Telecom Networks Virtualization

Overcoming the Latency Challenge

Dejene Boru Oljira
Telecom Networks Virtualization

Overcoming the Latency Challenge

Dejene Boru Oljira
Telecom Networks Virtualization: Overcoming the Latency Challenge

Dejene Boru Oljira

LICENTIATE THESIS

Karlstad University Studies | 2018:24

urn:nbn:se:kau:diva-67243

ISSN 1403-8099


© The author

Distribution:
Karlstad University
Faculty of Health, Science and Technology
Department of Mathematics and Computer Science
SE-651 88 Karlstad, Sweden
+46 54 700 10 00

Print: Universitetstryckeriet, Karlstad 2018
Telecom Networks Virtualization: Overcoming the Latency Challenge

Dejene Boru Oljira
Department of Mathematics and Computer Science

Abstract

Telecom service providers are adopting a Network Functions Virtualization (NFV) based service delivery model, in response to the unprecedented traffic growth and an increasing customers demand for new high-quality network services. In NFV, telecom network functions are virtualized and run on top of commodity servers. Ensuring network performance equivalent to the legacy non-virtualized system is a determining factor for the success of telecom networks virtualization. Whereas in virtualized systems, achieving carrier-grade network performance such as low latency, high throughput, and high availability to guarantee the quality of experience (QoE) for customer is challenging.

In this thesis, we focus on addressing the latency challenge. We investigate the delay overhead of virtualization by comprehensive network performance measurements and analysis in a controlled virtualized environment. With this, a break-down of the latency incurred by the virtualization and the impact of co-locating virtual machines (VMs) of different workloads on the end-to-end latency is provided. We exploit this result to develop an optimization model for placement and provisioning of the virtualized telecom network functions to ensure both the latency and cost-efficiency requirements.

To further alleviate the latency challenge, we propose a multipath transport protocol MDTCP, that leverage Explicit Congestion Notification (ECN) to quickly detect and react to an incipient congestion to minimize queuing delays, and achieve high network utilization in telecom datacenters.

Keywords: NFV, telecom network virtualization, latency, virtualization, cloud computing, network congestion, QoS.
Acknowledgements

First of all, I would like to express my gratitude to my main supervisor Professor Anna Brunstrom for her tremendous mentor, patience, motivation, and immense knowledge. I would like to thank her for encouraging my research and for allowing me to grow as a research scientist. I would also like to thank my co-supervisors Karl-Johan Grinnemo and Javid Taheri for their insightful feedback on my research and offering support whenever I needed. Special thanks to Karl-Johan Grinnemo, who has been tirelessly reviewing and commenting on my work. Thank you all for your patience, motivation, encouragement, and constructive criticisms even if things do not turn out as expected.

I would like to thank all colleagues and friends from the Department of Computer Science at Karlstad University, in particular, the Distributed Systems and Communication Research (DISCO) group for maintaining a motivating and friendly research environment.

I am indebted to my parents, for encouraging me in all of my pursuits and inspiring me to follow my dreams. I salute you all for the selfless love, care, pain, and sacrifice you did to shape my life. To my late father, Boru Oljira, thank you for your love, motivation, inspiration, sacrifices, and for doing everything possible to put me on the path to greatness. Thank you to my mother, Dagitu Ameni, for her understanding and encouragement to pursue my dream even if it is time when she needs my care and support. My sisters, brothers, and uncles (Daniel Lulesa, Assefa Sori, and Damene Jilcha) deserve my wholehearted thanks as well. I am also grateful to my cousins, Birhanu Kebede, Roobe Begna, and Calchisa Desisa for their encouragement and support.

Last but not least, I would like to thank my wife Asmara Gada for her unconditional love, support, and understanding during this journey. I appreciate my daughter, Sinbo, for abiding my ignorance, and standing by me with patience and tolerance.
Contents

List of Appended Papers ix

Introductory Summary 1

1 Introduction 3

2 Background 5
   2.1 Virtualization .......................... 5
      2.1.1 CPU Virtualization .................. 7
      2.1.2 Network I/O Virtualization .......... 8
   2.2 Network Functions Virtualization (NFV) .... 9
      2.2.1 NFV Requirements and Challenges ..... 11
   2.3 Cloud Computing .......................... 12
      2.3.1 Cloud Computing and NFV ............ 13
   2.4 Datacenter Network Performance .......... 14
      2.4.1 Load Balancing ..................... 14
      2.4.2 Transport Protocols ................ 15

3 Research Questions 18

4 Contributions 19

5 Research Methodology 20

6 Summary of Appended Papers 22

7 Conclusions and Future Work 23

Paper I: Analysis of Network Latency in Virtualized Environments 31

1 Introduction 31

2 Related Work 33

3 Background 33
   3.1 VMWare ESXi Hypervisor ................... 34
   3.2 TCP and SCTP Overview .................. 35

4 Experimental Setup and Results 36
   4.1 Experimental Setup .......................... 36
   4.2 TCP and SCTP Performance in VM .......... 37
   4.3 Packet Delay Breakdown in Hypervisor .... 39
   4.4 End-to-end Latency with Co-located VMs 39
4.5 Packet Segmentation Offloading ........................................ 42

5 Conclusions and Future Work ........................................... 43

PAPER II:
A Model for QoS-Aware VNF Placement and Provisioning ........ 47

1 Introduction ................................................................. 47

2 Background ................................................................. 49

3 Model Formulation ........................................................ 53
   3.1 Objective Function .................................................... 55
   3.2 Constraints .............................................................. 55
      3.2.1 VNF Instance Constraint .................................... 55
      3.2.2 Resource Constraints ......................................... 56
      3.2.3 Flow Conservation Constraint ............................... 56
      3.2.4 Latency Constraints .......................................... 56

4 Performance Evaluation .................................................. 58

5 Related Works ............................................................. 62

6 Conclusions and Future Work .......................................... 62

PAPER III:
MDTCP: Towards a Practical Multipath Transport Protocol for Telco Cloud Datacenters 67

1 Introduction ................................................................. 67

2 Design ............................................................................ 69
   2.1 Fluid Model ............................................................. 70
   2.2 MDTCP Algorithm .................................................... 72

3 Evaluation of MDTCP Design Requirements ......................... 73
   3.1 Fairness at the Shared Bottleneck and Convergence ......... 73
   3.2 Congestion Balancing ................................................ 74
   3.3 Resource Pooling ...................................................... 76

4 Comparative Performance Evaluation .................................. 77
   4.1 Permutation ............................................................. 78
   4.2 Short flow Completion with Background Traffic ............. 80
   4.3 Benchmark Traffic .................................................... 82
5 Related work 84
6 Conclusions and Future Works 86

List of Appended Papers

1. Dejene Boru Oljira, Anna Brunstrom, Javid Taheri, and Karl-Johan Grinnemo. Analysis of Network Latency in Virtualized Environments. GLOBECOM’16, 4-8 December 2016, Washington, DC, USA.


Comments on my Participation

Paper I  The idea of the paper came from my supervisor Anna Brunstrom and co-supervisors (Javid Taheri and Karl-Johan Grinnemo). I was responsible for conducting the experiments and analysis, and writing the paper. Javid Taheri helped me in setting up the experiment and offered help whenever I had a problem. My supervisors have provided useful inputs and feedbacks on the results. Karl-Johan Grinnemo helped in proofreading, commenting, and revising the paper. Anna Brunstrom and Javid Taheri also revised and commented on the paper.

Paper II  The idea of the paper came from me, and my supervisors suggested to use the measurement results in Paper I. I developed the model, implemented and performed the analysis. My supervisors provided feedback and useful input on the implementation, analysis, and presentation of the results. I was responsible for writing the paper, and Karl-Johan Grinnemo did detailed proofreading, commenting and revising the paper. Anna Brunstrom and Javid Taheri also revised and provided constructive comments.

Paper III  The ideas of the paper came from Anna Brunstrom and Karl-Johan Grinnemo. I developed the congestion control algorithm and implemented in the simulator. My supervisors offered their inputs during our discussion during supervisor meeting. Karl-Johan Grinnemo helped me in debugging some problems I had during simulation. I wrote the first draft of the paper and Karl-Johan Grinnemo and Anna Brunstrom helped in reviewing, commenting and proofreading the paper.
Introductory Summary


1 Introduction

The proliferation of smartphones of various capabilities has resulted in unprecedented traffic growth and rich media applications both in the Internet and enterprise networks. With the emergence of the Internet of Things (IoT) and machine type communications (MTC), the Internet is expected to experience a 'traffic explosion'. Customers' demand for new, personalized, and high-quality network services has risen to unprecedented levels. The over-the-top (OTT) service providers are constantly innovating their infrastructure to optimize service provisioning and to enrich their service offerings to demanding users at a faster rate. On the other hand, the inflexibility of telecom network infrastructures has limited the ability of telecom service providers to expand their network, roll-out innovative network services, and generate revenue from the new digital economy [1].

The deployment of telecom network services and functions requires special proprietary hardware devices. This results in a static chain of network services which lacks the flexibility to cope with the dynamic user and quality of service (QoS) requirements. To overcome these challenges, telecom service providers have adopted Network Functions Virtualization (NFV) [2]. NFV involves virtualizing network functions to offer services with the speed, flexibility, and efficiency of the cloud while maintaining high carrier-grade quality of service levels [3]. The virtualized network functions (VNFs) can be deployed in real time at the desired locations in the operator’s network, something which opens up for an optimization of resource usage, reduces operational costs, and enables the faster offering of new services and the agility to support new business opportunities.

The NFV paradigm is also the main enabler of the forthcoming next-generation network (5G) to embrace network scalability, flexibility, agility, and programmability requirements [4]. The 5G network will exploit the virtualization technology to create virtual network slices atop of a shared physical infrastructure. The virtual networks, i.e., network slices, can be dynamically customized and tailored to the needs of applications, services, and customers using cloud computing technology. The 5G use cases such as telesurgery, smart metering, autonomous driving, remote operation of machinery, etc., require network connectivity with different QoS characteristics. The virtualization technology and software-defined networking (SDN) enable the instantiation of network slices customized and chained together programmatically to meet the needs of each application and service.

The benefit of NFV comes with some level of network performance trade-offs and may pose network QoS degradations as a result of virtualization [5, 6]. The scheduling of VMs for physical resources, context switching, packet processing in the virtualization layer, and delayed interrupt handling can result in significant delay overheads [7, 8, 9].

The consolidation of a large number of VMs on the same physical host also affects the performance of transport protocols such as the Transmission Control Protocol (TCP). When multiple VMs share the same physical CPU,
the scheduling latencies can increase the round-trip times (RTTs) for TCP connection to the orders of tens of milliseconds in the datacenter. As a result, CPU access latency dominates the RTT between two VMs, causing degradation of throughput and an increase in the end-to-end latency [10].

Telecom applications are characterized by high QoS requirements such as low latency, high throughput, and high availability. To maintain the QoS of telecom network services in a virtualized environment, comprehensive performance measurement studies are important. Such profiling and analysis help in understanding the overhead of virtualization, the interaction of different workloads and the resource sensitivity of different applications. As part of this thesis, we focus on such network performance measurements in a controlled virtualized environment, to better understand the effects of virtualization on the end-to-end latency of communicating VMs.

In addition to the inherent overhead of virtualization, the performance of virtualized telecom applications is affected by resource allocation, the number of co-located VNFs, and where the VNFs of a service chain (SC) is deployed. The VNFs of the SC are usually deployed in distributed telecom datacenters or points of presences (PoPs) close to the end users, to improve service experience, and involves the conflicting goals of efficient resource utilization to minimize cost and assuring acceptable QoS requirements such as end-to-end latency. In this thesis, we propose an optimization solution for the placement and resource provisioning of VNFs to address this problem.

Another cause of network performance degradation in virtualized telecom networks is network congestion. Network congestion has the effect of increased delays and low throughput. For example, it can severely deteriorate the performance of the emerging 5G applications (e.g., self-driving cars, virtual reality, telesurgery, factory automation, smart grid, etc.) which require high network reliability and an end-to-end latency as low as 1 ms [11]. Internet congestion can be overcome by deploying VNFs at distributed telecom cloud datacenters close to the end user. However, the delay caused by congestion inside the datacenter network remains a challenge.

The main source of latency inside a datacenter network is queuing delays caused by the contention of the switching fabric or the output interface. Existing datacenter transport protocols including TCP are not suited to simultaneously achieve low queuing delays, high network utilization, and resilience to network failures. To address this problem, we propose a multipath transport protocol which efficiently utilizes the multiple paths of the datacenter network topology, and detects incipient congestion and quickly adjusts its sending rate to mitigate queuing delays.

The remainder of the Introductory summary is structured as follows. Section 2 provides a background on technologies and concepts related to the thesis. Section 3 introduces the thesis research questions. Section 4 describes the scientific research contributions of the thesis. Section 5 describes the research methods adopted in this thesis. Section 6 contains the list of appended scientific publications. Finally, Section 7 provides conclusions and outlines future work.
2 Background

In this section, we present the underlying concepts and technologies that are investigated in this thesis. We provide a brief overview of virtualization, network functions virtualization (NFV) and cloud computing. In addition, we briefly describe datacenter network performance, particularly the roles of load balancing and transport protocols are explained.

Fig 1 puts the main concepts into perspective. It presents the NFV-based network deployment and service (both mobile and fixed networks). The virtualization technology is used to virtualize the physical infrastructures (e.g., compute, network, storage) of distributed telecom datacenters (cloud datacenters) enabling the consolidation of different network functions on the same physical hardware. The datacenter network interconnects the datacenter resources and enables communication between them. The operator’s network provides interconnection between the different datacenters and presents a coherent pool of resources. The management and network orchestration (MANO) module has a global view of the resources and manages the infrastructure resources and orchestration of the VNFs.

2.1 Virtualization

Virtualization is an abstraction of physical resources that enables multiple VMs to run on the same physical server. The virtual machine applications are
isolated from each other and unaware of the virtualization of the underlying resources. The abstraction of the physical resources is provided by a software layer called the hypervisor or the virtual machine monitor (VMM). The hypervisor manages the physical resources and schedules between the VMs. The VMs are configured with virtual resources such as virtual CPU (vCPU) and does not own physical resources (e.g., physical CPU [pCPU]).

A hypervisor is called Type-1 (see Fig 2a) when it directly runs on the physical hardware and Type-2 (see Fig 2b) when it runs on the host operating system (OS) [12]. A Type-1 hypervisor does not require a host OS and is sometimes called a bare-metal or native hypervisor [13]. Compared to a Type-1 hypervisor, the extra layer between the physical resources and the VM in a Type-2 hypervisor results in hardware inefficiency. The VMware ESXi [14], Xen [15], KVM [16], and Microsoft Hyper-V [17] are examples of Type-1 hypervisors whereas VMware Player, VirtualBox [18], and QEMU [19] are Type-2 hypervisors.

A hypervisor-based virtualization offers great flexibility since each VM is free to choose its own operating system independently of the host. However, hypervisors constitute a level of indirection where all resource accesses go via the virtualization layer, which may result in performance degradation. For example, the CPU scheduler may affect the overall performance of the VM(s). Unlike the CPU scheduler of the guest OS (G-OS) which schedules vCPU
time to processes, the CPU scheduler in the hypervisor layer allocates pCPU time to vCPUs based on their entitlement. The pCPU scheduling latency for vCPU and context-switching cost from the guest to the hypervisor to run the pCPU scheduler results in additional delay. The overhead is more visible on frequent context-switching, i.e., frequent scheduling and de-scheduling of a vCPU, and incurs a non-negligible and variable latency [20].

Container-based light-weight virtualization (see Fig 2c) is an alternative to hypervisor-based virtualization [21]. In this type of virtualization, the OS resources (e.g., files, system libraries, routing tables, etc.) are virtualized to create multiple execution environments called containers on top of the same OS kernel. Container-based virtualization has a limited flexibility since the host OS is shared between the guest instances. Linux containers (LXC) [22], Docker [23], and FreeBSD jails [24] are examples of container-based virtualization.

In this thesis, for our work on performance measurement and analysis (Paper I), we used a hypervisor-based virtualization platform. CPU and network I/O virtualization overheads predominantly contribute to virtualization delay. In the next subsections, we present the techniques used for CPU and network I/O virtualization and discuss the pros and cons of each scheme.

2.1.1 CPU Virtualization

There are three types of virtualizations based on how sensitive and privileged instructions are executed on the x86 system [25]. These are:

- **Full virtualization**: In this type of virtualization, the hypervisor provides CPU emulation to handle and modify privileged and protected CPU operations of the guest OS kernels. Userspace operations are run unmodified at the native speed. Full virtualization can incur large performance overheads since it uses emulation, but offers the best isolation and security for VMs, and simplifies migration and portability as the same G-OS instance can run virtualized or on native hardware.

- **Para virtualization**: This type of virtualization involves modification of the G-OS kernel to replace non-virtualized instructions with hypercalls that communicate directly with the hypervisor. The hypervisor also provides hypercall interfaces for other critical kernel operations such as memory management, interrupt handling, and timekeeping. Paravirtualization has lower virtualization overhead, however, the overhead may vary depending on the workload.

- **Hardware-assisted virtualization**: The latest generation of x86 CPU architectures have hardware extensions to support virtualization. The Intel Virtualization Technology (VT-x) and AMD-V allow running unmodified G-OS VMs without the overheads inherent in full virtualization. The new processors provide a new privilege layer in which the hypervisor can operate, leaving the OS privilege level for the G-OS.
2.1.2 Network I/O Virtualization

Achieving a carrier-grade network I/O performance is paramount to the success of NFV. In a virtualized system, the I/O operations traverse two separate I/O stacks: the G-OS TCP/IP stack and the virtualization layer (see Fig 3).

Most virtualization platforms implement two types of I/O virtualization called emulated and paravirtualized I/Os. Emulation of I/O devices interprets accesses to I/O registers and replicates the behavior of the corresponding hardware. Therefore, an emulated I/O device is an emulation of the native physical device (e.g., Intel e1000e). I/O access from the guest OS, in the form of memory-mapped I/O (MMIO) instructions, causes a context switch, VM exit, that returns the CPU to the host mode. On modern hardware, the cost of VM exit/VM enter pair and I/O emulation is 3–10 μs compared to 0.1–0.2 μs for I/O instructions on bare-metal [9]. A paravirtualized I/O device is also created by emulating a new virtual I/O device (e.g., VMXNET3, VIRTIO, Xenfront, etc.) designed for a virtualized environment. However, it requires a custom device driver in the G-OS. In a paravirtualized I/O device, the guest and the VMM exchange I/O requests in a shared memory region. Hence, it minimizes the number of I/O interrupts and VM exits compared to the emulated I/O.

Hardware enhancements have been proposed to overcome the hypervisor overhead in paravirtualized and emulated I/O devices. A Single-root I/O (SR-IOV) [26] interface is an extension to the PCI Express (PCIe) and creates virtual functions that share the resources of a physical function. With SR-IOV, multiple VMs can have direct access to a single physical NIC. The main drawback of hypervisor pass-through solutions such as SR-IOV is the lack of support for features such as VM snapshots and VM migration.
2.2 Network Functions Virtualization (NFV)

The growth of mobile data traffic fueled by the increased usage of smartphones, the continuous demand for new services, and competition from the disruptive OTT players have put the telecom service providers under intense pressure. Telecom networks are composed of proprietary hardware devices, which makes network expansion and launching of new services increasingly difficult.

NFV [1] was proposed by a group of telecommunications network operators to overcome these challenges. The main goal of NFV is to leverage virtualization technology to decouple the software implementation of network functions from physical network equipment so that the network applications can run as VMs on a general purpose server. The VNFs can be easily and dynamically instantiated and managed at any desired location within the operator’s network. The different network functions are executed as VNFs in different points of presences (PoPs) called NFV Infrastructure-PoPs (NFVI-PoPs) as shown in Fig 1. The NFVI-PoPs may be full-fledged datacenters, network nodes, or smaller sites with a limited amount of resources.

NFV aims to implement network functions as software-only entities that run over the NFV infrastructure (NFVI). To achieve this, the ETSI ISG NFV has defined a high-level view of the NFV framework [28]. The framework consists of three main functional blocks (see Fig 4): NFVI, VNFs, and NFV Management and Orchestration (MANO). The NFVI domain comprises of the physical resources (compute, network, and storage), and the software (a hypervisor) that makes the virtualization possible. It is an environment in which VNFs are deployed. The VNF block is a software implementation of a network function (NF) which has well-defined interfaces and functional behavior (e.g., virtualized residential gateway, DHCP servers, firewalls, etc.).

The NFV MANO [29] provides the functionality required for the provisioning of the VNFs, and the related operations, such as the configuration of the VNFs and the infrastructure the VNFs run on. It encompasses the manage-
ment, orchestration, and life-cycle management of physical and/or software resources that support the infrastructure virtualization, and the life-cycle management of the VNFs. The NFV MANO also includes databases that are used to store the information and data models which define both the deployment as well as lifecycle properties of functions, services, and resources. It focuses on all virtualization-specific management tasks necessary in the NFV framework. In addition, it defines interfaces that can be used for communications between the different components of the NFV MANO, as well as coordination with traditional network management systems such as Operations Support System (OSS) and Business Support Systems (BSS) so as to allow for management of both the VNFs as well as functions running on legacy equipment.

In an NFV framework, the end-to-end network service (e.g., mobile voice and data) is described by an NFV Forwading Graph (NF-FG) [27] as shown in Fig 5. The NF-FG is a graph of virtual links connecting VNF nodes describing traffic flows between the VNFs. One or more network forwarding paths, each of which defines an ordered sequence of VNFs and virtual links to be traversed by traffic flows matching certain criteria, can be built on top of an NF-FG. In NF-FG, the endpoints and VNFs of the network service are represented as nodes and correspond to devices (e.g., smartphones), applications (e.g., firewalls and load balancers), and/or physical server applications (e.g., CDN server). The endpoints are connected to VNFs by a wired/wireless network infrastructure. The virtualization layer abstracts the hardware resources of the NFVI, and the VNFs run on top of it. A VNF can be composed of different VNFs (e.g., VNF2 is composed of VNF2-A, VNF2-B, and VNF2-C) called VNF components (VNFC). The VNF instances or VNFC can be implemented on different physical resources, (e.g., servers with different capacity), and can be deployed in geographically distributed NFVI-PoPs (datacenters) as long as the end-to-end service performance and other policy constraints are met.
The capacity and performance of a network service depend on the state and attributes of the network resources allocated to its VNF, and virtual link instances. In order to satisfy the QoS requirements, the NF-FGs need to be built, considering the state of NFVI resources for VNFs (e.g., availability, load), and virtual link instances (e.g., bandwidth, delay, delay variation, packet loss). Since the NFVI resources are shared by different network services and their deployment constraints are different from each other, the NFVI need to be carefully scheduled for multiple network services to optimize their performance indicators [30].

NFV offers a number of benefits compared to the traditional network service approach. The use of commodity servers and switches, the consolidation of different network functions on the same platform, automated network management and control, and optimal resource provisioning have the potential to reduce capital and operational expenses. NFV also increases the speed and agility to support new business opportunities and improves return on investment. Network services can be highly flexible and dynamically scaled to address varying demands.

NFV is applicable to any data plane packet processing and control plane functions in mobile and fixed networks. ETSI has identified a number of examples which can benefit from NFV [28]. For instance, the service provider can take advantage of NFV by virtualizing the Customer Premises Equipment (CPE) and the LTE Evolved Packet Core (EPC) [31]. The NFs that constitute CPE (e.g., DNS server, Firewall, NAT, etc.) and EPC (e.g., the Mobility Management Entity [MME], Serving Gateway [S-GW], Packet Data Network Gateway [P-GW], and Policy and Charging Rules Function [PCRF]) can be virtualized and deployed in a shared NFVI, the operator’s cloud. Then, the virtualized functions can be easily updated or scaled in/out to respond to changes in demands with limited human intervention.

Other network function candidates for NFV are switching elements (e.g., routers), tunneling elements (e.g., VPN gateways), traffic analysis and inspection nodes (e.g., DPI, QoE measurements).

2.2.1 NFV Requirements and Challenges

NFV has gained an increasing attention both by academia and industry due to its anticipated benefits. To deliver these benefits, NFV must be designed to meet a number of requirements and challenges, including the following:

- **Performance**: In NFV, the NFs are deployed as virtualized functions on industry standard servers. Moreover, several VNFs are co-located for resource optimization. This results in performance trade-offs compared to running VNFs on a specialized hardware. Therefore, maintaining carrier-grade network performance (e.g., low latency, high throughput, and availability) is a challenge.

- **Management and Orchestration**: The NFV management system must consistently exploit the dynamics and flexibility of NFV to provide network
and service solutions. The NFs that provide a service to a customer may be scattered across different server pools. In such scenarios, achieving an acceptable level of orchestration on a per-service level, instantiation of all the required functions in a coherent and on-demand basis, and ensuring manageability are one of the challenges [32].

- **Resource Optimization**: The physical resources should be efficiently utilized to achieve the economies of scale expected from NFV. The default configuration may result in sub-optimal resource allocation and consumption [33]. Thus, efficient algorithms should be used to determine on to which physical resources (servers) network functions are placed, and be able to move functions from one server to another to achieve load balancing, energy saving, recovery from failures, etc.

- **Security, Privacy, and Resilience**: NFV-based services should not impair the security, privacy, and resilience of telecom networks. In case public clouds are adopted for deployment of VNFs, privacy is a main concern. The VNFs represent subscriber services and deploying them in public clouds may result in the transfer of personally identifiable information to the cloud. Since NFV delivers software enabled automated provisioning of network functions, it can also open security vulnerabilities such as automated network configuration exploits, orchestration exploits, malicious misconfiguration, and SDN controller exploits. Thus, ensuring the privacy and security requirements is a challenge [34, 35].

This thesis partially attempts to address two of the above challenges: the network performance and resource optimization. Paper I primarily focuses on the understanding of the network performance overhead of virtualization, particularly latency. Paper II relies on Paper I and proposes a solution to overcome network performance and resource optimization problem. The solution was demonstrated based on LTE EPC VNFs deployment and provisioning in telecom distributed NFVI-PoPs (datacenters). Paper III also aims at improving network performance and efficient utilization of datacenter network. Hence, it implicitly addresses the resource optimization problem in addition to mitigating latency and improving throughput.

### 2.3 Cloud Computing

Cloud computing is a model for enabling dynamic, online access to a shared pool of computing resources (e.g., networks, servers, storage, and applications) that can be rapidly provisioned and released with minimal service provider interaction [36]. It has the benefits of lowering operational cost, high scalability, i.e., services can be easily scaled in order to handle increases in demand, and ease of access to services over the Internet. Cloud computing employs virtualization technology to provide computing resources as a utility. It shares some features with grid computing, utility computing, and autonomous computing but also differs from them in other aspects [37].
There are three categories of cloud service models. The software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). The SaaS is the offering of applications over the Internet. In SaaS, the user does not require any hardware or software installation, and hence SaaS is operating system (OS) agnostic. Some examples of SaaS are Microsoft Office 365, Dropbox, and G Suite. PaaS implies providing hardware and software tools (e.g., OS, software development frameworks) to develop, run, and manage applications. The Google App Engine and Microsoft Windows Azure are examples of PaaS. IaaS is an on-demand provisioning of computing infrastructure including servers, storage, and networking hardware. Amazon Web Services (AWS) and Google Cloud Platform (GCP) are examples of IaaS.

On the basis of how it is deployed, a cloud service can be public, private, or hybrid. In a public cloud, services are offered to the general public. The cloud user is free of initial investment on infrastructure but lack fine-grained control on data, network, and security settings. Private clouds are services within the enterprise’s internal datacenter. Unlike public cloud services, the private cloud offers the highest degree of control over performance, reliability, and security. A hybrid cloud is a combination of public and private clouds. In a hybrid cloud, some of the services run in private cloud while the remaining part runs in public cloud. In that way, it provides better control and security over application data compared to public cloud, while still facilitating on-demand service expansion.

2.3.1 Cloud Computing and NFV

While cloud computing predates NFV, they have many features in common. Both NFV and cloud computing leverage virtualization to decouple software from hardware to consolidate virtual machines or VNFs for resource optimization. The NFV architectural framework can be mapped to the cloud service model. The physical and virtual resources in NFVI correspond to the IaaS cloud, while the services and VNFs in NFV are similar to the SaaS service model in cloud computing [3]

The flexibility of cloud computing, such as the rapid deployment of new services and ease of scalability, makes the cloud the best candidate to achieve efficiency and cost reduction in NFV. However, the telecom networks are different from the cloud computing environment. In particular, the telecom services have low latency, very high availability and reliability requirements. Therefore, the cloud needs to be adapted in order to achieve carrier-grade network performance requirements.

Our experiments for the network performance profiling (Paper I) was based on a private cloud testbed built on a VMware virtualization platform so that we have full control of the environment. There are no other workloads which interfere with the experiments. Unfortunately, we did not have the chance to conduct our analysis based on the typical telecom network functions since such applications are usually not readily available. Rather, we used custom applications which allow us to generate traffic scenarios similar to telecom
applications.

2.4 Datacenter Network Performance

From an NFV perspective, the datacenter (NFVI-PoP) is the premises where the NFVI-nodes (compute, storage, and networking resources) used to execute the network functions are deployed. Telecom operators build geographically distributed datacenters of different scale (small, medium, and large) to deliver their services with an improved user experience.

The datacenter network plays an important role in facilitating communication between datacenter servers. Thus, the network topology should be carefully designed as it affects the performance of datacenter applications. The order of the number of servers is one of the factors that influence the design of the network topology. For example, for a small datacenter with tens of servers, a simple tree-like topology may be suitable. On the other hand, high-capacity, scalable, fault-tolerant topologies (e.g., Clos, Jellyfish, BCube) need to be adopted if the number of servers is in the orders of few hundreds.

The datacenter network has distinct characteristics compared with the Internet, such as very low round trip times (RTT) in the orders of microseconds, very high data rates (Gb/s), regular topology, and latency generally matters far more than throughput. Thus, protocols that are commonly used in the Internet may not perform well in the datacenter.

The datacenter network is built from shallow-buffered commodity switches, which are shared by multiple ports. When many flows converge on the same egress port of the switch, network congestion may be experienced due to queue overflows.

The nature of the datacenter traffic also poses another network performance problem. Telecom network applications, i.e., the VNFs, generate a diverse mix of short (e.g., alarms and notifications, signaling traffic, etc.) and long (e.g., VNF migration traffic, profiling and analytics data) traffic flows. When long and short flows share the same queue, the packet drop on latency-sensitive short flows can severely degrade performance. Even if packets are not lost, the short flows experience increased latency when queued behind packets from the long flows.

Improving the network performance in the datacenter has been one of the research focuses in cloud computing. The research solutions aim at the efficient utilization of the datacenter network and network congestion mitigation. In the proceeding, we introduce load balancing and transport protocol approaches which are often used to achieve these goals.

2.4.1 Load Balancing

Congestion can be experienced in the datacenter network even when there are spare capacity elsewhere. The use of an algorithm that can evenly load balance the traffic among the available multiple paths of the datacenter topology can remedy the problem.
Equal Cost Multi-Path (ECMP) is the most commonly used routing strategy to load balance the datacenter network. It determines the path of a packet among all the shortest paths based on the hash value of a flow’s five-tuples. However, ECMP performs poorly primarily due to (i) its lack of global congestion information, and (ii) hash collision among elephant flows, i.e., when the hash values of two or more elephant flows collide, they are routed on the same link with reduced throughput although other paths may be completely free, and (iii) inability to differentiate flows [41, 42].

Several works have been proposed to address the limitations of ECMP. In Fastpass [43], each sender delegates control to a centralized arbiter to decide when (i.e., a timeslot) the packet should be transmitted, and what path it should follow to prevent assigning multiple packets to a link in a single timeslot. Hedera [44], MicroTE [45], and Mahout [46] re-route elephant flows to compensate for the inefficiency of ECMP by hashing them onto the same path. Presto [41] divides flows into equal-sized units of packets called flowcells at virtual switch (vSwitch) in the end-host and distributes them evenly to load balance the network. Presto also employs a centralized controller to react to occasional asymmetries in topology such as failures. CONGA [42] splits traffic into flowlets and use in-network congestion feedback mechanisms to estimate load and allocate flowlets to paths based on the congestion feedback.

### 2.4.2 Transport Protocols

A transport protocol plays an essential role to achieve the network QoS requirements in the datacenter. The major goals of a datacenter transport protocol are to guarantee data delivery at the highest possible rate and with the lowest latency to meet the demands of datacenter applications. To this end, a transport protocol that offers reliable, in-order data delivery is preferred over the unreliable and best-effort protocol.

The Transmission Control Protocol (TCP) [47] and Stream Control Transmission Protocol (SCTP) [48] are the two standard connection-oriented transport protocols that can be used for reliable data delivery in the datacenter. TCP is a byte-stream, in-order data delivery transport protocol. In contrast, SCTP is a message-oriented protocol and ensures reliable, in-sequence transport of messages. Both TCP and SCTP achieve a reliable end-to-end transfer by employing a positive acknowledgment (ACK) and re-transmission of the lost packet.

SCTP shares the flow and congestion control features of TCP, but also has a number of unique attributes, such as support for multi-homing and multi-streaming [48]. Multi-homing allows the two endpoints of a connection to declare multiple interfaces (IP addresses) and provides an alternate route for the data in case the interface in use fails. The SCTP multi-streaming feature allows creating multiple data streams that can be independently sequenced and delivered. Thus, it lessens head-of-line blocking since the loss of a message in one stream will only affect delivery within that stream.

In Paper I, we study the performance of TCP and SCTP in our cloud testbed, primarily to understand how end-to-end latency is affected when these
protocols are used in a virtualized environment. There are already existing works which analyze the effects of virtualization on the behavior of the TCP protocol [10]. Hence, our analysis focuses on the application-level response time rather than the RTT variation of the protocols.

A congestion control algorithm is the main engine of a transport protocol. It is the mechanism by which a transport protocol controls congestion and keep the traffic below the capacity of the network. Several congestion control algorithms have been proposed for TCP, most of which consists of two phases: a slow-start phase and a congestion-avoidance phase. These phases determine the sending rate of TCP. In the slow-start, the congestion window is doubled every RTT, leading to an exponential increase of the sending rate. When the congestion window exceeds the slow-start threshold, TCP is in a congestion avoidance phase, and the congestion window increases by one segment once per RTT.

The TCP congestion control algorithm treats packet loss (detected through a timeout or three duplicate ACKs) as an indication of congestion in the network. Employing packet loss as a congestion signal may suit applications with little or no sensitivity to delay or loss of individual packets. However, these mechanisms can significantly impact the performance of latency-sensitive applications (e.g., interactive audio or video, signaling traffic, etc.). TCP determines its appropriate sending rate by gradually increasing its congestion window until it experiences a dropped packet, this results in a queue build-up at the bottleneck router. Packet discarding policies at the router that are not sensitive to the load of individual flows (e.g., tail-drop) may drop some of the packets of latency-sensitive flows. Moreover, such drop policies may lead to synchronization of loss across multiple flows [49].

Explicit Congestion Notification (ECN) is an alternative to loss-based congestion signaling. ECN congestion signaling is a method that allows network nodes (e.g., routers and switches) to propagate a congestion signal (i.e., Congestion Experienced [CE] codepoint) embedded in the IP packet header to the ECN capable transport if it would have otherwise dropped the packet. This has the potential of reducing the impact of loss on latency-sensitive flows. In ECN, an active queue management (AQM) mechanism at the network nodes detect congestion before queue overflows and provide an indication of congestion [49].

The regular TCP (e.g., TCP New Reno) essentially treats the ECN congestion signals as packet drop, i.e., the source TCP is required to halve its congestion window. In essence, an ECN-capable TCP reacts to the presence rather than the extent of congestion. Reducing the congestion window by half results in under-utilization and loss of throughput in a high-speed datacenter environment. Datacenter transport protocols such as DCTCP [39], and its derivatives (e.g., D2TCP [50]) address this problem by adopting an instant queue length based ECN congestion signaling which allows to react to the extent of congestion.

In DCTCP, the network switches mark the packet when the queue length is greater than a configured marking threshold. When an ECN-marked packet
is received by the DCTCP receiver, it sets an ECN-Echo flag in the ACK packet to convey the flag to the DCTCP sender. The sender maintains an estimate of the fraction of packets marked. Then, the sender cuts its congestion window in proportion to the fraction of marked packets, unlike TCP which always reduces its window size by a factor of 2 in response to a marked ACK. A DCTCP sender reacts to congestion as soon as the queue length exceeds the threshold value, which keeps queuing delay very low. DCTCP shares the other features of TCP such as slow start, additive increase in congestion avoidance, and fast recovery from packet loss.

Multiple network paths are common between pairs of endpoints on the Internet [51, 52] and datacenters have many redundant paths between servers. Resource usage within the network would be more efficient were it possible for these multiple paths to be used concurrently. In single path protocols such as TCP, DCTCP, etc., a flow cannot be load balanced across multiple paths within the network since it results in packet reordering. The TCP protocol misinterprets the reordering as congestion and unnecessarily slows down.

Multipath protocols such as parallel TCP (pTCP) [53], Concurrent Multipath Transfer for SCTP (CMT-SCTP) [54], and Multipath TCP (MPTCP) [55] were proposed with the goal of improving the overall throughput for a flow and to provide a better reliability in case of path failures in a network that provides multiple paths between endpoints. Among these protocols, MPTCP has been already standardized by IETF [55].

MPTCP is an extension of TCP that has been implemented in several platforms such as Linux Kernel, Apple iOS, macOS, and FreeBSD. MPTCP sets up multiple subflows to simultaneously transfer data across the available multiple paths. An MPTCP session starts with an initial subflow, which is similar to a regular TCP connection. Additional subflows can be established after the first subflow is set up. Each subflow is similar to a regular TCP connection but is bound to an MPTCP session. The MPTCP protocol uses a coupled multipath congestion control [56] that links the congestion window of each subflow and dynamically adjust the congestion window of each subflow according to the congestion information of the path.

MPTCP achieves robustness and resilience to link failures, provides seamless handovers over different networks for communication with smart devices equipped with multiple access interfaces such as Ethernet, WiFi, and 3G/4G. MPTCP also exploits the available network resources by load balancing traffic over all equal cost paths of the datacenter network. However, MPTCP performs poorly in terms of latency in a datacenter network since it, in the same way as single-path TCP, uses packet losses as a congestion signal.

Recently, a new multipath-aware datacenter transport protocol, NDP [57], has been proposed. NDP uses receiver-based pacing to balance the incoming rate to the per-interface line rate and maintains very shallow switch buffers to overcome queuing delay. When the switch queue fills, the switches trim packets to headers and priority forward the headers. NDP’s design impacts the whole network stack including the switch, routing, and is completely a new transport protocol. Hence, NDP may face deployment challenges. NDP has
also been shown to perform poorly for short messages in terms of latency since it employs only two priority levels with static assignment, and also significantly hurt performance at high network load as it wastes bandwidth at receiver downlink [58].

This thesis partly aims at designing a multipath transport protocol which leverages the features of MPTCP and DCTCP, to achieve low latency and high throughput requirements of datacenter applications.

3 Research Questions

The main research objective of this thesis is

*To mitigate latency in virtualized telecom networks to satisfy the low latency requirement of applications and services.*

In order to achieve this objective, we identify the following research questions that we attempt to answer:

RQ1 *What is the delay overhead incurred by virtualizing telecom network functions?*

In a virtualized system, it is reasonable to anticipate some overhead due to the virtualization layer, compared to a non-virtualized system. Telecom network services have strict network performance requirements (e.g., low latency). Successful virtualization of telecom network functions starts with a comprehensive understanding of where the overhead comes from. The network performance of telecom network functions can also be affected by other network functions or virtualized applications competing for the same resources. The analysis of the network performance of co-located VNFs is key in resource allocation of VNFs to maintain similar network performance as in a non-virtualized system. We performed extensive profiling and analysis in a controlled cloud testbed, to further the understanding of the problem.

RQ2 *How can we ensure both the latency requirements of applications and cost-efficient operation in virtualized telecom networks?*

Achieving both the QoS requirements (e.g., end-to-end delay) and minimizing the cost of operation is a complex process. The network performance and the cost are affected by factors such as the traffic load, the resources allocated to VNFs, the number of instantiated VNFs, and the deployment sites of VNFs service chains. To guarantee the end-to-end latency, the VNFs should be provisioned with sufficient resources (e.g., CPU cores, network bandwidth), be deployed close to the user, and the overhead of virtualization should be taken into account. On the other hand, to minimize the cost of operation, resource over-provisioning should be avoided. An optimization technique can be used to meet both
the latency and cost requirements. The optimization model enables the relationship between the different entities to be formulated and optimized on the basis of the defined objective function. Thus, we propose an optimization model for placement and provisioning of VNFs.

**RQ3 How can we improve network performance in telecom datacenters?**

The telecom VNFs are deployed in cloud datacenters and are expected to support services with strict network QoS requirements such as low latency, high reliability, and high throughput. To meet the QoS of these services, network performance degradation in telecom cloud datacenters needs to be mitigated. For example, the network topology needs to offer higher aggregate bandwidth and robustness by creating multiple paths. Improving and/or developing a transport protocol that simultaneously delivers high bandwidth and utilization at low latency is equally important. In the datacenter, the main cause of network performance degradation is network congestion. Thus, a transport protocol which quickly detects an incipient congestion as well as efficiently utilizes the datacenter network capacity is desired. We develop a multipath transport protocol to achieve these goals.

### 4 Contributions

This thesis provides an understanding of the challenges of ensuring carrier-grade network performance in virtualized telecom networks with an emphasis on network latency and proposes solutions to address some of the problems. The main contributions of the thesis are:

1. **A better understanding of the delay overhead of virtualization.**

   We contribute to the understanding of the network performance effects of virtualization, particularly latency, by conducting controlled measurements and packet trace analysis at the virtualization layer. In contrast to previous works, we identify the delay overhead of the components of the virtualization layer, perform a thorough analysis of the performance of transport protocols such as TCP and STCP, and study the impact of kernel optimizations on end-to-end latency in a virtualized environment. A detailed analysis of the interaction of co-located virtual machines of different workloads on the end-to-end latency is also provided. In this contribution we address RQ1.

2. **A solution to ensure both the latency requirements of applications and cost-efficient operation in virtualized telecom networks.**

   One of the goals of virtualizing telecom network functions is to reduce cost by efficiently utilizing resources. Thus, several virtual network functions are deployed on the same physical hardware. The interaction of
the virtualized network functions of various workloads competing for resources and the overhead of virtualization can result in QoS degradations. To overcome this challenge, we propose a way to optimally place VNFs and provision resources, something which answers RQ2. Although several works exist to address this problem, our approach differs from these works in that it provides a fine-grained end-to-end latency model which also accounts for the delay overhead of virtualization as one of the constraints to ensure the delay guarantees of applications.

3. A multipath transport protocol to improve network performance in telecom datacenters.

In NFV, the virtualized telecom network functions are deployed at telco cloud datacenters of different sizes. A carrier-grade network performance such as low latency should be maintained even if the VNFs run in virtualized datacenter infrastructure. The deployment of VNFs at distributed datacenters close to the end-user is believed to reduce propagation delay and Internet network congestion. However, network congestion in the datacenter network can still result in a significant increase in latency and throughput degradation. To mitigate the problem, we developed a multipath transport protocol which can quickly detect and react to congestion. Unlike existing multipath protocols, our solution exploits Explicit Congestion Notification (ECN) to quickly detect an incipient congestion and queue overflows to reduce latency. The protocol also efficiently utilizes the multiple paths of the data center topology to achieve high network throughput and can be safely deployed with regular DCTCP. This contribution addresses RQ3.

5 Research Methodology

Computer Science (CS) is a systematic study of algorithmic processes that describe and transform information [59, 60]. The CS field is a combination of science, engineering, and mathematics [59]. The study of information processes in CS follows a scientific paradigm of forming hypothesis and testing through experiments. The engineering aspect is tied to the design and development of complex hardware and software systems. The mathematical nature stems from the use of mathematics to build theories of computation and design of computational systems [60].

Scientific research is described by using rigorous methods for achieving a new knowledge. Scientific methods are an iterative cycle of literature review, problem formulation, hypothesis description, and analysis.

In CS, four scientific methods are commonly used: experimental, analytical, simulation, and emulation [61]. The experimental method involves the use of experiments with real-world implementations and measurements. The real world experiments have the lowest level of abstraction, but it is often difficult to design the experiment and difficult to control environmental factors. In
the analytical/theoretical approach, a system is represented by mathematical models or theories. In the simulation method, a model of a real system is implemented as computer programs. A simulation method offers the possibility to study complex systems, and investigate systems outside the experimental domain. Computer simulations involve detailed features of the underlying system, but are often based on assumptions and artificial modeling to achieve a certain degree of accuracy. Emulation method is a middle ground between real measurements and simulations. Some of the components of the experiment are abstracted while others run in a real environment.

In this thesis, all methods except emulation are exploited. For the measurements and analysis (Paper I), we mostly relied on testbed experiments, since the other methods impose complexity and abstraction to design a true representative of a virtualized environment. For example, using emulation adds another level of abstraction and results in lack of accuracy. There are several cloud simulation tools to study the network performance of cloud computing systems. However, a simulation method is not suitable for the problem we want to address in Paper I since modeling these kinds of problems requires different simplifying assumptions. A real measurement allowed us to control workload generation, resource configuration, traffic capture, and observe the interaction between different parameters. However, there were also several challenges during the experiment. First, the VMWare ESXi hypervisor source code is not open source, and hence it was difficult to reason about some of the observations during the experiment. The other challenge was related to traffic capture. We used a tracing tool provided by VMware, but the tool sometimes became unresponsive during the capture process. As this tool is not open source, it was also difficult to analyze how it might have impacted the profiling.

For Paper II, we used the analytical method and simulation. A mathematical model obviously fits due to the nature of the problem, and simulation is the straightforward method to verify the model. Real experiments are challenging for this problem since it requires both financial and time investments. It can take a long time to build a distributed cloud environment and validate the model. Emulation might have been used in Paper II, but it still takes time to build the experimental environment, implement and validate the model. One of the challenges in Paper II was the validation of the model. Since the model was an extension of a previous model, we were interested to validate our model with the previous one. However, the data used to analyze the previous model was not readily available.

In Paper III, simulation and mathematical modeling are used. This work is in its early stage, thus it makes sense to start with the analytical model and then verify with simulation. However, we intend to implement the protocol and conduct real-world experiments using actual traffic scenarios. We also aim to exploit an emulation approach after the protocol is implemented since it will be easier to design datacenter topologies and provides the flexibility to tune parameters. The main problem that we encountered in Paper III was the lack of a fully functional MPTCP implementation in the most commonly used network simulators such as NS-2 and NS-3. We intended to extend our solu-
tion based on the patches and implementations of MPTCP available in NS-2 and NS-3 since the simulators are well developed and have rich features to perform simulations. However, the MPTCP implementation was not functioning properly, so we finally resorted to the htsim network simulator.

6 Summary of Appended Papers

This section consists of a summary of the appended papers and their main contributions.

Paper I – Analysis of Network Latency in Virtualized Environments

In this paper, we perform controlled experiments to study the overhead of virtualization, its impact on the end-to-end latency of virtualized applications and performance of transport protocols, and the interaction of VMs of different workloads. By analyzing packet traces in the virtualization layer, the packet delay breakdown of a virtualization layer is obtained. The delay overhead of virtualization results in an increase of the response time of applications compared to the non-virtualized system. However, no impact is observed on the performance of transport protocols. The result also indicates an increase in the response time of latency sensitive VMs when deployed with a higher number of other VMs or when the co-located VM(s) are throughput sensitive.

Paper II – A Model for QoS-Aware VNF Placement and Provisioning

In this paper, we propose an optimization model for the placement of VNFs that ensures both the latency requirement of telecom network applications and cost efficiency. A related VNFs placement model is extended by a fine-grained latency model. We formulate the end-to-end latency of VNFs service chain including the virtualization overhead obtained from the measurement data in Paper I. The end-to-end latency is used as one of the constraints for the optimization model. We evaluated the model with a simulated network, and it indeed converges faster. The model deploys the VNFs on cheaper nodes while the end-to-end latency is guaranteed. We compared the original model with and without considering the overhead of virtualization. Our results show that when the virtualization delay is not included in the end-to-end latency, the placement result may violate the latency requirement.

Paper III – MDTCP: Towards a Practical Multipath Transport Protocol for Telco Cloud Datacenters

This paper proposes a multipath transport protocol MDTCP which employs ECN to quickly detect and react to congestion in telco cloud datacenters. MDTCP builds on MPTCP but uses DCTCP subflows rather than regular TCP subflows. Hence, the MDTCP subflows react to congestion signals in the same way as in regular DCTCP. However, the subflows window increase
is coupled. The coupling parameter is different from that of MPTCP and is obtained by solving the fluid model of MDTCP and DCTCP. The evaluation of MDTCP with various simulated traffic scenarios indicates that it achieves higher network utilization than existing single path protocols such as DCTCP and NDP, and than the multipath protocol MPTCP. MDTCP also improves latency compared to MPTCP, and performs as good as DCTCP and NDP except in some scenarios (e.g., incast communication).

7 Conclusions and Future Work

This work aims at addressing the latency challenge of virtualizing telecom network functions to ensure the latency requirements of telecom network services and applications. The thesis approaches the problem in three ways: understanding the impact of virtualization, devising a solution to mitigate the effect of virtualization, and developing solutions that can further improve the network performance of telecom networks.

To understand the delay overhead of virtualization, comprehensive measurements and analysis are performed in a controlled virtualized environment. The impact of virtualization on the performance of transport protocols, and the interaction of VMs of different workloads on the end-to-end latency are investigated. The study primarily resulted in the delay breakdown of the overhead of virtualization and an understanding of the impact of co-locating VMs of different workloads on the end-to-end latency.

The thesis then answers the question of how to deal with the delay overhead of virtualization and minimize the interference between co-located VMs to satisfy the latency requirements of applications. To overcome this challenge, a placement and provisioning optimization model are proposed. The solution deploys and allocates VNFs at cost-efficient locations in distributed telco cloud datacenters or PoPs while ensuring the latency requirement. To guarantee the latency requirement, a fine-grained end-to-end latency model, which also includes the delay overhead of virtualization, is considered as one of the constraints of the optimization model. The optimization model also overcomes the interference of co-located VNFs since the latency model is dependent on the traffic loads of VNFs service chain.

The VNFs of telecom network functions are deployed in telco cloud datacenters. Hence, an improved network performance in the datacenter network plays a significant role to ensure the latency requirements of telecom applications. Network congestion is the main reason for increased delay in a datacenter network. In this thesis, we develop a multipath transport protocol, MDTCP, to mitigate network congestion as well as to efficiently utilize the datacenter network capacity.

We have proposed solutions which aim to mitigate latency in a datacenter. However, the MDTCP protocol is yet to be implemented and tested in real networks. Hence, we aim to implement MDTCP in the Linux Kernel and carry out a real-world evaluation of the protocol. The MDTCP protocol can also be used over the Internet to ensure the end-to-end reliability and low latency
in 5G networks. The challenge is the lack of support for ECN at various network nodes in the Internet. Therefore, in our future work we will also focus on adapting MDTCP so that it works on the Internet, and integrate it with the Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service architecture.

References


Telecom Networks Virtualization
Overcoming the Latency Challenge

Telecom service providers are adopting a Network Functions Virtualization (NFV) based service delivery model, in response to the unprecedented traffic growth and an increasing customers demand for new high-quality network services. In NFV, telecom network functions are virtualized and run on top of commodity servers. Ensuring network performance equivalent to the legacy non-virtualized system is a determining factor for the success of telecom networks virtualization. Whereas in virtualized systems, achieving carrier-grade network performance such as low latency, high throughput, and high availability to guarantee the quality of experience (QoE) for customer is challenging.

In this thesis, we focus on addressing the latency challenge. We investigate the delay overhead of virtualization by comprehensive network performance measurements and analysis in a controlled virtualized environment. With this, a break-down of the latency incurred by the virtualization and the impact of co-locating virtual machines (VMs) of different workloads on the end-to-end latency is provided. We exploit this result to develop an optimization model for placement and provisioning of the virtualized telecom network functions to ensure both the latency and cost-efficiency requirements.

To further alleviate the latency challenge, we propose a multipath transport protocol MDTCP, that leverage Explicit Congestion Notification (ECN) to quickly detect and react to an incipient congestion to minimize queuing delays, and achieve high network utilization in telecom datacenters.

ISSN 1403-8099
LICENTIATE THESIS | Karlstad University Studies | 2018:24