coming from the above two challenges.

The virtualisation of the available physical resources (i.e., the so-called substrate network) has been considered as a promising solution for next generation mobile networks, including 5G, in order to achieve efficient and on-demand network sharing. It is worth pointing out that some form of passive (i.e. non-adaptive) physical infrastructure sharing is currently used within the cellular operators. Current predictions point out that by the end of 2015, 90% of mobile operators will have explored this avenue in some form[1].

NV will in essence enable dynamic network sharing and hence propel further operational and capital cost reduction by increasing the utilisation of the currently deployed network infrastructures (substrate networks). With respect to low latency communications, the construction of virtual networks that explicitly consider delay will undoubtedly play an important role towards the Tactile Internet vision.

[1]Mobile Network Sharing Report Developments, www.reportlinker.com

The main advantages of multi-sharing of physical network infrastructures, in summary, are the increase of utilisation of the available physical resources, thus raising the energy efficiency, the improved flexibility, manageability and scalability that it offers and finally the prospect of robustness for mobile operators [43]. Clearly, the potential advantages of NV are such that this trend is expected to be more and more adopted in the future. Moreover, network sharing is going to move deeper into the network becoming more dynamic and flexible, able to allow multi-tenancy schemes at different network elements within the core and wireless access network. Although NV has recently been considered for mobile networks, the research interest is significant during the last years. An example of the implementation of NV in LTE-A mobile networks can be seen in [56] [57].

The main motivation behind NV is the sharing of a single physical infrastructure by multiple tenants. The procedure of the efficient mapping of virtual networks on the substrate network is called Virtual Network Embedding (VNE). On this area, significant research has been conducted so far. An extended survey of the related work on this area is given in [23]. The main taxonomy of the current embedding algorithms is conducted along three distinct dimensions: static and dynamic, centralized and distributed, concise and redundant.

Low latency communications is gradually gaining an increasing attention as a key requirement in emerging and future (5G) wireless networks since it allows the implementation of haptic based applications with tactile latencies [58]. Network provisioning for low latency applications on the order of single digit delay (in milliseconds) will unlock the potential of a large number of applications ranging from autonomous robots, to healthcare and energy sectors. In that respect, the effect of the delay, although a very important network metric, has not yet been considered in the area of VNE for mobile networks.

In this chapter a VNE optimisation algorithm is proposed that minimizes the end-

to-end delay, allows service differentiation and takes into account the user mobility effect [59]. In order to deal with the non-linearity of the delay variable, the problem is decomposed and solved using the projected subgradient method. The convergence of the proposed scheme is confirmed by computational simulations.

5.2 Previous related work

Different approaches have been proposed so far to address the VNE problem. In this chapter, selected related works from the literature are presented.

The authors in [45] model the problem of VNE as a mixed integer programming optimisation problem. Then, they present mapping algorithms after introducing a coordination between node and link mapping phases. In order to do so, they transform the mixed integer programming into linear by using deterministic and randomized rounding techniques and in this way the problem's sub-optimal solution is achieved in polynomial-time. The simulation results show that their algorithms outperform the so far existing approaches in terms of acceptance ratio, revenue and provisioning cost.

The authors in [46], also, propose a mixed integer programming optimisation algorithm that aims to minimize the total resource consumption and to perform on-line load balancing across the network. To this end, they develop three cost functions: one that minimizes the total load on every virtual network, one that minimizes the total number of links that are embedded and maps nodes that have more available resources and finally a cost function that includes the demanded capacity by the virtual network requests in the objective function. Their proposal, according to conducted simulations, achieves better overall performance compared to heuristic algorithms.

The authors of [51], motivated by the fact that previous VNE algorithms are designed for static networks, propose an integer programming optimisation algorithm that aims to minimize the upgrading cost of virtual networks as the network evolves, with respect to node resource and path delay constraints. Although delay is considered, because of the problem's complexity, the optimisation algorithm is not solved, so they develop a heuristic algorithm instead and they present its efficiency through simulations.

In [60] the authors propose a VNE algorithm that aims to map virtual links onto substrate paths with minimum bandwidth cost while retaining the delay constraints. In the problem modelling, the assumption that the queuing delay is directly proportional to the flow rate is used. The proposal outperforms a mapping algorithm from the literature. In [61], the authors also assume predefined constant delays associated to the arcs of the substrate network. Then, they propose the Virtual Network Mapping Problem with Delay, Routing and Location Constraints (VNMP-DRL) and they formulate it as a mixed linear optimisation problem.

The authors in [62] formally define the VNE problem for multicast virtual networks and prove its NP-hard nature. Then they propose two novel approaches to solve the multicast embedding problem with end-delay and delay variation constraints: a three-step multicast embedding technique, and a Tabu-Search algorithm. Their results prove the competitiveness of their proposals.

In [63] the authors provide a distributed protocol that provably converges to the optimum multipath routing, achieving at the same time optimal load balancing as well as increased robustness. By carrying out packet level simulations with realistic topologies, feedback delays, link capacities, and traffic loads, they show that the distributed protocol is adaptive and sufficiently robust. Lastly, in the *DaVinci* [64] architecture, each substrate link periodically reassigns bandwidth shares between its virtual links; while at a smaller time-scale, each virtual network runs a distributed

protocol that maximizes its own performance objective independently. Concerning both aforementioned works, a very good approximation of the delay function is being considered and the optimal solutions are provided.

In all the above presented previous works the effect of users mobility combined with end-to-end delay has not yet been considered for wireless networks. As shown in the sequel, these are two important parameters that need to be taken into account for creating efficient virtual networks.

5.3 System Model Description

In the hereafter presented system model, a central controller that is responsible for slicing the physical resources and ensuring isolation among them is considered. In particular, the controller is handled by the VNE algorithm and it serves the requests with the corresponding strategy (i.e. minimisation of the delay, minimisation of routing cost etc.). Hence, the embedding algorithm creates a strategy for virtualisation of the physical resources and for their allocation to each one of the virtual networks according to the requests.

The controller, after having designed the plan for the slicing of the physical resources according to the predefined mapping strategy, communicates with the network elements in order to form the virtual networks. In this way, the virtual networks are set and become operational for the virtual network mobile operators. Each virtual network is fully independent and isolated from the others and has no awareness of their co-existence in the same substrate network.

Regarding the architecture of the implemented model, a single gateway and several destination edge routers are assumed. Each virtual network request is defined by demands from the source gateway to the end edge routers. Hence, the formed virtual

networks have a substrate topology dependence and should include the same source and destination nodes.

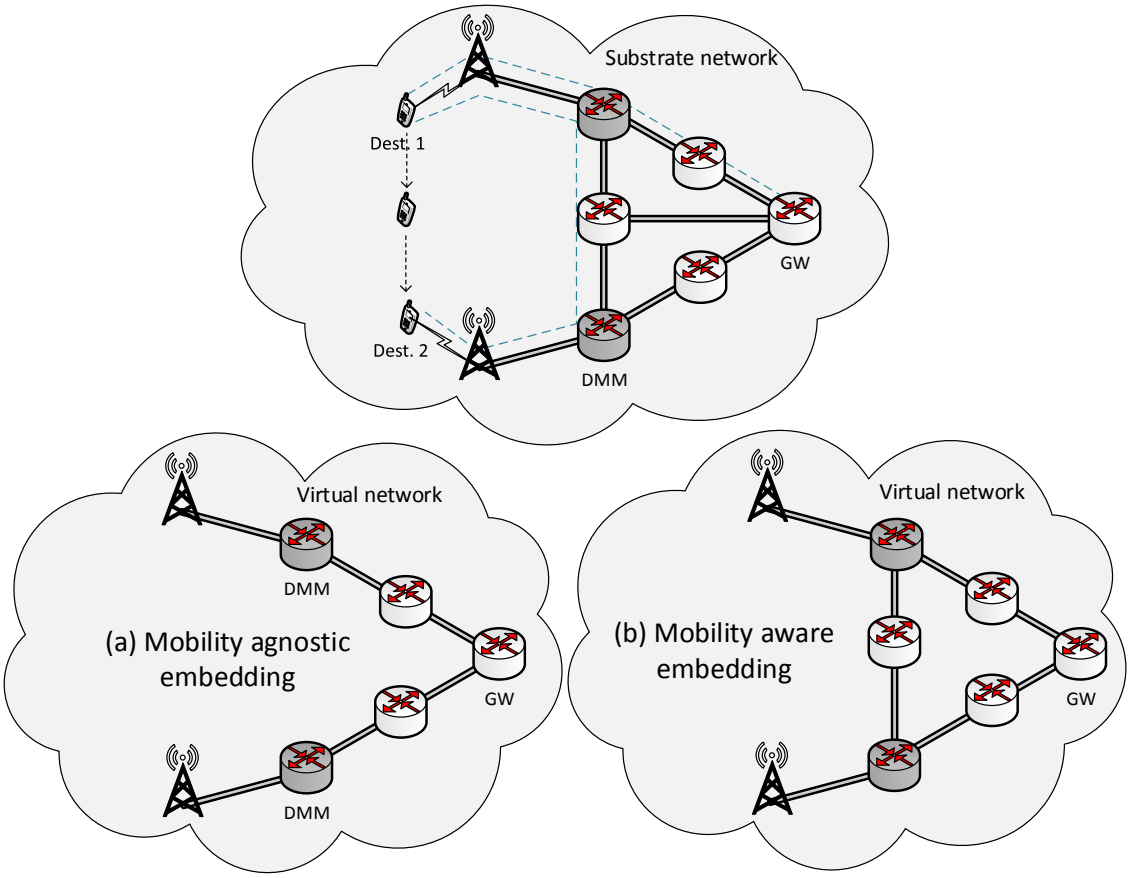

Figure 5.1: (a) Mobility agnostic and (b) mobility aware embedding algorithm.

As for the mobility management function, since distributed mobility management (DMM) scheme is deployed, every single destination edge router acts as a mobility management anchor point. As described in [9] , when a handover occurs, the flow anchors to the previous edge router and it gets forwarded to the next new destination anchor point.

An example of how mobility awareness in VNE can be implemented is illustrated in Fig. 5.1. As already detailed in the previous chapter, the main idea behind a mobility aware VNE algorithm [44], [65] is that, in contrast to a mobility unaware solution, it also maps intermediate paths for the data forwarding to the next cell after a handover, provisioning in this way the users' mobility. In this way, the virtual

networks become more efficient in terms of handling the handover and forwarding the data to the next edge router. However, this time, the main metric that will be monitored is not the number of hops but the latency that is proportional to the bandwidth utilisation on the links.

Since the proposed algorithm is mobility aware, it will embed the optimal paths that connect those edge routers according to a deterministic handover matrix. Therefore, it is considered that by applying Software Defined Networking (SDN) and Network Function Virtualisation (NFV) deployment, the mobility function is virtualised by being co-located at the edge of the core network (i.e. the edge routers).

5.3.1 Minimum delay virtual network embedding algorithm

Substrate network

The substrate mobile core network can be modelled as an undirected planar hierarchical graph $G = (Q, L)$. Q represents the set of nodes and L depicts the set of links. Let $C : L \rightarrow \mathbb{R}^+$ be the capacity of each link. The network gateway is located on the source node of the graph that connects the core network to the Internet, while the end-nodes of the graph represent the destination edge routers where base stations are connected. The rest of the nodes are the intermediate routers of the core network. The algorithm that is presented hereafter assigns nodes and links in order to form the virtual networks upon the virtual network requests and with respect to capacity constraints.

Virtual network requests

In order to define the set of virtual network requests Q it is considered that each request $v \in V$ can be represented as a graph $G' = (Q', L')$ that has to be mapped

on the graph $G = (Q, L)$. Each graph $G' = (Q', L')$ is consisted of a set of pre-calculated paths $s \rightarrow k$ that connect the virtual gateway s with the virtual edge routers $k \in K$, where $K \subset Q$.

In order to capture the actual mobility of the users, a handover matrix $\boldsymbol{H}_{V \times K \times K}$ is defined, the elements $h_{vkj} \in (0, 1)$ of which represent the probability of the flows destined to edge router $k \in K$ to migrate to another edge router $j \in J$ (where $J \equiv K$) for the virtual network request $v \in V$. These data correspond to a specific time window where network performance is expected to follow an average trend. Is is also assumed that $h_{rkj} = 1$ for $k = j$ in order to map paths that first connect the edge routers. For the following simulations, handover up to two consecutive hops is assumed and for those pairs $\{k - j\}$ is assumed that $h_{vkj} \in (0, 0.4)$ uniformly. Let, also, $d^r_{vkj} \in D$ be the demand for virtual resources.

In this work and without loss of generality, it is assumed again that DMM anchoring takes place at the edge routers. While DMM scheme is in use, there is an additional need to assign paths that connect edge routers or network elements where DMM anchoring is taking place. This means that the graph $G' = (Q', L')$ also includes paths $\{k - j\}$ that connect directly the edge routers. It needs to be noted that there is a topology dependence of the substrate and the virtual networks in a way that the virtual gateway has to be mapped on the physical one and the virtual edge routers have to be also mapped on the corresponding physical edge routers.

Delay modelling

For each link $l \in L$ a capacity C_l is defined. In order to capture the propagation delay for a flow that uses this link, δ_l is set to be a constant number that represents this delay. In this work, as in [66], the M/M/1 queueing delay is used:

$$F_l(T_l, C_l) = \frac{1}{C_l - T_l}, \tag{5.1}$$

where T_l is the total traffic on the link l.

In order to simplify the hereafter presented mathematical programming formulation, a linear piecewise approximation of the hyperbolic function F is used as it was presented in (5.1). In this way the delay function for the link l is:

$$f_l(T_l, C_l) = g_l T_l + b_l, \forall T_l \in [n\omega, (n+1)\omega], \tag{5.2}$$

where $n \in \{0, 1...(\zeta - 1)\}$ and $\zeta \in \mathbb{N}^+$ is the accuracy (i.e. number of pieces) of the approximation. It can be easily observed via simulations that as the accuracy increases, there is a point after which using the approximation is as good as the original function. Moreover,

$$\omega = \frac{C_l}{a} \tag{5.3}$$

is the stepsize for each linear piece,

$$g_l = \frac{\frac{1}{C_l - (n+1)\omega} - \frac{1}{C_l - n\omega}}{\omega} \tag{5.4}$$

is the slope of the linear piece for $T_l \in [n\omega, (n+1)\omega]$,

$$b_l = \frac{1}{C_l - n\omega} - gn\omega. \tag{5.5}$$

Service differentiation

In order to provide differentiation in terms of end-to-end delay among the $R \in \mathbb{N}^+$ different services, the set of weights A is introduced. Let $a_r \in A$ be the weight for each type of service. This solution can form virtual networks upon request giving priority to some of them against others for explicitly low latency purposes.

Problem parameters and variables

Based on the above setting, the goal is to find the optimal selection of paths in order to minimize the weighted average delay and at the same time to achieve tenant differentiation. Below, there is a summary of the parameters used for the formulation of the linear mathematical optimisation and the introduction of the linear variables:

- $G = (Q, L)$: undirected planar tree-like graph,
- π_{kjp}: p^{th} alternative substrate path that connects: i) the source node with the edge router k if $k = j$ or, ii) the edge router k with the edge router j if $k \neq j$. For each pair $\{k - j\}$, there are $P \in \mathbb{N}^+$ alternative paths,
- $r \in R \subset \mathbb{N}^+$: set of services,
- $v \in V \subset \mathbb{N}^+$: set of virtual network requests,
- $K, J \subset Q$: set of edge routers (destination nodes),
- $d^r_{vkj} \in D$: demand for mapping resources for the service $r \in R$, of the virtual network request $v \in R$ on π_{kjp},
- $h_{vkj} \in \boldsymbol{H}_{V \times K \times K}$: mobility matrix,
- $a_r \in A$: weight of each service, and
- $z_{kjpl} = \begin{cases} 1, \text{ if link } l \in \pi_{kjp} \\ 0, \text{ otherwise} \end{cases}$.

The binary parameter z shows if a link l is part of a pre-calculated path π. One link can belong to multiple paths. Then, the linear variable $x^r_{vkjp} \in \boldsymbol{x}$ is defined, which is the rate of the resources that are mapped (valued for zero for no mapping case) on the substrate path π_{kjp} to satisfy the service $r \in R$ of the virtual network request $v \in V$.

Objective function and problem constraints

$$\Gamma(\boldsymbol{x}) = \sum_{rvkjp} a_r x^r_{vkjp} \sum_{l} z_{kjpl} \big(\delta_l + f_l(T_l, C_l)\big). \tag{5.6}$$

Based on the above definitions the mathematical program can be formulated as follows:

$$\min \Gamma(\boldsymbol{x}) \tag{5.7}$$

s.t.

$$\sum_{rvkjp} x^r_{vkjp} z_{kjpl} \leq C_l \quad \forall l \in L \tag{5.8}$$

$$\sum_{p} x^r_{vkjp} = h_{vkj} d^r_{vkj} \quad \forall r \in R, v \in V, k \in K, j \in J \tag{5.9}$$

$$x^r_{vkjp} \geq 0 \quad \forall r \in R, v \in V, k \in K, j \in J, p \in P \tag{5.10}$$

where constraint (5.8) makes sure that the capacity of link l is not violated, constraint (5.9) ensures that the total resources mapped in all the alternative paths P for a pair of nodes is equal to the desired demand and, lastly, (5.10) defines that the problem variables $x^r_{vkjp} \in \mathbb{R}_{\geq 0}$.

Optimisation problem decomposition

The objective function is convex, since the delay function (5.1) and its piecewise approximation (5.2) are convex. Consequently, the proposed optimisation problem (5.7) is convex as well, therefore a unique optimal solution exists that can be obtained in polynomial time. The Lagrangian function of this problem can be written as

follows:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{q}) = \Gamma(\boldsymbol{x}) + \sum_{l} \lambda_l (T_l - C_l) \\
&+ \sum_{rvkj} q^r_{vkj} \Big(h_{vkj} d_{vkj} - \sum_{p} x^r_{vkjp} \Big) = \\
&\sum_{rvkjp} a_r x^r_{vkjp} \Bigg(\sum_{l} z_{kjpl} \big(\delta_l + \frac{\lambda_l}{a_r} + f_l(T_l, C_l) \big) - q^r_{vkj} \Bigg) \\
&+ \sum_{rvkj} q^r_{vkj} h_{vkj} d^r_{vkj} - \sum_{l} \lambda_l C_l,
\end{aligned} \tag{5.11}$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier for the inequality constraint (5.8) and $\boldsymbol{q}$ for the equality constraints (5.9). Next, the partial derivatives of the Lagrangian function (5.11) with respect to the linear variable $\boldsymbol{x}$ and the dual variables $\boldsymbol{\lambda}, \boldsymbol{q}$ are given:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{q})}{\partial x^{r_o}_{v_o k_o j_o p_o}} = a_{r_o} \Bigg(\sum_{l} z_{k_o j_o p_o l} \Big(\delta_l + \frac{\lambda_l}{a_{r_o}} \\
&+ g_l \sum_{rvkjp} \big(\frac{a_r + a_{r_o}}{a_{r_o}} \big) z_{kjpl} x^r_{vkjp} + b_l \Big) - \frac{q^{r_o}_{v_o k_o j_o}}{a_{r_o}} \Bigg)
\end{aligned} \tag{5.12}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{q})}{\partial \lambda_l} = \sum_{rvkjp} x^r_{vkjp} z_{kjpl} - C_l \tag{5.13}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{q})}{\partial q^r_{vkj}} = h_{vkj} d^r_{vkj} - \sum_{p} x^r_{vkjp} \tag{5.14}$$

Projected subgradient method

Next, the projected subgradient method is performed for this optimisation problem. The feedback updates for the variables $\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{q}$:

$$x_o^{(\kappa+1)} = \left[x_o^{(\kappa)} - \beta_x^{\kappa} \left(\frac{\partial^2 \mathcal{L}}{\partial x_o^2} \right)^{-1} \frac{\partial \mathcal{L}}{\partial x_o} \right]^+, \tag{5.15}$$

where

$$\frac{\partial^2 \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{q})}{\partial (x^{r_o}_{v_o k_o j_o p_o})^2} = \sum_l 2 a_{r_o} (z_{k_o j_o p_o l})^2 g_l, \tag{5.16}$$

$$\lambda_l^{(\kappa+1)} = \left[\lambda_l^{(\kappa)} - \beta_l^{\kappa} \frac{\partial \mathcal{L}}{\partial \lambda_l} \right]^+, \tag{5.17}$$

$$q_o^{(\kappa+1)} = \left[q_o^{(\kappa)} - \beta_q^{\kappa} \frac{\partial \mathcal{L}}{\partial q_o} \right]^+, \tag{5.18}$$

$\forall x_o = x^{r_o}_{v_o k_o j_o p_o} \in \boldsymbol{x}$, $\forall \lambda_l \in \boldsymbol{\lambda}$, $\forall q_o = q^r_{vkj} \in \boldsymbol{q}$, with $[y]^+ = \mathbf{max}\{0, y\}$. The selection of the stepsizes cannot affect whether the subgradient method converges or not β [66]. However, the speed of convergence depends on the selection of the stepsizes. In the following simulations constant values $\beta_x, \beta_q, \beta_\lambda \in (0, 1)$ are used.

As already stated, the variable $\boldsymbol{x}$ at constraint (5.10) can only take non-negative values, and for this reason the projection $[x]^+$ is used on the set $\mathbb{R}_{\geq 0}$. However, in order to ensure that equality constraints (5.9) can never be violated, project the values of the $\boldsymbol{x}$ are also projected as calculated by (5.15) after every iteration κ. In particular, after each iteration κ, the current variable vector $\boldsymbol{x}$ is replaced with the

Algorithm 6: MINIMUM DELAY VNE ALGORITHM

```
Find initial vector x
while κ ≤ N do
    for r = 1 : R do
        for v = 1 : V do
            for k = 1 : K do
                for j = 1 : J do
                    Find q_vkj^{r(κ+1)}
                    for p = 1 : P do
                        for l = 1 : L do
                            Find λ_l^{(κ+1)}
                        end
                        Find x_vkjp^{r(κ+1)}
                        Project x^{(κ+1)} on feasible space
                    end
                end
            end
        end
    end
    κ = κ + 1
end
```

$\boldsymbol{x}_{proj}$ as follows:

$$\boldsymbol{x}_{proj}^{\kappa} = \Big(\boldsymbol{I} - \boldsymbol{A}_{eq}^{T}(\boldsymbol{A}_{eq}\boldsymbol{A}_{eq}^{T})^{-1}\boldsymbol{A}_{eq}\Big)\boldsymbol{x}^{\kappa} + \boldsymbol{A}_{eq}^{T}(\boldsymbol{A}_{eq}\boldsymbol{A}_{eq}^{T})^{-1}\boldsymbol{b}_{eq}, \tag{5.19}$$

where $\boldsymbol{A}_{eq}$ and $\boldsymbol{b}_{eq}$ are the equality constraint matrices (where (5.9) is written in the form of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

The next step, before proceeding with the computing of the optimal $\boldsymbol{x}^{*}$ is the definition of an initial set of values for the vector $\boldsymbol{x}$. For this reason, firstly the optimal VNE is formed in terms of minimum routing cost as presented in the next section, in (5.21).

The steps of the proposed minimum delay embedding algorithm, for N iterations can be summarized as follows:

5.3.2 Optimal minimum routing cost virtual network embedding

In this section, an optimisation algorithm that optimally maps virtual networks in terms of minimum routing cost is presented, similar to the previous chapter's solution. Firstly, the total routing cost B is defined as follows:

$$B(\boldsymbol{x}) = \sum_{rvkjp} a_r x^r_{vkjp} \mu_{kjp} \tag{5.20}$$

where μ_{kjp} is the routing cost for each path π_{kjp}. Based on the above definition, the linear programming optimisation problem can be formulated as follows:

$$\min B(\boldsymbol{x}) \tag{5.21}$$

s.t.

$$\sum_{rvkjp} x^r_{vkjp} z_{kjpl} \leq C_l \quad \forall l \in L \tag{5.22}$$

$$\sum_{p} x^r_{vkjp} = h_{vkj} d^r_{vkj} \quad \forall r \in R, v \in V, k \in K, j \in J \tag{5.23}$$

$$x^r_{vkjp} \geq 0 \quad \forall r \in R, v \in V, k \in K, j \in J, p \in P \tag{5.24}$$

The rest of the parameters are the same as in the previously presented algorithm. The constraints of this problem have also the same physical meaning as with the previous problem.

Regarding the procedure of the virtual networks set up, it is not required to consider it as a network function with on-line service requirements. This means that the proposed VNE mechanisms are envisaged to take place in a pseudo-real time way. In addition, in terms of network provisioning, a virtual network-enabled cloud environment may be considered; hence, some form of centralisation required for the

proposed optimisation problem can be available.

Furthermore, it needs to be noted that as more network functionalities with diverse set of characteristics and performance requirements move to such cloud-based environment, focus should be given to an efficient processing of the control plane. These issues clearly relate to the network performance as a whole but fall beyond the scope of this chapter.

5.4 Numerical Evaluation

In this section, the proposed algorithm is modelled using MATLAB, in order to prove its convergence and to compare it with the optimal shortest path embedding algorithm (5.21). First, the optimisation toolbox of MATLAB is used in order to solve the minimum routing cost embedding problem. Then, the solution is used as the initial set of values for the minimum delay embedding Alg. 6.

For the simulations, a dense binary tree with $Q = 15$ nodes , $K = 8$ destination edge routers and $L = 36$ links is used. Firstly, $V = 2$ virtual network requests with no service differentiation ($a_1 = a_2 = 1$) are formed. The demand d_{vkj} ranges uniformly in $\{0, 3\}$.

As it is shown in Fig. 5.2, the proposed algorithm manages to converge to a minimum value for the aforementioned parameter values, where no service differentiation is in use. With a step-size $\beta = 0.1$, the convergence is achieved after approximately 30 iterations. It can be seen that as the capacity of the links increases the optimality gap in comparison to the solution of the shortest path algorithm (iteration $\kappa = 1$) decreases. This means that in high utilisation cases the achieved improvement of the proposed algorithm increases. For this scenario, the proposed algorithm achieves up to 27% improvement ($C = 7.5$) in comparison to the shortest path solution (5.21). It

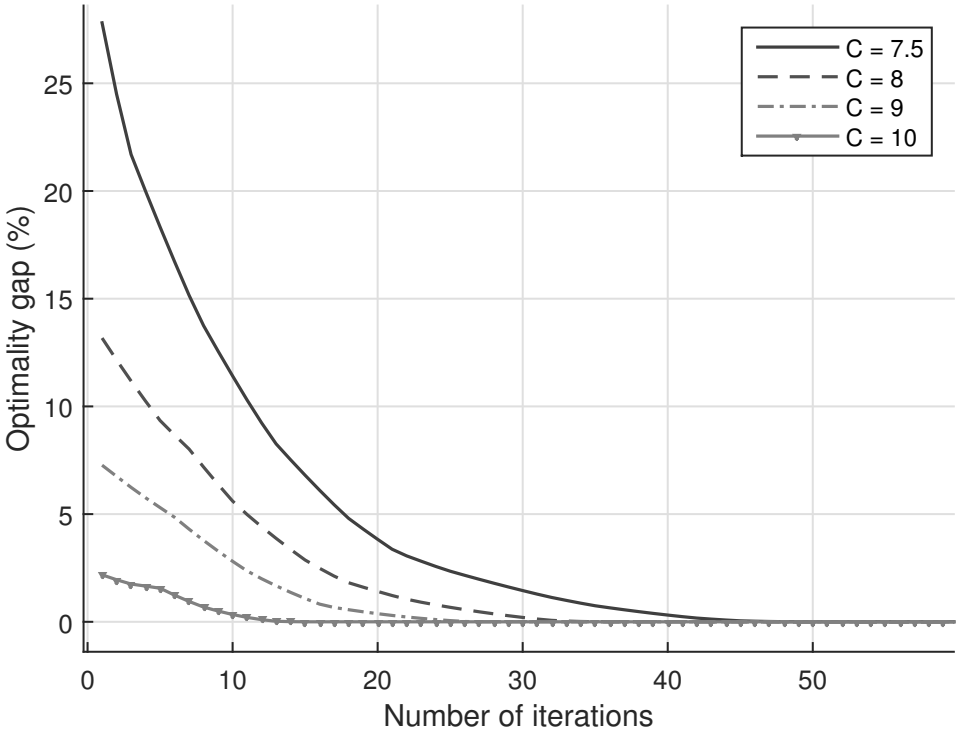

Figure 5.2: Convergence of proposed algorithm for different values of link capacities.

has to be noted that both of the compared algorithms are mobility aware algorithms that take into account the users mobility and handover procedures.

Then, the degree of service differentiation that the proposed algorithm performs is evaluated. For this reason, a scenario with one virtual network with two different services r_1, r_2 of the same demand as before is considered; then the improvement of the delay sensitive service in comparison to the elastic one as the maximum utilisation increases is evaluated. As is presented in Fig. 5.3, as the maximum link utilisation across the network increases, the delay sensitive service achieves to decrease its total end-to-delay 16% less in comparison to the elastic service's one. The degree of the ratio of the two services' delays depends on the topology characteristics and on the potential gain that the alternative paths can offer.

If the ratio a_1/a_2 equals to 1 there is no service differentiation and the algorithm performs only load balancing across the substrate network. The service differenti-

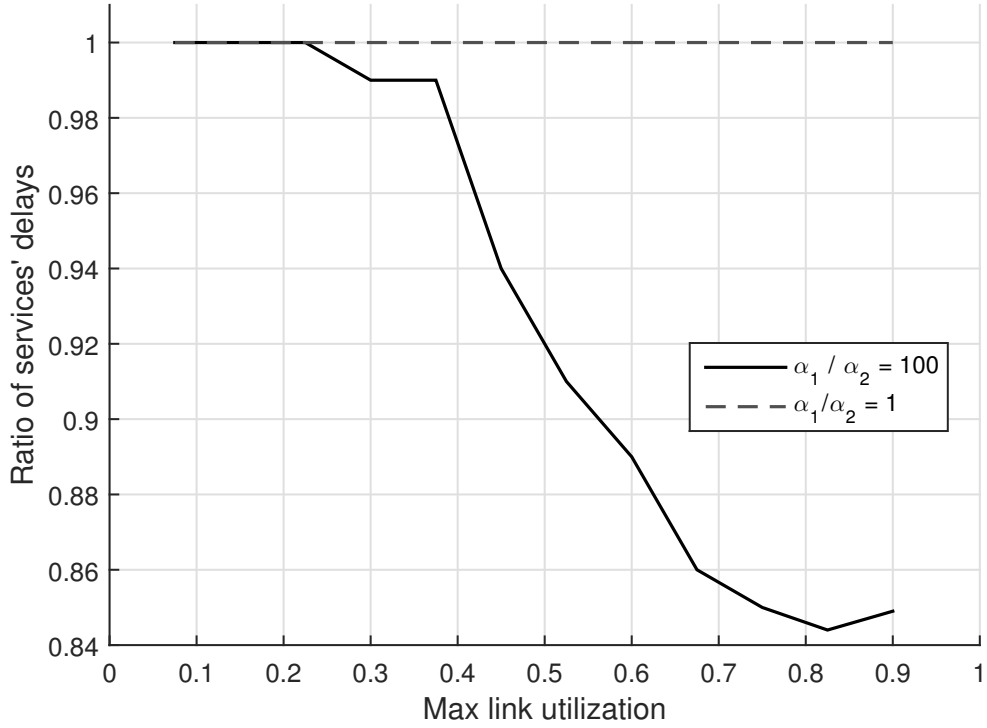

Figure 5.3: Ratio of two services' delays versus maximum link utilisation.

ation is an important feature of this algorithm since it can offer the embedding of virtual networks that can host delay sensitive traffic, exploiting in this way the most out of the available resources.

In conclusion, as it can be seen from the numerical results, the low latency virtual network embedding algorithm converges to an optimal solution and its performance can achieve significant improvement in comparison to a simple shortest path mobility aware virtual network embedding algorithm, as this was firstly presented in [44] and [65] and compared to previous mobility agnostic solutions.

5.5 Publications

1. G. Chochlidakis and V. Friderikos, "Low Latency Virtual Network Embedding for Mobile Networks", in *IEEE International Conference on Communications - Communications QoS, Reliability and Modelling Symposium (ICC16 CQRM), Kuala Lumpur, Malaysia*, May. 2016.

Chapter 6

Robust Virtual Network Embedding

Network virtualisation which will, inter alia, allow for dynamic network sharing, has turned into one of the most prominent solutions that can efficiently deal with the upcoming increase of data demand in mobile networks. In order to allow multiple virtual network operators to share the same substrate infrastructure, the need for the development of efficient virtual network embedding algorithms and techniques is imperative. The main purpose of this chapter is to provide a robust optimization framework for shortest path virtual network embedding, where the traffic demands as well as the user mobility are uncertain (stochastic) rather than deterministic parameters. The proposed algorithm takes into account the effect of the actual users' mobility in order to allow efficient mapping for mobile wireless networks.

The performance of the proposed algorithm is evaluated and compared with existing shortest path based virtual network embedding algorithms that are based on deterministic optimization problems, i.e., using nominal values on the demand. Numerical results reveal that the proposed algorithm can achieve virtual network embedding with adjustable robustness where the trade-off between conservatism and

utilization of resources can be efficiently managed and controlled.

6.1 Introduction

Over the past few years, according to modern network economics, the best-effort and drop-tail based packet processing in IP networks led to the emergency of suitable network over-provisioning as the only way to stay ahead of demand. This orthodox way of network operation is starting to raise real signs of concern for two intertwined reasons.

Firstly, mobile operators in a worldwide scale are facing a steadily, almost exponential traffic increase. According to predictions about the future mobile networks, this booming trend is clearly going to remain the same [67]. This firm increase of data traffic leads to unsustainable network operation, especially due to the well documented fact that revenues from mobile users are inevitably are reaching a plateau [41].

Secondly, emerging types of network traffic related to Machine-to-Machine (M2M) communications and haptic-based applications foreshadow the dawn of a new era for wireless telecommunications where high reliability and robustness will be of great importance. These emerging types of applications will provide a further increase on the aggregate network traffic. Therefore, efficient utilization of installed and emerging substrate network capacity is of paramount importance. To this end network virtualisation is believed to be a key technology to ease the pressure coming from the above two challenges.

As with respect to wireless networks specifically, the virtualisation of the physical available resources (i.e. the so-called substrate network) has been considered as a promising solution for next generation mobile networks, including 5G, in order

to enable efficient and on-demand network sharing, increasing in this way, among other potential advantages, the total utilization of the limited wireless access and core network resources. It is worth mentioning that at the present time, some form of passive (i.e. non-adaptive) physical infrastructure sharing is already used within cellular network operators. Current predictions[1] point out that 90% of mobile operators worldwide will have explored this avenue in some form by the end of 2015. Moreover, network sharing is going to move deeper into the network becoming more dynamic and flexible, able to allow multi-tenancy schemes at different network elements within the core and wireless access network.

It has been well documented that the main advantages of multi-sharing of physical network infrastructures, besides the increase of utilization of the available physical resources, are in summary, the high energy efficiency, the improved flexibility, manageability and scalability that offers and finally the prospect of robustness for mobile operators [43]. Clearly, the potential advantages of network virtualisation are such that this trend is expected to be further adopted in the future.

Network virtualisation will in essence enable dynamic network sharing and hence propel further operational and capital cost reduction by increasing the utilization of the currently deployed network infrastructure. With respect to high reliable communications, construction of virtual networks that explicitly consider the spatial-temporal uncertainty of the real data demand from mobile users will undoubtedly play an important role towards the Tactile Internet vision.

In order to allow multi-sharing, the need for efficient network embedding techniques is imperative. The procedure of mapping virtual networks upon requests on a substrate network is called virtual network embedding, or mapping. In this particular area significant research has been conducted so far. An extended survey of the so far research efforts on virtual network embedding is given in [23] where the authors

[1]Mobile Network Sharing Report Developments, www.reportlinker.com

provide a taxonomy of the algorithms along three distinct dimensions: static versus dynamic, centralized versus distributed, and concise or redundant.

In this chapter, an optimization algorithm for shortest path based virtual network embedding is presented, where the data demands on the virtual networks are subject to uncertainty, i.e., stochastic demands are considered [68]. The main motivation is to provide a scheme that can ensure a robust solution to the embedding problem so that an operator can hedge against the variability of user traffic demand. Then, the proposed scheme makes use of the robust optimization transformation by D. Bertsimas and M. Sim [69] where they approach the data parameter uncertainty by providing a flexible adjustment of the model's conservatism. The final formulated robust mathematical programming model is also a linear optimization problem where the robustness is expressed by probabilistic bounds. An extended survey of the key primary research methods, both theoretical and applied in the area of robust optimization is given in [70].

In addition, the proposed scheme takes into account in the robust optimization framework the user mobility effect in the sense that this solution provisions the traffic caused by data re-direction during the handover process and optimizes its routing accordingly. As already has been extensively detailed, the main logic behind a mobility aware virtual network embedding algorithm [68] is that, in contrast to a mobility unaware solution, it also maps intermediate paths for the data forwarding to the new cell after a handover, provisioning in this way the users' mobility. Hence, the virtual networks become more efficient in terms of handling the handover and forwarding the data to the next edge router (the assumed mobility management is explained in the sequel).

6.2 System Model Description

In the hereafter presented system model, a central controller that is responsible for slicing the physical resources and ensuring isolation among them is considered. The controller is handled by the virtual network embedding algorithm and it serves the requests using a set of pre-defined policies (i.e. minimum delay, minimum routing cost etc.). Hence, the embedding algorithm creates a strategy for resources' virtualisation and for their allocation to each one of the virtual networks according to the incoming requests.

Then it communicates with the network elements in order to form the virtual networks. In this way, the virtual networks are formed and become operational for the virtual network mobile operators. Each virtual network is fully independent and isolated from the others and has no awareness of their co-existence in the same substrate network.

Regarding the architecture of this model, and without loss of generality, as with the previous solutions, a single gateway and several destination edge routers are considered. Each virtual network request is defined by demands from the source gateway to the edge routers. Hence, the virtual networks have a substrate topology dependence and should include the same source and destination nodes.

As for the mobility management function, the emerging distributed mobility management (DMM) [9] is assumed as the deployed solution, in which every single destination edge router acts as a mobility management anchor point. When a handover occurs, the flow anchors to the previous edge router are forwarded to the next one. Since the presented algorithms are mobility aware, they will also embed the best paths that connect those edge routers according to a deterministic handover matrix. Therefore, it is considered that by applying SDN and NFV deployment, the mobility function is virtualised by co-locating it at the edge of the core network (i.e.

the edge routers).

6.2.1 Shortest path virtual network embedding algorithm

First the optimization algorithm for shortest path virtual network embedding when no uncertainties exist is presented.

Substrate network

The substrate mobile core network can be modelled as an undirected planar tree-like graph $G = (Q, L)$. Q represents the set of nodes and L depicts the set of links. Let $C : L \rightarrow \mathbb{R}^+$ be the capacity of each link. The network gateway is located on the source node of the graph, while the leaves of the graph represent the destination edge routers. The rest of the nodes are the routers of the core network. The algorithm that is presented hereafter assigns nodes and links in order to form the virtual networks upon the virtual network requests and with respect to capacity constraints.

Virtual network requests

In order to define the set of virtual network requests V it is considered that each request $v \in V$ can be represented as a graph $G' = (Q', L')$ that has to be mapped on the graph $G = (Q, L)$. Each graph $G' = (Q', L')$ is consists of a set of virtual paths $s \rightarrow k$ that connect the virtual gateway s with the virtual edge routers $k \in K$, where $K \subset Q$.

Mobility effect

As with the previous proposals, in order to capture the actual users' mobility, a handover matrix is considered $\boldsymbol{H}_{V\times K\times K}$, the elements $h_{vkj} \in (0,1)$ of which represent the probability of the flows destined to edge router $k \in K$ to migrate to another edge router $j \in J$ (where $J \equiv K$) for the virtual network request $v \in V$. It is considered that $h_{rkj} = 1$ for $k = j$ as the pairs $\{k-k\}$ represent the paths that connect the source gateway with the edge router k. For the following simulations, handover up to two consecutive hops is considered and for those pairs $\{k-j\}$ let $h_{vkj} \in \left[0, \xi\right]$ uniformly.

Demand for resources

In the nominal virtual network embedding algorithm there are deterministic demands for assigned resources; these can be for example average values over a specific time horizon. Let $d_{vkj} \in D$ be the demand for virtual resources for the source-destination pair $\{k-j\}$. It is also considered a minimum threshold for mapping of resources B_{vkj} that is set by the virtual network operator and has to be assigned on the corresponding path.

Mobility management scheme

In this work and without loss of generality, again, it is assumed that DMM anchoring takes place at the edge routers. While DMM scheme is in use, there is an additional need to assign paths that connect edge routers or network elements where DMM anchoring is taking place. This means that the graph $G' = (Q', L')$ also includes paths $\{k-j\}$ that connect directly the edge routers. It is needed to be noted that there is a topology dependence of the substrate and the virtual networks in a way that the virtual gateway has to be mapped on the physical one and the virtual edge

routers have to be also mapped on the corresponding physical edge routers (e.q. Fig. 5.1).

Problem parameters and variables

Based on the above setting, the goal is to find the optimal selection of paths in order to minimize the total routing cost. Below, it is presented a summary of the parameters used for the formulation of the linear mathematical optimization and the introduction of the linear variables:

- $G = (Q, L)$: undirected planar tree-like graph,
- π_{kjp}: p^{th} alternative substrate path that connects: i) the source node with the edge router k if $k = j$ and, ii) the edge router k with the edge router j. For each pair $\{k - j\}$, there are $P \in \mathbb{N}^+$ alternative paths,
- $v \in V \subset \mathbb{N}^+$: set of virtual network requests,
- $K, J \subset Q$: set of edge routers (destination nodes),
- $d_{vkj} \in D$: demand for mapping resources for the the virtual network request $v \in V$ on π_{kjp},
- $B_{vkj} \in B$: set of minimum assigned resources,
- $h_{vkj} \in \boldsymbol{H}_{V \times K \times K}$: mobility matrix, and
- $\beta_{kjpl} = \begin{cases} 1, \text{ if link } l \in \pi_{kjp} \\ 0, \text{ otherwise} \end{cases}$.

Then, the linear variable $x_{vkjp} \in \boldsymbol{x}$ is defined, which is the data rate of the resources that are mapped (valued for zero for no mapping case) on the substrate path π_{kjp} to satisfy the virtual network request $v \in V$.

Objective function

The optimization problem algorithm minimizes the summation of every assigned data rate of resources multiplied by the minimum required demand for resources. Hence, the objective function can be written as follows:

$$\Omega(\boldsymbol{x}) = \sum_{vkjp} x_{vkjp}\mu_{kjp}B_{vkj} \tag{6.1}$$

where μ_{kjp} is the routing cost for each path π_{kjp}. Based on the above definition, the linear programming optimization problem can be formulated as follows:

$$\min \boldsymbol{\Omega}(\boldsymbol{x}) \quad \text{s.t.} \tag{6.2}$$

$$\sum_{vkjp} h_{vkj}d_{vkj}\beta_{kjpl}x_{vkjp} \leq C_l \ , \ \ \forall l \in L, \tag{6.3}$$

$$\sum_{p} h_{vkj}d_{vkj}x_{vkjp} \geq B_{vkj} \ \ , \forall v \in V, \ k \in K, \ j \in J, \tag{6.4}$$

$$x_{vkjp} \geq 0 \ , \ \ \forall v \in V, \ k \in K, \ j \in J, \ p \in P. \tag{6.5}$$

where constraint (6.3) ensures that the capacity of every substrate link is not violated after the set up of the virtual networks, constraint (6.4) defines the minimum required resource mapping for each virtual network request and constraint (6.5) makes sure that the variable $\boldsymbol{x}$ is always non negative.

6.2.2 A Robust Optimization Approach

The previously presented algorithm considers explicitly deterministic variables, which in essence express expected values of user demand. However, in reality traffic demand varies. In this proposed scheme the following problem is assumed: having a set of virtual network requests and a substrate network with capacity constraints

and threshold demands for assigning resources, how can the optimal (in terms of shortest path) way to embed those virtual networks be calculated, assuming that the demands and the users' mobility are subject to uncertainty.

In particular, it is assumed that after setting the minimum requested amount of resources to be mapped, the demands of the traffic and the mobility of the users is not deterministic but uncertain and thus, in the above formulation, each coefficient $\tilde{r}_{vkj} = h_{vkj}d_{vkj}$ is now a random variable. Firstly, the optimization problem (6.2) can be rewritten in the following form:

$$\min \boldsymbol{\Omega}(\boldsymbol{x}) \quad \text{s.t.} \tag{6.6}$$

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \tag{6.7}$$

$$\boldsymbol{x} \geq 0 \tag{6.8}$$

Let I represent the number of constraints (rows of inequalities matrix $\boldsymbol{A}$) of the minimization problem (6.2) and N the length of $\boldsymbol{x}$ (columns of matrix $\boldsymbol{A}$). Let $\alpha_{ij} \in A$ be the elements of $\boldsymbol{A}$. For each constraint $i \in I$ $N_i \subset N$ is the set of coefficients $\tilde{\alpha}_{in}$ that are random variables, following an unknown symmetric distribution in $\left[\alpha_{ij} - \hat{\alpha}_{in}, \alpha_{in} + \hat{\alpha}_{in}\right]$ with a mean value equal to the nominal value α_{in}.

Following [69], it is considered that for every constraint $i \in I$, only $\Gamma_i \in \left[0, |N_i|\right]$ coefficients will be subject to parameter uncertainty. The main role of parameter Γ_i is to adjust the robustness of the proposed virtual network embedding scheme. This is because of the fact that eventually not all of the $\alpha_{in} \in N_i$ will change leading to a violation of the constraints.

In particular, the motivation behind the implementation is to hedge against traffic increases that can can lead to a violation of the capacity constraints, harming the

performance of the particular virtual network. A rather pessimistic approach to this problem is to consider the worst case scenario, assuming that the all of the demands will get their maximum values. This can be achieved by setting the parameter $\Gamma_i = 0, \forall i \in I$. In this way the scheme embeds virtual networks that can serve the maximum possible demand but it wastes the substrate's network resources, causing as a result a low resource utilization.

On the other hand, if it is set $\Gamma_i = |N_i|, \forall i \in I$, considering that none of the random parameters $\alpha_{in} \in N_i$ will change, this is similar to the case of the nominal problem (6.2), which can be considered when using average values.

The great advantage that parameter Γ_i offers is that it makes it possible to adjust the conservatism and thus the robustness of the virtual network embedding algorithm. Moreover, the robustness becomes deterministic: if less than $\lfloor \Gamma_i \rfloor$ coefficients $\alpha_{in} \in N_i$ change indeed then the algorithm is robust but even if more than $\lfloor \Gamma_i \rfloor$ are increased then the solution will be feasible with very high probability.

It needs to be taken into account that for this case, only the increase of the traffic demand that will be served by a virtual network can lead to infeasibility because it may cause a capacity violation. If the actual demand is less than the nominal, although leading to lower utilization, the robustness remains intact.

Based on the above definitions, the robust virtual network embedding algorithm is provided, formulated as a linear programming optimization problem by introducing three new variables $\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{z}$ that will logically bind the Γ_i parameter with the problem's constraints:

$$\min \boldsymbol{\Omega}(\boldsymbol{x}) \quad \text{s.t.} \tag{6.9}$$

$$\sum_{n} \alpha_{in} x_n + z_i \Gamma_i + \sum_{n \in N_i} u_{in} \leq b_i \ , \ \forall i \in I, \tag{6.10}$$

$$z_i + u_{in} \geq \hat{\alpha}_{in} y_n \ , \ \forall i \in I, \ \forall n \in N_i, \tag{6.11}$$

$$-y_n \leq x_n \leq y_n \ , \ \forall n \in N, \tag{6.12}$$

$$x_n \geq 0 \ , \ \forall n \in N, \tag{6.13}$$

$$u_{in} \geq 0 \ , \ \forall i \in I, \ \forall n \in N_i, \tag{6.14}$$

$$y_n \geq 0 \ , \ \forall n \in N, \tag{6.15}$$

$$z_i \geq 0 \ , \ \forall i \in I. \tag{6.16}$$

6.3 Numerical Evaluation

After having provided a mathematical framework for virtual network embedding with adjustable robustness, the proposed scheme is implemented using MATLAB's optimization toolbox. For these simulations it is assumed a dense binary tree with 31 nodes in total as the scenario's substrate network, with the following parameters: $C_l = 100$, $h(k, j) = \big[0, 0.2\big]$, $K = 16$, $B = h \cdot d = \alpha$ (nominal value).

In Fig. 6.1 the values of the objective function Ω as the value of Γ increases is presented. The simulations are performed for $\alpha_{ij} = 1$, $\hat{\alpha}_{ij} = 1$ and for $\alpha_{ij} = 4$, $\hat{\alpha}_{ij} = 4$. As already stated, when $\Gamma = 0$ this becomes the nominal problem case and when $\Gamma = |J_i|$ then this becomes the most conservative approach by solving the problem using the worst case scenario in terms of the realization of the stochastic demands (also known as Soyster's formulation [71]). As the value of the parameter Γ increases, inevitably more resources of the substrate network are mapped and the value of the objective function Ω increases.

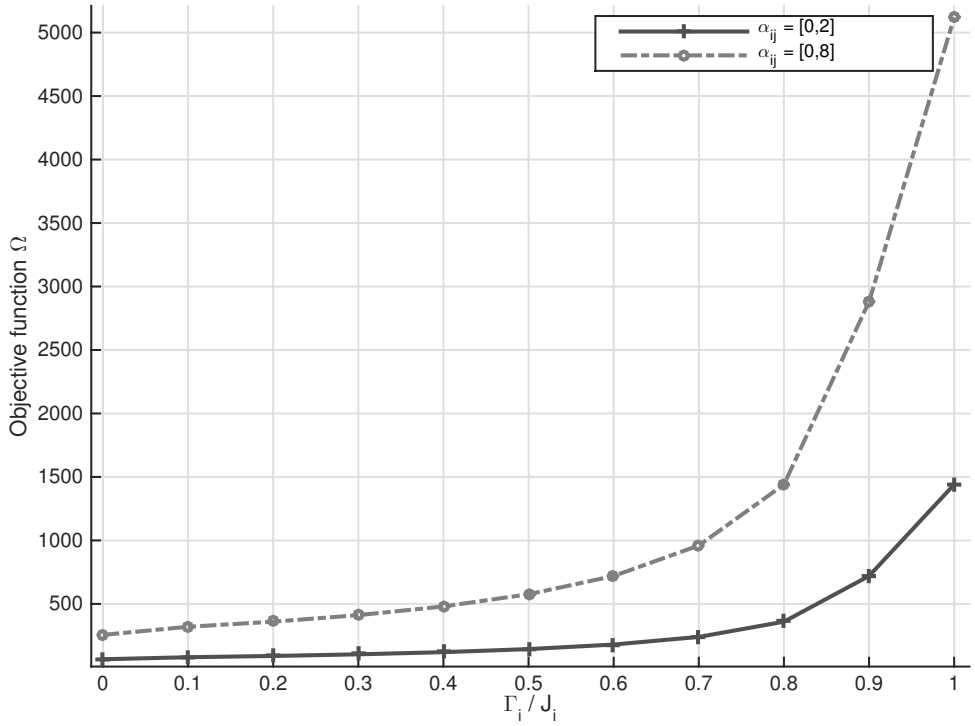

Figure 6.1: The objective value against different values of $\Gamma_i/|J_i|$

The main advantage of the implementation of the parameter Γ is that it offers the tuning flexibility of the scheme's robustness. In order to evaluate the relationship between robustness and conservatism, the probability of violation is captured after the calculation of the optimal vector $\boldsymbol{x}$ while the value of the parameter Γ increases, via Monte Carlo simulations.

In Fig. 6.2 the violation probability is presented for $V = 1$ and $V = 4$ virtual network requests. As can be observed, while the number of virtual network requests increases, the length of the vector $\boldsymbol{x}$ and of the uncertain coefficients increases as well.

Interestingly, taking into account only 40% of the uncertain coefficients, almost up to 10% violation probability can be achieved, while, when approaching the worst-case conservative approach, there is no clear benefit in terms of probability violation, but the value of the objective function increases (Fig. 6.1).

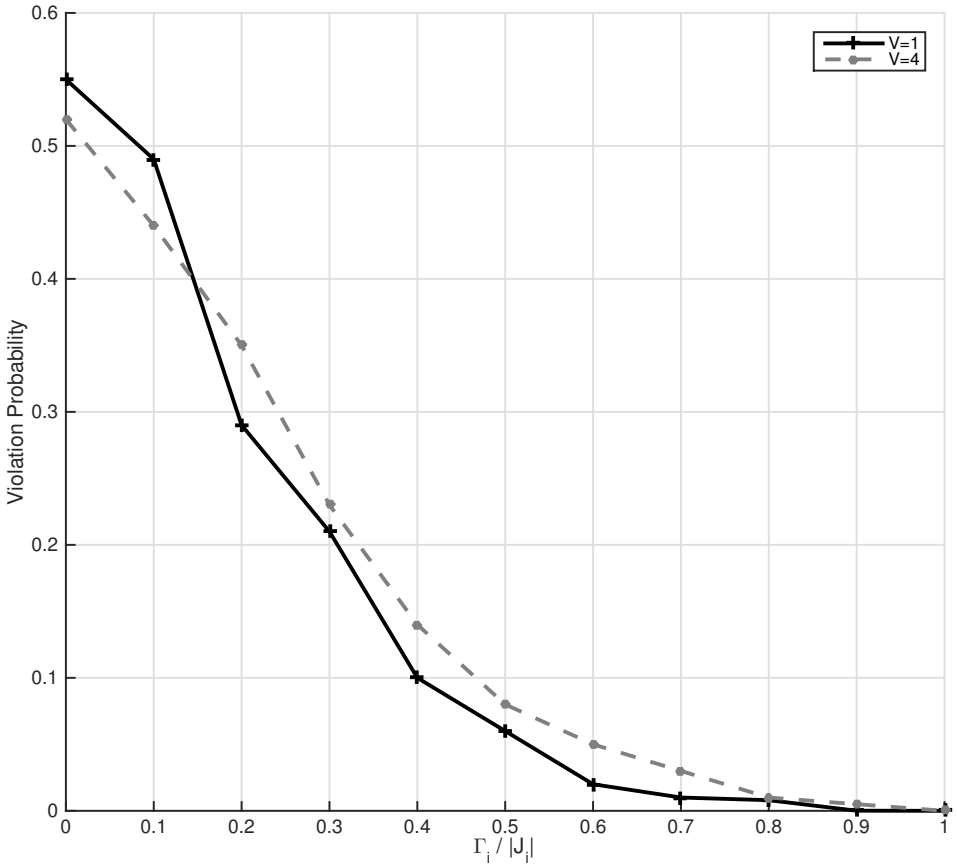

Figure 6.2: Violation probability against different values of Γ_i $|J_i|$

The key beneficial feature of the proposed approach is parsimony in essence that only one parameter is required to adjust the level of robustness. The required level of robustness can be estimated from traffic measurements and clearly expected to change according to various exogenous parameters such as time of the day, hot-spot creation etc.

After the decision of the desired level of robustness, by selecting the appropriate values of Γ coefficient, the problem can be solved against this level of robustness, with the trade-off of the correspondent used resources. This particular proposal does not provide a stochastic optimisation solution but a adjustable-robustness optimisation framework for virtual network embedding with a flexible level of robustness against a probabilistic nature of parameters and constraints.

6.4 Publications

1. G. Chochlidakis and V. Friderikos, "Robust Virtual Network Embedding for Mobile Networks", in *IEEE 26th International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): Mobile and Wireless Networks (IEEE PIMRC2015 Mobile & Wireless), Hong Kong, P.R. China*, Aug. 2015, pp. 18671871.

Chapter 7

Conclusion

7.1 Concluding remarks

To resolve the challenges stemming out of the increased traffic demand and heterogeneity of that traffic, research towards 5G will inevitably have to put in the epicentre some big questions surrounding the future of mobile communications. The proliferation of small base stations towards the vision of HetNets and network densification creates by itself a set of complex management issues ranging from uses such as spectrum re-use to coverage, consistency on the service delivery and QoE support. In this significant more complex environment the notion of hexagonal cells will be decreasing and mobile nodes will participate more actively on overall network management and control.

Bearing that in mind, it becomes clearer that emerging architectural paradigms such as the cell on demand and the phantom cell concept that aim to split control and data planes physically in order to ease network management will become more and more prominent at next-generation mobile networks. In separate efforts, SDN/NVF in essence will allow for logical separation of control and data forwarding planes as

well as decoupling between hardware and software of various network functionalities to allow for further innovation in networks. Hence combining these two emerging paradigms is an area of significant interest for future research.

Next-generation wireless networks will also inevitably tend towards a decentralization of the mobility management, in order to achieve better handover performance, more efficient routing and generally improved network performance. For this reason, in this work the the main mobility management technique that was considered was the Distributed Mobility Management (DMM) solution. The simulation results that were conducted showed that Distributed Mobility Management achieves better overall performance than the centralized scheme in terms of total routing cost. In addition, both of these solutions present a notable topology dependence, with the DMM scheme being more affected by the network topology in some cases.

Based on these findings, a Hybrid DMM scheme was proposed, where the mobility function is also distributed at the Access Routers (ARs) except specific areas, where data forwarding can lead to very high latency and routing cost; for this reason, in DMM, these areas are supported by Mobility Anchors (MAs) that are located hierarchically higher across the network. Simulations results showed that the proposed scheme can achieve a significant performance improvement especially for supporting time-critical over-the-top services compared to traditional distributed and centralised approaches.

Network virtualisation has been considered as an enabler to provide quick roll outs of new techniques, reduce cost via infrastructure sharing while at the same time some key challenges with respect to network virtualisation have been identified as pertain to wireless networks. In the path towards 5G, there is an urgent need to explore revolutionary approaches in terms of network orchestration and management compared to the current paradigms in order to allow for service consistency as well as efficient utilization of the available resources.

On the other hand, an evolution is envisioned on lower layer techniques that will aim to further increase capacity by mainly exploring higher radio frequencies. The expectation is that network sharing via virtualization will move deeper into the core network of operators and will allow for significant lower capital and operational expenditure. A key technical aspect of network virtualization is virtual network embedding, which relates to efficient mapping between physical and virtual resources. The problem of virtual network embedding has been previously studied mainly for fixed networks without taking into account the effect of mobility management solution, which affect routing decisions in case of user mobility.

7.1.1 Mobility and Virtual Network Embedding

In this work, the impact of adopting DMM on virtual network embedding was revealed and it was showed how forming virtual network without considering the mobility can potentially entail in suboptimal virtual networks. To this end, the problem of virtual network embedding was formulated as an integer mathematical program in a way that takes explicitly into account the users mobility effect. The simulation results showed that the users mobility effect has a significant impact on the virtual network embedding procedure. Numerical investigations revealed that the performance and efficiency of the networks can be significantly increased when the proposed mobility aware algorithm is compared to embedding algorithms that are mobility agnostic.

7.1.2 Delay Sensitive Virtual Network Embedding

Furthermore, a virtual network embedding algorithm that minimizes the total end-to-end delay under service differentiation was proposed. In addition, the user mobility effect was again taken into account, making the algorithm suitable for mobile

network sharing scenarios, especially when emerging low latency applications with tactile required latency will need to be supported. The proposed algorithm was compared with nominal previous solutions based on a shortest path optimization embedding algorithm and the simulation results revealed a potentially significant improvement in terms of total delay and service differentiation.

7.1.3 Robust Virtual Network Embedding

Lastly, a virtual network embedding solution for non-deterministic problem parameters was proposed. The proposal was modelled as a virtual network embedding algorithm formulated as a linear programming problem with adjustable robustness in order to hedge against traffic variability. The proposed algorithm was compared with previous solutions that are based on a shortest path embedding algorithm and use nominal values for the requested demand. The simulation results revealed that the proposed approach can provide significant benefits, since it can effectively deal with traffic demand uncertainty.

7.2 Future work

As the softwarisation of the network becomes more and more prominent at the era of next-generation mobile networks, the continuation of this work will be focused on the migration of the main and secondary network functions at the cloud, a technique known as Network Function Virtualisation (NFV), and the problem of the optimal location of those logical function. Scenarios where the problem of caching servers' location for content delivery is still challenging, due to the scale and the complexity of the infrastructure, could make use of similar solutions to the mobility related ones, proposed in this work, in order to maximise the efficient use of the available resources. In combination to the mobility effect, forming virtual networks along with

efficiently locating the virtual network functions on them can pose future challenges that can be considered as a future work and continuation of this proposal.

In particular, content caching at the network level is attracting increasing interest as a technique that not only eases the data traffic generated to the base station, but also reduces the access latency and the overall power consumption. By caching popular content along the network, user requests can be then satisfied by closely located mobile clients without the need to retrieve them from the source, which is assumed to be, in the best case scenario, the edge of the network. However, due to the devices capacity restrictions, the involved users cannot cache a large number of files but only a subset of these. To this end, various cache management policies are applied in order to achieve effective content delivery among the network terminals.

A very prominent area for future research, also seen as extension of this work, is the field of Network Coding. Although not included in this work, there has already been a collaborative work that provided a network coded, cooperative cache management method that considered the compression of the existing files in a user's cache to store more contents of interest [72]. In combination with the flexibility that SDN offers, there is a very good potential for extending the mobility aware virtual network embedding techniques that were presented in this work. Lastly, the NFVs included in the process of VNE can be part of huge set of network capabilities and essential functions.

Appendix A

Acronyms

Access Router	AR
Application Programming Interface	API
Average Revenue Per User	ARPU
Binding Update	BU
Carrier Aggregation	CA
Centralized Mobility Management	CMM
Coordinated Multipoint	CoMP
Correspondent Node	CN
Distributed Mobility Management	DMM
Evolved Packet Core	EPC
Hierarchical Mobile IPv6	HMIPv6
Home Address	HoA
Home Agent	HA
Hybrid Distributed Mo- bility Management	HDMM
Internet Engineering Task Force	IETF
Internet Multimedia Subsystem	IMS
Local IP Address	LIPA

Local Mobility Anchor	LMA
Long Term Evolution - Advanced	LTE-A
Machine-to-Machine	M2M
Mobile IPv4	MIPv4
Mobile IPv6	MIPv6
Mobile Node	MN
Mobility Access Gateway	MAG
Mobility Agent	MA
Mobility Anchor Point	MAP
Network Function Virtualisation	NFV
Network Virtualisation	NV
Network Virtualisation Substrate	NVS
On-Link Care of Address	LCoA
Open Networking Foundation	ONF
Orthogonal Frequency-Division Multiple Access	OFDMA
Over-the-Top	OTT
Radio Access Network	RAN
Radio Access Technology	RAT
Regional Care of Address	RCoA
Software-defined Networking	SDN
Router Advertisement	RA
Proxy Binding Acknowledgment	PBA
Proxy Binding Update	PBU
Proxy Mobile IPv6	PMIPv6
Quality of Experience	QoE
Quality of Service	QoS
Selected IP Traffic Offloading	SIPTO
Self-Organising Networking	SON

Virtual Network	VN
Virtual Network Embedding	VNE

Table A.1: List of acronyms

List of Algorithms

Bibliography

[1] R. Tamijetchelvy and G. Sivaradje, "An optimized fast vertical handover strategy for heterogeneous wireless access networks based on ieee 802.21 media independent handover standard," in *2012 Fourth International Conference on Advanced Computing (ICoAC)*, Dec 2012, pp. 1–7.

[2] H. Bing, C. He, and L. Jiang, "Performance analysis of vertical handover in a umts-wlan integrated network," in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.*, vol. 1, Sept 2003, pp. 187–191 Vol.1.

[3] C. Perkins, "IP Mobility Support for IPv4, Revised," RFC 5944 (Proposed Standard), IETF, Nov. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5944.txt

[4] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275 (Proposed Standard), IETF, Jul. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6275.txt

[5] R. Koodli, "Fast Handovers for Mobile IPv6," RFC 4068 (Experimental), Internet Engineering Task Force, July 2005, obsoleted by RFC 5268. [Online]. Available: http://www.ietf.org/rfc/rfc4068.txt

[6] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380 (Proposed Standard), IETF, Oct. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5380.txt

[7] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213 (Proposed Standard), IETF, Aug. 2008, updated by RFC 6543. [Online]. Available: http://www.ietf.org/rfc/rfc5213.txt

[8] J. h. Lee, J. m. Bonnin, P. Seite, and H. A. Chan, "Distributed ip mobility management from the perspective of the ietf: motivations, requirements, approaches, comparison, and challenges," *IEEE Wireless Communications*, vol. 20, no. 5, pp. 159–168, October 2013.

[9] D. Liu, J. Zuniga, P. Seite, H. Chan, and C. Bernandos, "Distributed Mobility Management: Current practices and gap analysis ," Internet Draft: draft-ietf-dmm-best-practices-gap-analysis-03, IETF, 2014. [Online]. Available: http://datatracker.ietf.org/doc/draft-ietf-dmm-best-practices-gap-analysis//

[10] H. Chan, D. Liu, P. Seite, H. Yokota, and J. Korhonen, "Requirements for Distributed Mobility Management," Internet Draft: draft-ietf-dmm-requirements-15 (work-in-progress), IETF, 2014. [Online]. Available: http://datatracker.ietf.org/doc/draft-ietf-dmm-requirements/

[11] J. Zuniga, C. Bernardos, A. de la Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed mobility management: A standards landscape," *Communications Magazine, IEEE*, vol. 51, no. 3, pp. 80–87, March 2013.

[12] P. McCann, "Authentication and Mobility Management in a Flat Architecture," IETF Internet Draft, IETF, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/draft-mccann-dmm-flatarch-00

[13] W. Hahn, "3gpp evolved packet core support for distributed mobility anchors: Control enhancements for gw relocation," in *2011 11th International Conference on ITS Telecommunications*, Aug 2011, pp. 264–267.

[14] ——, "Flat 3gpp evolved packet core," in *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Oct 2011, pp. 1–5.

[15] C. J. Bernardos, J. C. Zniga, and A. Reznik, "Towards flat and distributed mobility management: A 3gpp evolved network design," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 6855–6861.

[16] T. Guo, A. ul Quddus, N. Wang, and R. Tafazolli, "Local mobility management for networked femtocells based on x2 traffic forwarding," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 326–340, Jan 2013.

[17] D. Xenakis, N. Passas, L. Merakos, and C. Verikoukis, "Mobility management for femtocells in lte-advanced: Key aspects and survey of handover decision algorithms," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 64–91, First 2014.

[18] A. Khan, W. Kellerer, K. Kozu, and M. Yabusaki, "Network sharing in the next mobile network: Tco reduction, management flexibility, and operational independence," *IEEE Communications Magazine*, vol. 49, no. 10, pp. 134–142, Oct 2011.

[19] X. Costa-Perez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 27–35, July 2013.

[20] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "Cellslice: Cellular wireless resource slicing for active ran sharing," in *2013 Fifth International Confer-*

ence on Communication Systems and Networks (COMSNETS), Jan 2013, pp. 1–10.

[21] J. S. Panchal, R. D. Yates, and M. M. Buddhikot, "Mobile network resource sharing options: Performance comparisons," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4470–4482, September 2013.

[22] T. Guo and R. Arnott, "Active lte ran sharing with partial resource reservation," in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sept 2013, pp. 1–5.

[23] A. Fischer, J. Botero, M. Till Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 4, pp. 1888–1906, Apr. 2013.

[24] "Open Networking Foundation, White paper: Software Defined Networking: The New Norm for Networks," Apr. 2012.

[25] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[26] R. Enns and M. Bjorklund and J. Schonwalder and A. Bierman, "NETCONF Configuration Protocol," RFC 6241 (Proposed Standard), IETF, Jun. 2011.

[27] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Technical Report: FlowVisor: A Network Virtualization Layer," no. OPENFLOW-TR-2009-2, Oct. 2009.

[28] ——, "Can the Production Network Be the Testbed?" in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–6.

[29] S. Gutz, A. Story, and N. Foster, "Splendid Isolation: A Slice Abstraction for Software-Defined networks," in *in ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, 2012.

[30] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in sdn-openflow networks," *Computer Networks*, vol. 71, no. 0, pp. 1 – 30, 2014.

[31] N. Operators, "Network Functions Virtualization, An Introduction, Benefits, Enablers, Challenges and Call for Action," in *SDN and OpenFlow SDN and OpenFlow World Congress*, 2012.

[32] R. Bolla, C. Lombardo, R. Bruschi, and S. Mangialardi, "DROPv2: Energy Efficiency Through Network Function Virtualization," *Network, IEEE*, vol. 28, no. 2, pp. 26–32, March 2014.

[33] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE Mobile Core Gateways, the Functions Placement Problem," in *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*, ser. AllThingsCellular '14. New York, NY, USA: ACM, 2014, pp. 33–38.

[34] V. Yazici, U. Kozat, and M. Sunay, "A New Control Plane for 5G Network Architecture with a Case Study on Unified Handoff, Mobility, and Routing Management," *Communications Magazine, IEEE*, vol. 52, no. 11, pp. 76–85, Nov 2014.

[35] Cisco. (2014) Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20132018. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html

[36] A. Pragad, V. Friderikos, and A. Aghvami, "Optimal configuration of mobility agents in broadband wireless access networks," in *GLOBECOM Workshops, 2008 IEEE*, Nov 2008, pp. 1–6.

[37] P. Ernest, O. Falowo, and H. Chan, "Network-based distributed mobility management: Design and analysis," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference*, Oct 2013, pp. 499–506.

[38] T.-X. Do and Y. Kim, "Distributed network mobility management," in *Advanced Technologies for Communications (ATC), 2012 International Conference*, Oct 2012, pp. 319–322.

[39] L. Yi, H. Zhou, D. Huang, and H. Zhang, "Performance analysis for distributed mobility management schemes based on flow duration," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, Dec 2012, pp. 1068–1072.

[40] G. Chochlidakis and V. Friderikos, "Hybrid distributed mobility management for next-generation wireless networks," in *2014 International Conference and Workshop on the Network of the Future (NOF)*, Dec 2014, pp. 1–6.

[41] M. Nicosia, R. Klemann, K. Griffin, S. Taylor, B. Demuth, J. Defour, R. Medcalf, T. Renger, and P. Datta, "Rethinking Flat Rate Pricing for Broadband Services," *CISCO Internet Business Solutions Group (IBSG)*, Jul. 2012, White Paper.

[42] Chowdhury, N.M.M.K. and Boutaba, R., "Network Virtualization: State of the Art and Research Challenges," *Communications Magazine, IEEE*, vol. 47, no. 7, pp. 20–26, July 2009.

[43] N. M. K. Chowdhury and R. Boutaba, "A Survey of Network Virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862 – 876, 2010.

[44] G. Chochlidakis and V. Friderikos, "Mobility Aware Virtual Network Embedding," in *IEEE International Conference on Communications (ICC)*, London, United Kingdom, Jun. 2015.

[45] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual Network Embedding with Coordinated Node and Link Mapping," in *INFOCOM 2009, IEEE*, Apr. 2009, pp. 783–791.

[46] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, "Optimal Virtual Network Embedding: Node-Link Formulation," *Network and Service Management, IEEE Transactions on*, vol. 10, no. 4, pp. 356–368, Dec. 2013.

[47] G. Hernando, S. Perez, and J. Cabero, "Mobility-Aware Distributed Embedding (MADE) of Virtual Networks," in *Future Network and Mobile Summit, 2010*, Jun. 2010, pp. 1–8.

[48] I. Houidi, W. Louati, and D. Zeghlache, "A Distributed Virtual Network Mapping Algorithm," in *Communications, 2008. ICC '08. IEEE International Conference on*, May 2008, pp. 5634–5640.

[49] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNE-AC: Virtual Network Embedding Algorithm Based on Ant Colony Metaheuristic," in *Communications (ICC), 2011 IEEE International Conference on*, Jun. 2011, pp. 1–6.

[50] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.

[51] Z. Cai, F. Liu, N. Xiao, Q. Liu, and Z. Wang, "Virtual Network Embedding for Evolving Networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec. 2010, pp. 1–5.

[52] R. Gomes, L. Bittencourt, and E. Madeira, "A Bandwidth-Feasibility Algorithm for Reliable Virtual Network Allocation," in *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, May 2014, pp. 504–511.

[53] Y. Zhu and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–12.

[54] C. H. Papadimitriou, "On the Complexity of Integer Programming," *J. ACM*, vol. 28, no. 4, pp. 765–768, Oct. 1981.

[55] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.

[56] G. Tseliou, F. Adelantado, and C. Verikoukis, "Scalable RAN Virtualization in Multi-Tenant LTE-A Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[57] ——, "Resources negotiation for network virtualization in LTE-A networks," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 3142–3147.

[58] G. Fettweis, "The Tactile Internet: Applications and Challenges," *Vehicular Technology Magazine, IEEE*, vol. 9, no. 1, pp. 64–70, March 2014.

[59] G. Chochlidakis and V. Friderikos, "Low latency virtual network embedding for mobile networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[60] L. Shengquan, W. Chunming, Z. Min, and J. Ming, "An Efficient Virtual Network Embedding Algorithm with Delay Constraints," in *16th International*

Symposium on Wireless Personal Multimedia Communications (WPMC), June 2013, pp. 1–6.

[61] J. Infuhr and G. Raidl, "Introducing the Virtual Network Mapping Problem with Delay, Routing and Location Constraints," in *Network Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6701, pp. 105–117.

[62] S. Ayoubi, C. Assi, K. Shaban, and L. Narayanan, "Multicast Virtual Network Embedding in Cloud Data Centers with Delay Constraints," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–1, 2015.

[63] U. Javed, M. Suchara, J. He, and J. Rexford, "Multipath Protocol for Delay-sensitive Traffic," in *Proceedings of the First International Conference on COMmunication Systems And NETworks*, ser. COMSNETS'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 438–445.

[64] J. He, R. Zhang-shen, Y. Li, C. yen Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet," in *Proc. CoNEXT*, 2008.

[65] G. Chochlidakis and V. Friderikos, "Mobility aware virtual network embedding," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1343–1356, May 2017.

[66] D. Bertsekas and R. Gallager, *Data Networks (2Nd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[67] CISCO, "Global Mobile Data Traffic Forecast Update, 2014-2019," in *Cisco White Paper*, February 2015.

[68] G. Chochlidakis and V. Friderikos, "Robust virtual network embedding for mobile networks," in *2015 IEEE 26th Annual International Symposium on Per-*

sonal, Indoor, and Mobile Radio Communications (PIMRC), Aug 2015, pp. 1867–1871.

[69] D. Bertsimas and M. Sim, "The Price of Robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004.

[70] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and Applications of Robust Optimization," *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, Aug. 2011.

[71] A. L. Soyster, "Technical NoteConvex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming," *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973.

[72] C. Vlachos, G. Chochlidakis, J. Heide, and V. Friderikos, "Network Coded Compression-based Caching for Device-to-Device Communications," in *European Conference on Networks and Communications (EuCNC)*, Athens, Greece, Jun. 2016.