Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. P. Tran-Gia

Design and Evaluation of Components for Future Internet Architectures

Dominik Werner Klein

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 1/14

Würzburger Beiträge zur Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr.-Ing. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Am Hubland
D-97074 Würzburg

Tel.: +49-931-31-86630
Fax.: +49-931-31-86632
email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage vom Autor.
Gesetzt in LaTeX Computer Modern 9pt.

ISSN 1432-8801

Design and Evaluation of Components for Future Internet Architectures

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius–Maximilians–Universität Würzburg

vorgelegt von

Dominik Werner Klein

aus

Schweinfurt

Würzburg 2014

Eingereicht am: 06.12.2013
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr.-Ing. P. Tran-Gia
2. Gutachter: Prof. Dr. habil. M. Zitterbart
Tag der mündlichen Prüfung: 04.02.2014

Danksagung

Mit der erfolgreichen Disputation am 4. Februar 2014 ist fast genau fünf Jahre nach dem Abschluss meines Diploms nun auch die Promotion in der Informatik nahezu abgeschlossen. In dieser Zeit habe ich wertvolle Einblicke in die wissenschaftliche Tätigkeit gewonnen und bin an positiven, aber auch an weniger erfreulichen Erfahrungen gewachsen. Mit dem Abschluss der Promotion beginnt für mich ein neuer Lebensabschnitt und es ist daher an der Zeit, einmal Danke zu sagen.

Ein besonderer Dank gilt natürlich meinem Doktorvater Prof. Phuoc Tran-Gia, der mir nach meinem Diplom die Möglichkeit gegeben hat, mein Studium mit der Promotion fortzuführen. Prof. Tran-Gia bietet den Mitarbeitern an seinem Lehrstuhl ein ideales Umfeld für die Promotion und hat daher einen enormen Beitrag zu meiner wissenschaftlichen Entwicklung geleistet. Abseits der Projekt- und Lehrtätigkeiten hat er mir die Teilnahme an vielen Konferenzen und Workshops ermöglicht, so dass ich meine Arbeiten anderen Forschern präsentieren und mit ihnen diskutieren konnte.

Ebenfalls danken möchte ich Prof. Martina Zitterbart dafür, dass sie sich für meine Forschung interessiert hat und das Zweitgutachten für meine Doktorarbeit angefertigt hat. Darüber hinaus möchte ich mich bei Prof. Frank Puppe für den Vorsitz der Prüfungskommission und bei Prof. Frank Steinicke für die Rolle als Prüfer bedanken.

Neben Prof. Tran-Gia haben meine beiden Betreuer während der Diplomarbeitszeit, Prof. Michael Menth und Matthias Hartmann, ebenfalls einen sehr wichtigen Einfluss auf meine wissenschaftliche Entwicklung gehabt. Von beiden habe ich sehr viel in Bezug auf das wissenschaftliche Arbeiten gelernt und

bin ihnen zu großem Dank verpflichtet. Des Weiteren möchte ich an dieser Stelle auch Prof. Kurt Tutschku danken, der mir nach dem Studium die Gelegenheit gegeben hat, im Rahmen des Euro-NF Projektes an seinem Lehrstuhl in Wien erste Erfahrungen als wissenschaftlicher Mitarbeiter zu sammeln.

In meiner anschließenden Zeit am Lehrstuhl in Würzburg durfte ich mit vielen netten Menschen zusammenarbeiten und dank der freundlichen Atmosphäre unter den Kollegen, nahm ich die langen Autofahrten immer gerne auf mich. Ganz besonders bedanken möchte ich mich hier bei Dr. Thomas Zinner für die enorme Unterstützung und das Feedback zu meiner Dissertation. PD Dr. Tobias Hoßfeld danke ich für wertvolle und kritische Kommentare zu Vorträgen sowie für lehrreiche Lektionen beim Kickern. Meinen Kollegen Michael Jarschel, Matthias Hirth und Christian Schwartz danke ich abseits der sehr angenehmen Zusammenarbeit am Lehrstuhl für zahlreiche kurzweilige und sehr unterhaltsame Besprechungen im Rahmen der Arbeitsgruppe für gepflegte virtuelle Abenteuer. Weiterhin danken möchte ich David Hock für den gegenseitigen Austausch von Erfahrungen während der gemeinsamen letzten Phase der Promotion sowie Florian Wamser für interessante Diskussionen und das Brainstorming während der Skizzierung von Projektanträgen. Steffen Gebert danke ich für die Tipps zu Git, Michael Seufert für die wertvollen Details zu sozialen Netzen und Valentin Burger für die zuverlässige Organisation der Skitouren. Zusätzlich danke ich meinen Studenten Stanislav Lange und Kathrin Borchert für die sehr gute Arbeit im Rahmen des COMCON Projektes.

Dank sagen möchte ich auch meinen ehemaligen Kollegen, Dr. Simon Oechsner und Dr. Frank Lehrieder, für die lehrreichen Gespräche im Kaffeeraum, Dr. Robert Henjes und Dr. Daniel Schlosser für die administrative Unterstützung und Hilfe sowie Dr. Dirk Staehle für Tipps zum Verfassen von Forschungsanträgen. Meiner ehemaligen Mitbewohnerin Dr. Barbara Staehle danke ich für die angenehme Büroatmosphäre und unzählige Tipps in Bezug auf die richtige Verwendung von Matlab. Dr. Rastin Pries danke ich für die Unterstützung während meines ersten Industrieprojektes und für meine erste Geocaching Tour während der Dienstreise in Ottawa. Weiterhin danken möchte ich Dr. Michael Duelli für

die Einarbeitung in das COMCON Projekt und die denkwürdige Dienstreise in Kaiserslautern.

Neben der fachlichen Unterstützung war auch die organisatorische Unterstützung durch Frau Förster und durch ihre Nachfolgerin Frau Wichmann sehr wichtig und ich möchte mich hierfür bei beiden recht herzlich bedanken.

Abschließend gebührt mein Dank meiner Familie und Freunden, die mir in all den Jahren Beistand leisteten und mich immer unterstützten. Meinen Eltern Claudia und Georg möchte ich besonders dafür danken, dass sie mir das Studium ermöglichten und somit den Grundstein für meine Promotion gelegt haben. Des Weiteren danke ich meinen beiden Schwestern Rebekka und Carina sowie meinen beiden Omas Eleonore und Maria. Auch bedanken möchte ich mich bei meinen Schwiegereltern Ulrike und Rainer für die geleistete Unterstützung und das Anvertrauen ihres wertvollsten Schatzes, der Person, der ich am meisten Dank schulde. Meine Frau Katharina hat mich ohne Ausnahme in meiner Arbeit bestärkt und hat mir immer einen Platz gegeben, an dem ich mich zurückziehen und abschalten konnte. Dafür meinen tiefsten Dank.

Contents

1 Introduction

Today's Internet architecture was not designed from scratch but was driven by new services that emerged during its development. Hence, it is often described as patchwork where additional patches are applied in case new services require modifications to the existing architecture. This process however is rather slow and hinders the development of innovative network services with certain architecture or network requirements. Currently discussed technologies like *Software-Defined Networking (*SDN) or *Network Virtualization (*NV) are seen as key enabling technologies to overcome this rigid best effort legacy of the Internet. Both technologies offer the possibility to create virtual networks that accommodate the specific needs of certain services. These logical networks are operated on top of a physical substrate and facilitate flexible network resource allocation as physical resources can be added and removed depending on the current network and load situation. In addition, the clear separation and isolation of networks foster the development of application-aware networks that fulfill the special requirements of emerging applications. *Application-Aware Networking (*AAN) enables dynamic networks that adapt to satisfy the needs of hosted applications, which is in contrast to traditional approaches and provisioning methods.

In future virtualized environments, there is no single architecture to deal with various services and their specific needs. Instead, a stable physical substrate accommodates highly flexible virtual networks. These networks exist side-by-side and enable adaptations towards the specific needs of various use cases. Within the G-Lab [18] project *COntrol and Management of COexisting Networks (*COMCON) [19], several such use cases were discussed for virtual networks [20, 21]. The most promising use case is denoted as service component

mobility. Services hosted on *Virtual Machines (*VMs) follow their mobile users so that both the access latency and the network resource consumption is reduced. In [22], service mobility was demonstrated inside a campus network for an interactive multi-player game. Another use case involving video streaming [23] was discussed in the European *Seventh Framework Program (*FP7) project *Architecture and Design for the Future Internet (*4WARD) [24]. Especially the video streaming use case is highly relevant since video streaming is responsible for the largest fraction of the global Internet traffic [25]. However, there are several open challenges regarding service component mobility for video streaming and in this work, we detail those challenges and discuss possible solutions.

Service component mobility is realized by migrating the hosting VMs, e.g., between virtual networks of different data center locations. The addressed challenges related to VM migrations itself are twofold. First, this thesis investigates the specific data transmission requirements of VM migrations and develops an optimized architecture to schedule VM migrations between connected virtual networks. Second, we detail the requirements of VM migrations with respect to the underlying network and evaluate, which available solution is most suitable to enable service component mobility.

Providing that the consuming endpoint is also mobile and roams between heterogeneous access networks, a suitable mobility or flexible routing architecture is required so that the video streams are redirected without perceivable interruptions. The current inter-domain naming and routing architecture however is not designed for such a scenario and also experiences scalability issues due to mechanisms like multihoming or traffic engineering. Hence, a new flexible routing architecture is required that supports the given mobile video streaming scenario. Among others, the *Locator/ID Separation Protocol (*LISP) has received the most attention and also offers a suitable mobility extension. However, the mobility extension has several drawbacks related to control and data plane and this thesis details those challenges and presents possible solutions.

During the operation of virtual networks, a continuous application-aware resource management is required which includes monitoring of the user-perceived

quality as well as suitable control mechanisms to adapt the virtual network according to different load situations. However, to get a reasonable estimate of the user perceived quality for video streaming applications, it is not sufficient to just measure QoS parameters and map them to the perceived quality. The mapping highly depends on the video content, the used video encoder, and the chosen settings for the encoding process and hence, such an approach does not offer the required accuracy. Instead, video content and used video encoder settings must be include so that a reasonable accuracy can be achieved. Hence, this thesis presents a network-based video quality monitoring solution that uses precomputed error information to improve the accuracy of the monitoring in the network.

1.1 Classification from Scientific Viewpoint

The scientific background for this work is a future virtualized network architecture that provides application-specific networks that are tailored towards a certain application with unique requirements. Such an architecture has been introduced in the European FP7 project 4WARD [24] and which was further enhanced in the G-Lab [18] project COMCON [19]. The architecture comprises a business role model which contains four distinct roles. Deploying application-specific networks in such a virtualized architecture involves several issues and challenges and the overall goal of this work is to outline some of these challenges on the different layers of the architecture and to discuss possible solutions. In the remainder, we first describe the COMCON architecture and then detail the considered application scenario for this work and the inherent challenges.

1.1.1 Scenario Description

Subsequently, we first introduce the role model introduced with the COMCON architecture and present background information related to the creation and operation of virtual networks. Afterwards, the considered use case is explained and inherent challenges are presented.

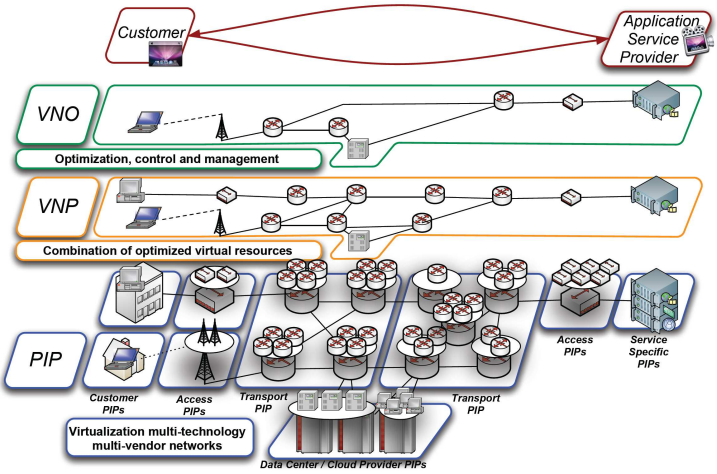


Figure 1.1: COMCON virtual network architecture and business role model [26].

Business Role Model

An important aspect for future virtualized environments is that the classical provider customer relationship is changing and new business roles become important. In the COMCON architecture, four main business roles are distinguished as shown in Figure 1.1.

On the lowest layer in the role model, *Physical Infrastructure Providers (*PIPs) run and manage the physical resources and create virtual resources on top of the physical substrate. These virtual resources are offered to customers which combine these resources to form virtual networks. The specific properties of virtual resources are stipulated with *Service Level Agreements (*SLAs) and PIPs deploy monitoring and control components to ensure that the contracted agreements are fulfilled. In addition, PIPs provide their customers interfaces to the offered vir-

tual resources so that customers are able to configure and deploy their leased resources. In general, PIPs are any entities that operate physical resources and can be categorized according to their main business model as customer, access, transport, data center, or other service specific PIPs.

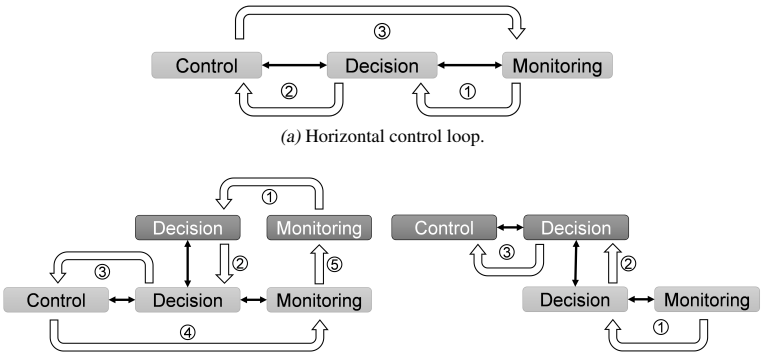
On the second layer in the hierarchy, *Virtual Network Providers (*VNPs) act as brokering entity between PIPs on the lowest layer and *Virtual Network Operators (*VNOs) on the third layer. VNPs are not restricted to the resources of a single PIP but are able to combine the virtual resources of different PIPs to form virtual networks that span several PIPs. Combining the resources may be required to create customer networks with global connectivity or to connect specialized PIPs such as transport, access, and data center PIPs to form certain services like a video streaming platform with dedicated customer access. By partitioning the physical resources into virtual resources, the VNP optimizes the utilization by allocating free capacities among its customers.

On the third layer, VNOs configure and manage the virtual networks and deploy monitoring and control components to ensure that contracted SLAs with the VNP are fulfilled. The VNPs provide the necessary interfaces to the virtual resources so that required protocols and mechanisms can be installed. Further, VNOs interact with customers and receive the network requirements of their applications. These requirements are then translated to virtual resource descriptions that are sent to the VNP.

On the highest fourth layer, *Application Service Providers (*ASPs) get access to the virtual network from the VNO and deploy the application in the network. The ASP may itself be the customer or just deploys and manages the application for a customer.

Creation of Virtual Networks

The creation of virtual networks in such an architecture involves several steps and interactions between the different roles [27]. Prior to the creation of the virtual network, VNP and PIP have already established a trust relationship and have



(a) Horizontal control loop.

(b) Vertical control loop triggered by upper layer.

(c) Vertical control loop triggered by lower layer.

Figure 1.2: Control loops in the COMCON architecture [27].

exchanged information about provided virtual resources. As first step during the creation, the ASP sends a request to the VNO which contains an informal description of the virtual network. The VNO refines the informal network description so that it can be translated into a virtual resource description and hands it over to the VNP. The VNP analyzes the virtual resource description and plans a virtual network accordingly. During this process, the VNP iterates over all required PIPs and reserves and requests the resources for the virtual network. As last step, the VNP interconnects the resources from different PIPs and hands over the unconfigured virtual network to the VNO. The VNO then configures the individual resources and hands over the configured network to the ASP. Eventually, the ASP receives the virtual network and is able to deploy the desired application.

Information Exchange and Control Loops

During the operation of virtual networks, the different involved business roles need to monitor and control the offered resources to discover and solve problems as soon as possible. To achieve this, all roles deploy control, decision, and monitoring components. Depending on the concerned roles, three different con-

trol loops arise, which are shown in Figure 1.2. The horizontal control loop that relates to only one role is shown in Figure 1.2a. The monitoring components gather data about the current state of the observed resources and store this data in a central monitoring database. The control component has access to this data and may initiate certain actions via the control component. These actions then influence the current state of the resources which is recognized by the monitoring component and the control loop is closed.

Due to the strict separation of resources on different layers, a decision component may not have the required authority to trigger actions on other layers. These cases require an interaction between different layers which may be initiated by the upper or the lower layer, as shown in Figures 1.2b and 1.2c respectively. In the first case, the monitoring component on the upper layer detects a problem and informs the decision component. The problem however is not related to the own layer but to a lower layer and hence, the decision component of the lower layer is contacted. This decision component has the required authority to initiate a countermeasure which is again recognized by the monitoring on the same and the upper layer. Again, this interaction results in a closed loop. In the last case, a monitoring component on a lower layer detects a problem which is related to an upper layer. The decision component of the upper layer is informed by the decision component of the lower layer and triggers the required actions via its control component. Here however, these actions may not be recognized by the monitoring component on the lower layer and the control loop is not necessarily closed.

In the COMCON architecture, PIPs monitor the state of the physical resources to detect for example hardware failures or increased load. VNPs in contrast have no control over own resources but only act as intermediate between PIPs and VNOs. Hence, VNPs only propagate occurring events between the layers. The VNO is responsible for the state of the virtual network and monitors for example the remaining resources in the virtual network. In case of increasing load, the VNO may react either locally by adapting its virtual topology via routing changes or requests more resources from the VNP. On the highest layer, the ASP

monitors the user-perceived service quality of its offered application to evaluate the severity of network problems with respect to the service quality and requests more resources from the VNO if needed.

1.1.2 Use Case

In addition to the business-role model, several use cases for a future virtualized architecture have been developed in the COMCON project [20, 21]. A so-called Beta slice is used to deploy new services in the network. Therefore, the service is first tested in a small isolated part of the network and in subsequent steps of deployment, the size and reach of the virtual network is extended until the service is ready for production deployments. Another use case is called *service component mobility*. The underlying idea is that services hosted on virtual nodes follow their customers so that the perceived quality of costumers is the same regardless their current location. A combined use case and the involved challenges is the motivation for this work and detailed in the following.

Overview

As mentioned earlier, video streaming is responsible for the largest fraction of global Internet traffic [25]. The number of different video streaming services and platforms is increasing remarkably. Popular examples are *Video-on-Demand (*VoD) platforms like Netflix or YouTube and IPTV services from network providers or TV broadcasters. In parallel, the number of mobile subscriptions is also rapidly increasing and there are currently 1.1 billion active mobile broadband subscriptions [28]. Hence, more and more consumers access video streaming services via mobile networks while still expecting the same video quality as in fixed or local access networks.

In the considered scenario, a video streaming service is hosted in the cloud and is streamed to mobile endpoints in wireless access networks, c.f. Figure 1.3. The mobile endpoints roam between different public or private access networks with distinct characteristics but still want the receive the video stream in a good quality

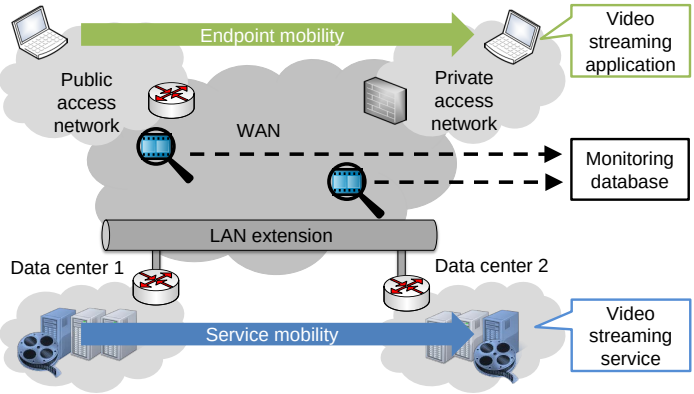


Figure 1.3: Considered video streaming scenario.

without noticeable interruptions. As the video streaming servers are deployed on virtual machines inside a data center, the services itself also may be mobile and change its hosting data center. Migrating the video streaming servers could be necessary to reduce the delay between streaming servers and mobile endpoints after a handover, e.g., between different access technologies or between public and private access networks. Another use case is to adapt the video streaming topology inside the connecting *Wide Area Network (*WAN) to reduce the load on possible bottleneck links in the network.

Challenges

A virtual network tailored for the specific needs of mobile users that access video streaming services involves several challenges on the different layers of the architecture. On the lowest layer, several different PIPs need to be interconnected in case the virtual network spans access PIPs, transport PIPs, or data center and cloud computing PIPs. In particular, the interconnection of data centers is important if services running on virtual machines should be migrated between different locations. Migrating servers within an Ethernet broadcast domain does not

require changes on network layer and it is sufficient if both data centers are transparently connected on Layer 2 [29–31]. This however leads to a large number of hosts connected to the same broadcast domain and hence, to a high amount of broadcast traffic due to flooding protocols like the *Address Resolution Protocol (*ARP). This scalability issue due to address resolution traffic in large data center networks is discussed within the *Internet Engineering Task Force (*IETF) [32]. There are many different solutions to tunnel Ethernet frames over a WAN [33] and also ongoing standardization activities on further possibilities to exchange MAC address reachability information [34]. Hence, one challenge in this area is the evaluation of available mechanisms with respect to the address resolution scalability. Another challenge is how planned migration processes can be scheduled and what is a suitable solution which considers the specific requirements of virtual machine migrations.

Migrating VMs within a virtual network that spans several data center locations involves another challenge related to the ingress path, i.e., the location at which packets enter the spanning virtual network. After the migration process is finished, packets are still first forwarded to the old location since IP routing entries for the IP of the migrated VM still point to that location. The underlying problem is that the current IP address both serves as identifier on transport layer and as routing locator on network layer [35]. Hence, to avoid the detour via the old location for the traffic to the migrated VM, a new flexible routing architecture is required that decouples identification and location information of today's IP naming and addressing architecture. Several new routing architectures [36] have been developed and standardized in the *Internet Research Task Force (*IRTF) *Routing Research Group (*RRG) [37] from which the *Locator/ID Separation Protocol (*LISP) [38] has received the most attention. LISP decouples the combined identification and location functions of today's IP address and offers the integrated mobility extension *LISP Mobile Node (*LISP-MN) [39] as well as a specific solution for virtual machine migrations [40]. However, there are still several challenges with respect to LISP-MN in such a scenario that involve both the control and the data plane.

On the highest layer, a suitable video quality monitoring solution in the network is required so that possible video quality degradations can be detected and countermeasures can be initiated before the user perceived quality is influenced. The challenge is that the perceived quality of consumers has several influence factors like the physical and social context, the expectation and usage history of the human user, and the technical system itself [41]. Despite the technical network parameters, these factors are difficult to measure. Another important factor is the video quality itself and there are several monitoring solutions which try to infer the video quality from technically measurable parameters. The underlying idea of these approaches is to include available information like spatio-temporal error propagation or errors produced by spatial and temporal concealment to provide an approximation of full reference metrics or subjective user surveys. Such approximations and mappings to full reference metrics however introduce errors, since context and content information may not be reflected sufficiently by such a solution. Hence, developing a suitable video quality monitoring solution featuring the necessary accuracy is the challenge in that area.

1.2 Scientific Contribution

This monograph details challenges that arise when deploying a certain service, i.e., video streaming in a future virtualized network architecture and discusses possible solutions. First, mechanisms enabling service component mobility within virtual networks spanning several PIPs are addressed. Second, endpoint mobility related protocols are discussed. Finally, at the highest layer in the hierarchy, a monitoring solution is presented that monitors the video quality in virtual networks.

Figure 1.4 gives an overview of the contribution of this work. The individual research studies carried out during the course of this work are classified according to their used methodology. In particular they are classified with respect to practice oriented methods like proof-of-concept implementations, demonstrations, and simulations or theory-oriented methods like analysis, proposed archi-

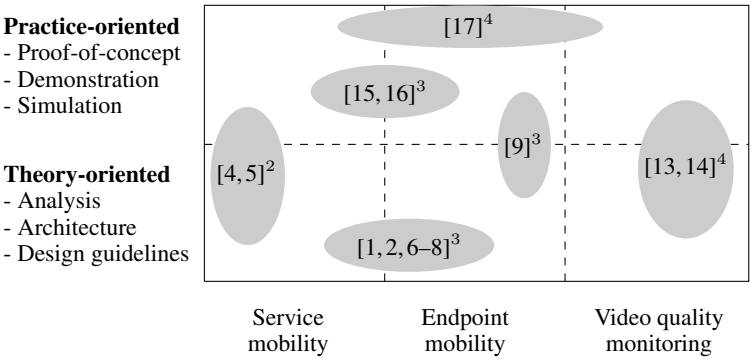


Figure 1.4: Contribution of this work illustrated as a classification of the research studies conducted by the author. The notation $[x]^y$ indicates that the scientific publication $[x]$ is discussed in Chapter y of this monograph.

tectures, and design guidelines. The respective focus of the studies cover transparent connection of data centers, future Internet routing architectures, and video quality monitoring solutions in the network. The markers $[x]^y$ indicate that the scientific publication or demonstration $[x]$ provides the basis for Chapter y.

The first part covers the transparent connection of PIPs and in particular the migration of virtual machines between different locations of a data center. First, we investigate the specific requirements during the migration of virtual machines and present an optimized architecture for live server migrations. This architecture comprises advance reservations and a subscription model to reduce the time for such a migration. Transmission requests are served consecutively and have access to the physical transmission capacity if needed. A scheduler organizes the transmission of flows in the network. To avoid excessive waiting times for flows until they may start transmission, we introduce a subscription model that gives customers priority access to the scheduler so that a bounded amount of their transmission requests are preferentially served.

Another contribution is the evaluation of *Data Center Interconnect (*DCI) solutions with respect to their address resolution scalability. Two prominent protocol examples are *Virtual Private LAN Services (*VPLS) [42] and *Overlay Transport Virtualization (*OTV) [43,44]. VPLS uses *Multiprotocol Label Switching (*MPLS) tunneling and data plane learning along with flooding of unknown frames to connect Ethernet networks. OTV in contrast uses IP tunneling and a separate overlay control plane to connect Ethernet networks over an IP core network. Both protocols hence apply different principles to transparently connect data center locations. In consideration of the mentioned scalability problems within a data center, it is important to understand possible side effects or performance issues due to address resolution on these technologies. To assess their performance in this problem domain, we utilize the mentioned scalability issues as benchmark and investigate the behavior of both technologies. In detail, we first model the address resolution traffic and then study the signaling load caused by this traffic for VPLS and OTV. In addition, we show how an ARP proxy can improve the overall scalability for both interconnect solutions.

In the second part, we discuss issues that arise in a scenario where LISP and its mobility extension LISP-MN are used to enable service component and endpoint mobility for virtual networks spanning different PIPs. First, the mobility related forwarding of LISP-MN is not always optimal and we propose a set of improvements to optimize the forwarding behavior. The optimizations aim at several issues on control and data plane and either reduce the signaling delay or the encapsulation overhead. Second, another severe challenge is that mobile nodes behind NAT boxes are not supported by LISP-MN. Hence, we present a NAT traversal mechanism to restore connectivity behind NAT boxes. Finally, the different handover mechanisms proposed by LISP-MN have not been studied so far for a video streaming scenario. To solve this, we implemented LISP in a simulation framework to evaluate both the efficiency of our proposed improvements as well as the handover performance for the video streaming use case.

The third part of this monograph presents a video quality monitoring solution that uses a full reference metric to precompute the distortion per *Group of Pic-*

*tures (*GOP) for different frame loss scenarios. This precomputed information is used to improve the accuracy of the monitoring in the network, which infers the video quality from lost frames. However, including all possible frame loss combinations per GOP introduces a large number of frame loss scenarios and hence, excessive computing power is required. To achieve a better scalability of our approach, higher frame loss scenarios are approximated by adding the distortion of single frame loss scenarios. Hence, only the distortions for single frame loss scenarios need to be precomputed. This approach however reduces the accuracy compared to a full reference metric. Hence, we evaluate the accuracy of our solution by comparing it with a full reference metric for different frame loss scenarios. As an example, we apply the *Structural SIMilarity (*SSIM) metric [45] as *Video Quality Assessment (*VQA) metric to precompute the frame distortions. Further, we consider different high definition test video sequences and GOP structures and investigate the influence on the accuracy of our proposed approximation. In addition, to prove the rationality of our solution, we compare it for different loss scenarios on *Real-time Transport Protocol (*RTP) layer with the current state-of-the-art in network-based video quality monitoring. This comparison involves the correlation with the full reference SSIM metric as well as the percentage of wrongly classified GOPs.

1.3 Outline of This Thesis

The organization of this monograph is shown in Figure 1.5. Each chapter contains a section that shows the background and related work of the covered aspects and summarizes the lessons learned. The three columns cover from left to right (1) the problems and challenges, (2) the methodologies and mechanisms to cope with them, and (3) the impact of the applied mechanisms on the performance and the results derived. The arrows between the sections show their relation, background, and findings that are used in later sections. The section numbers of the building blocks are given in parentheses.

The remainder of this thesis is organized as follows. Chapter 2 covers the connection of PIPs on the lowest level in the hierarchy and we discuss the requirements due to migration of virtual machines between different physical locations. First, we present an optimized architecture for the planned transmission of VMs between locations and compare our architecture with the current state-of-the-art in the Internet, i.e., concurrent transmissions based on TCP. Second, we discuss different protocols to transparently connect the Ethernet networks of data centers and evaluate their address resolution scalability by applying a model for address resolution traffic. We further show how an address resolution proxy reduces the amount of address resolution traffic and increases the scalability.

Chapter 3 focuses on the redirection of flows on network layer which is required to optimize the ingress path after a VM migration within a virtual network spanning several data center locations. We focus on LISP as one approach to achieve the redirection and present improvements to LISP and its mobility extension LISP-MN. We further detail how the connection to mobile nodes breaks if they roam behind a NAT gateway and explain, how our NAT traversal mechanism solves this issue. Finally, we show our simulation framework for LISP and study the handover capabilities of LISP-MN, which are among others important during endpoint mobility.

In Chapter 4, we present a monitoring solution that can be implemented in virtual networks to assess the perceived video quality of video consumers by monitoring the lost frames in the network. First, we explain the basic architecture and the made assumptions and approximations. Then, we prove the accuracy of our mechanism by comparing it with a full reference approach. In the last part, we demonstrate the viability of our monitoring solution by comparing it with other monitoring approaches.

Finally, this monograph is concluded in Chapter 5 by a summary of the presented results and achievements.

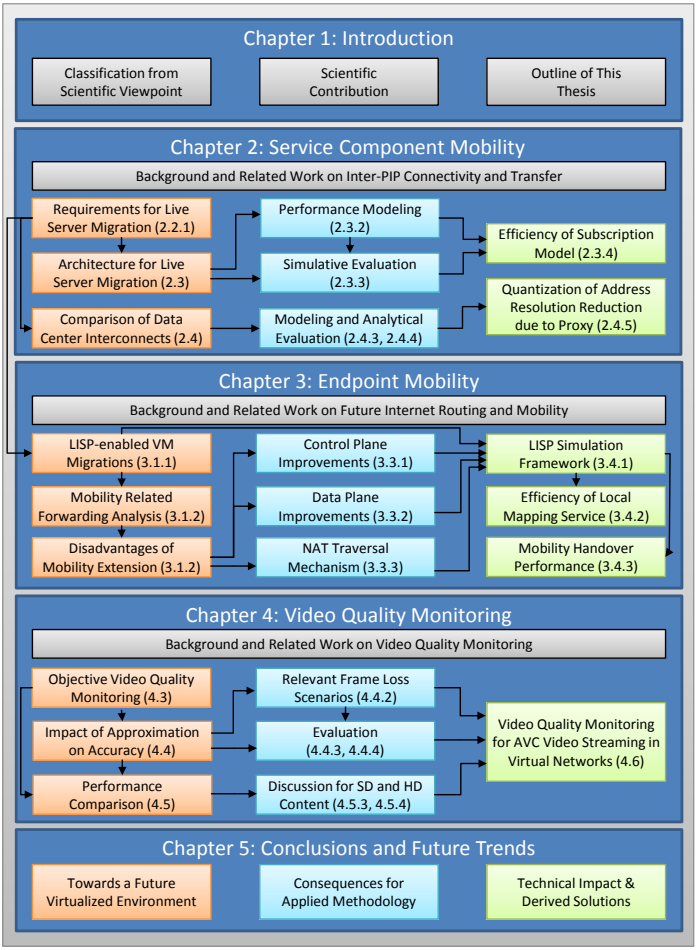


Figure 1.5: Organization and contribution of this monograph.

2 Service Component Mobility

In the previous chapter, we introduced the COMCON virtual network architecture and the considered video streaming use case for this work. The video streaming use case introduces several challenges on the different layers of the architecture.

In this chapter, we investigate mechanisms to connect physical infrastructures. Connecting different PIPs, especially data center PIPs, is important to enable mechanisms like service component mobility for the video streaming servers. Service component mobility in this field of application means to migrate video streaming servers between different locations of a data center provider, e.g., to adapt the video streaming topology between video source and consumer. The process comprises the migration of the VM on which the video streaming server is hosted and redirecting the current flows to the new location, ideally without perceivable interruptions. The focus of this chapter lies on mechanisms that enable the migration of VMs and we study technical principles and intrinsic requirements regarding the data transfer and the underlying network.

An important characteristic of VM migration is that the total data transfer volume increases with increasing transfer time. The reason is that already copied memory pages need to be resent when they are overwritten by the VM during the migration [46]. Hence, the migration process benefits from a high bandwidth since in that case, the amount of resent memory pages during the transmission process is kept minimal. The highest bandwidth can be achieved by allocating all available resources exclusively to one transmission process and serving jobs sequentially rather than in parallel. Considering this circumstance, we propose a network service that reduces the time for such a migration in the first part of this chapter. Transmission requests are served consecutively and have access to

the physical transmission capacity if needed. A scheduler organizes the transmission of flows in a network. To avoid excessive waiting times for flows until they may start transmission, we introduce a subscription model that gives customers priority access to the scheduler so that a bounded amount of their transmission requests are preferentially served. The proposed network service with subscription model is denoted as data beaming in the remainder as transmission requests are served at very high bandwidth during a short time slot.

Regarding the underlying network, the migration of VMs requires a transparent interconnection of different data center sites, i.e., the transport of Ethernet frames over a WAN. There are many different solutions to tunnel Ethernet frames over a WAN [33]. However, if a large number of nodes is attached to a flat data center network, the broadcast traffic caused by the ARP can result in scalability issues [30]. In consideration of these scalability problems within a data center, it is important to understand possible side effects or performance issues due to address resolution on these *Data Center Interconnect (*DCI) technologies. To assess their performance in this problem domain, we utilize the mentioned scalability issues as benchmark and investigate the behavior of both technologies. In detail, we first model the address resolution traffic and then study the signaling load caused by this traffic for selected interconnect technologies. In addition, we show how an ARP proxy can improve the overall scalability for both interconnect solutions.

The content of this chapter is mainly taken from [4, 5]. Its remainder is structured as follows. First, we explain technical principles for VM migrations and discuss specific requirements with respect to the data transfer and the underlying network. Second, we present work related to our proposed data beaming solution in the area of delay tolerant data transfers and give a brief overview to data center interconnect solutions. Third, we detail our data beaming solution and show the advantages compared to the state-of-the-art. Finally, we present an evaluation of data center interconnect solutions with respect to the address resolution scalability and conclude with the lessons learned.

2.1 Background

In the following, we explain the technical principles and requirements of the VM migration process and discuss the addressed challenges in the area of VM migrations between different data center locations.

2.1.1 Requirements for VM Migration

Live server migration facilitates the management and administration of virtual nodes and reduces downtime resulting from maintenance operations. It enables load balancing and live workload mobility for a service provider without perceivable interruption of its provided services by currently connected clients [22]. In addition, it improves energy efficiency of a service provider hosting several servers. To avoid idle or less loaded servers which nevertheless require about 66% of their total energy consumption [47], less loaded server processes may be migrated to only a few physical nodes so that the remaining nodes can be switched off to save energy [48].

Technical Principles

Migrating a VM within a LAN usually requires only the transfer of the run-time memory state, the VMs image. The local persistent state, the VMs file system is stored in a *Network Attached Storage (*NAS) and is hence accessible from the source and destination server and does not need to be transferred. This changes however if a VM is migrated between data center locations connected via a WAN. In that case, source and destination server usually do not have access to the same NAS so that also the local persistent state must be transferred [49]. This increases the amount of transferred data and hence influences the *downtime* and *total migration time*. The downtime comprises the time period where no instance of the VM is running and the offered service is not available to possible customers. The total migration time starts with the initiation of the migration process and is finished, once the destination server takes over the service prior running on the

source server. Both time intervals should be kept minimal but the downtime is more critical as during this period, the offered service is not available. There are many solutions to migrate VMs [50] and depending on the induced downtime and total migration time, the different approaches can be categorized according to three generalized phases. Although it is possible to combine all three phases in one approach, most reasonable solutions implement only one or two phases.

Push phase: the server on the source node continues its operation while certain parts of the working memory are pushed towards the destination node. Modified parts of the working memory at the source node need to be resent over the network to keep the working memory of the destination node consistent. Thus, the total migration time depends on the transmission rate and the rate at which the working memory changes.

Stop-and-copy phase: the server on the source node is stopped, the memory is copied from the source node to the destination node, and the server on the destination node is started once all working memory has been transferred. This approach is rather simple and requires no complex mechanisms but induces the longest downtime which depends on the total amount of transferred data and the available bandwidth.

Pull phase: after a short copy phase, the server on the source node is stopped and the server on the destination node is started. The new node pulls missing parts of the working memory from the source node. This implies a very short downtime, but the performance of the new node is usually very weak until a sufficiently large part of the working memory has been transferred.

An efficient approach for live migration is the pre-copy migration scheme [50]. During an initial push phase, the memory is sent in several iterations and modified pages of already sent parts of the memory are resent in subsequent iterations. Once a sufficiently large part of the memory has been transferred, the server on the source node is stopped and the remaining parts of the memory are sent to the destination node. Eventually, the server on the destination node is started.

Thus, a certain downtime during the migration process cannot be avoided, but it should be kept small. The length of the interruption mainly depends on the available bandwidth, the workload of the migrated server process [51], and the used technique to migrate a virtual node [50]. In [52], the downtime and total migration time have been analyzed for different workloads and VM sizes. The migration of a VM with 4096 MB executing a MapReduce task took for example about 15 s and the induced downtime was about 350 ms. The migration process in that case had access to the full bandwidth of 10 Gbps.

Requirements for Data Transmission

During the push phase of a migration process, the server on the source node keeps working and may modify already copied parts of the memory so that they need to be resent to the destination node. The rate at which already copied memory is overwritten by the server on the source node is called *dirtying rate*. The dirtying rate depends on the workload of the service offered by the virtual server.

Live server migration between different sites benefits from a high bandwidth for two reasons. First, more bandwidth reduces the number of copy iterations during the push phase and thus the transferred data volume. Second, with higher bandwidth, the remaining amount of data is also faster transferred in the stop-and-copy phase. In summary, the larger the bandwidth between the source and destination site, the shorter is the downtime of a migrated service. Therefore, high bandwidth is a key requirement for live migration. The live migration of a single common web server for example requires a bandwidth of 10 Gbps to achieve a migration time of about 8 s and a downtime of about 450 ms [52].

Network Requirements

After the migration process, the IP address of the migrated VM must not change because otherwise, existing transport connections with endpoints would break. Hence, a requirement to the underlying network is that source and destination servers are part of the same subnet and same broadcast domain. In a typical Eth-

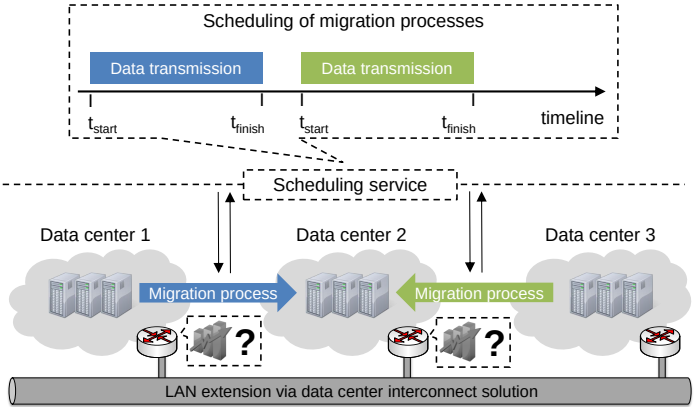


Figure 2.1: VM migration challenges.

ernet network, switches learn the mapping from MAC address to ingress port and after a migration has occurred, the mappings in all affected switches and gateway routers need to be changed. Therefore, the *Virtual Machine Monitor (*VMM) or hypervisor on the destination server sends an ARP announcement using gratuitous ARP requests [53] to inform all affected switches and servers about the new location of the migrated machine. This is done once the hypervisor detects that the migration process is finished. All servers and routers which receive this gratuitous ARP request update their ARP caches and all switches which receive this gratuitous ARP request update their MAC-to-IP table so that future packets to the migrated virtual machine are correctly forwarded to the new destination server. Hence, if VMs are migrated over a WAN, both source and destination server must be part of the same Ethernet broadcast domain and Layer 2 connectivity is required between source and destination data center location.

2.1.2 Migration Challenges

As explained above, the data transfer during the VM migration process requires a large bandwidth so that the total migration time and the downtime are sufficiently

low. As live server migration is a planned process of the data center management, a time scheduling is possible, i.e., the migration could be started when sufficiently high bandwidth is available. Postponing the process is typically also possible, at least for a short time duration. Hence, one challenge that is addressed in the remainder is how the scheduling of VM migrations can be improved and what is the benefit with respect to the current state-of-the-art in the Internet, i.e. concurrent transmission based on TCP. Ongoing research in that problem domain is related to delay tolerant data transfers or advance reservations and in the next section, we summarize the most important research.

Concerning the network requirements of VM migrations over a WAN, source and destination data center must be transparently connected on Layer 2. There are various DCI solutions available which establish Layer 2 connectivity between data centers [33] and hence enable VM migrations over a WAN. However, if a large number of nodes is attached to the same Ethernet network, the broadcast traffic caused by address resolution can result in scalability issues [30]. Hence, a second challenge is to investigate the effect of a large amount of broadcast traffic on the DCI solutions which enable transparent Layer 2 connectivity between data center networks. A combined scenario showing both challenges is shown in Figure 2.1. Three data centers are transparently connected on Layer 2 via a DCI solution and a scheduling service ensures that each data transmission gets access to the link bandwidth so that downtime and migration time are sufficiently low. We investigate how a more sophisticated scheduling of VM migrations reduces the total migration time and downtime and what is the scalability of different DCI solutions with respect to broadcast traffic.

2.2 Related Work

In this section, we first present work related to the scheduling of delay tolerant data transfers. This comprises research in the area of advance reservations as well as architectures that enable delay tolerant data transfers. Subsequently, we give an overview to DCI solutions in the area of transparent Layer 2 connectivity of

data center networks and classify the different solutions according to selected building blocks.

2.2.1 Delay Tolerant Data Transfer

Delay tolerant network architectures address computer networks where full connectivity is not always available. Examples are mobile or vehicular ad-hoc networks as well as terrestrial or space networks. An application in those networks must be delay tolerant since upcoming data transfers must be delayed until network connectivity is again available. In [54], a survey of delay tolerant applications is given. Although data center networks are not directly related to delay tolerant networks, the migration of VMs in those networks is a planned process and hence to some degree delay tolerant. Hence, relevant other research to our proposed data beaming architecture is related to the topic of delay tolerant data transfers and the similar advance reservations concept. In the following, we first describe the concept of advance reservations and then introduce other architectures that also support reservation of network resources in advance and explain, how data beaming differs from them.

Advance Reservation

Requests for an advance reservation of network resources usually contain a source and destination node, a certain bandwidth requirement, and a specific time window during which the reservation becomes active. In contrast, requests for immediate reservation do not contain a time window and become immediately active once they have been admitted. With respect to the exact specification of the time window, the authors of [55] present a taxonomy to characterize advance reservation requests into three different categories.

Specific start and duration (STSD): customers indicated a specific starting time and a specific duration and do not tolerate any time displacement.

Specific start and unspecified duration (STUD): customers only specify the starting time and expect to get the desired network resource as long as possible.

Unspecified start and specific duration (UTSD): customers specify the duration of an advance reservation but do not specify a starting time. The customers in this case expect to get the desired network resource as soon as possible.

Data beaming can be categorized as a special case of UTSD where a customer indicates a maximum ending time for the advance reservation. The customers tolerate a certain delay until the communication starts as long as the data transmission is finished before the specified maximum ending time.

The flexibility of data beaming with respect to the time window avoids a big issue of advance reservations. Strict advance reservation reduces the resource utilization and acceptance rate due to a fragmentation of available network resources [56]. Data beaming loosens the strict time window constraints of advance reservation requests and hence improves the resource utilization and acceptance rate. The correlation between the laxity of advance reservations and an improved resource utilization and acceptance rate has been shown for example in [57] and [58].

Architectures

The GridFTP protocol [59] is part of the Globus Toolkit (GT) [60] and used for data access and movement. It is an extension of the FTP protocol and offers a reliable and high performance data transfer. In comparison to data beaming, GT is rather a protocol suite used to build grid networks than a new network service concept like data beaming. Both concepts enable the transfer of large date volumes, whereas GT is intended to enable applications with federated resources and data beaming primarily offers a new network service for customers which require fast and reliable data transfer at a very high bandwidth.

The framework proposed in [61] extends the GMPLS suite and its routing and signaling protocols to support advance reservation of network resources. The objective is to enable automated provisioning of advance reservations in a GMPLS-based network. The framework covers only the network reservation aspect of the data beaming architecture and does not propose a full architecture with network management. However, the framework could be used as underlying signaling protocol to realize the data beaming architecture on top of a GMPLS-based transport network.

The concept of a bandwidth broker as network management entity for a GMPLS-based network is used in the DRAGON architecture [62]. The objective is to provide large data rates for file transfer in optical networks. However, in contrast to data beaming, resources are reserved for immediate use and not in advance. The architecture specified in [63] also uses a bandwidth broker to support flexible advance reservations based on a MPLS network. The reservation requests allow certain bandwidth constraints that specify a minimum and a maximum bandwidth and a total deadline for the data transmission. This concept is very similar to data beaming.

However, the main difference between these architectures and our data beaming approach is that they do not consider a subscription model for resource reservation requests. The subscription model in data beaming facilitates the resource provisioning from the provider's perspective. Requests for a reservation of a data beaming slot are only considered if they conform to the current subscription of the customer. Otherwise, the data beaming scheduler does not guarantee that a data beam request can be realized. This is a significant improvement and both interesting for providers and customers.

2.2.2 Data-Center Interconnect Solutions

VM migrations between different locations of a data center require a transparent connection across a WAN. There are a multitude of different approaches available and we give an overview of selected solutions in Table 2.1. The main goal of

Building block	EoMPLS (EoMPLSoGRE)	VPLS VPLSoGRE	OTV	E-VPN	VXLAN	NVGRE
Basic idea	Transparent connection of Ethernet domains across a wide area network				Extension of the VLAN address space	
Backers	Metro Ethernet Forum (MEF)	Alcatel-Lucent, Juniper Networks	Cisco Systems	Cisco Systems, Alcatel-Lucent, AT&T, Verizon, Juniper Networks	Arista, Broadcom Cisco Systems, VMware, Citrix, Red Hat	Microsoft, Arista, Intel, Dell, HP, Broadcom, Emulex
Status of standard	RFC ([64, 65])	RFC ([42, 66])	Draft ([43])	Draft ([34])	Draft ([67])	Draft ([68])
Transport	MPLS (MPLSoGRE)	MPLS (MPLSoGRE)	IP/UDP	MPLS	IP/UDP	GRE
Address learning	Data plane	Data plane	Data plane and control plane	Data plane and control plane	Data plane	Data plane and control plane
Unknown frames	Flooding	Flooding	Dropped	Flooding as failover	Flooding	Flooding as failover
Control Plane	–	–	IS-IS	MP-BGP	–	Not yet specified
Broadcast handling	Flooding	Flooding	Multicast group	Flooding	Multicast group	Multicast group

Table 2.1: Comparison of data center interconnect solutions.

the first four solutions, *Ethernet over MultiProtocol Label Switching (*EoMPLS) [64, 65], *Virtual Private LAN Services (*VPLS) [42, 66], *Overlay Transport Virtualization (*OTV) [43], and *Ethernet Virtual Private Network (*E-VPN) [34] is to transparently connect Layer 2 networks over a WAN while the last two solutions, *Virtual eXtensible Local Area Network (*VXLAN) [67] and *Network Virtualization using Generic Route Encapsulation (*NVGRE) [68] aim to extend the *Virtual Local Area Network (*VLAN) address space inside a data center. The different solutions can be distinguished according to the used transport technology and the applied address learning principle. EoMPLS and VPLS use MPLS as transport technology and apply data plane learning along with flooding of unknown Ethernet frames. E-VPN also uses MPLS as transport but applies data plane and control plane address learning. Address reachability information is exchanged using *Multiprotocol Border Gateway Protocol (*MBGP). OTV in contrast uses UDP in IP encapsulation but also applies data plane and control plane address learning. In this case however, the *Intermediate System to Intermediate System (*IS-IS) protocol is used as control plane protocol to exchange reachability information.

As stated earlier in this chapter, we want to investigate how the different solutions handle broadcast traffic caused due to address resolution and evaluate the respective performance. The two important building blocks that mainly influence the broadcast traffic are *address learning* and the *handling of unknown frames*. One subset of the approaches uses data plane learning along with flooding of unknown Ethernet frames and the other subset uses data plane and control plane learning along with a routing protocol for exchanging address reachability information via the control plane. As exemplary solutions for both subsets, we take a closer look at VPLS and OTV and evaluate the address resolution scalability in detail in the remainder of this chapter. We consider different load situations and aim at studying which solution can be used under which condition.

2.3 Optimized Architecture for VM Migrations

In the previous section, we explained the technical principles and the special requirements for the data transfer during the VM migration process. In this section, we present our optimized architecture for live server migrations called *data beaming*. In the remainder of this section, we first give an overview of data beaming, explain its architecture, and propose a simple scheduler that we use in our performance study. Second, we describe the performance modeling and compare our architecture with the current state-of-the-art in the Internet, i.e., concurrent transmissions based on TCP. As last part, we show the efficiency of our proposed subscription model.

2.3.1 Data Beaming Solution

A *data beam* is a transmission request that should be served at very high bandwidth so that data are transmitted within very short time. Customers wishing to transmit a certain data volume at a certain rate indicate that through the data beaming interface to the management of a transport network which then returns a connection over which the customer can transmit the data. It may be implemented in packet-switched or circuit-switched networks where *Generalized MultiProtocol Label Switching (*GMPLS) is an example as it is able to set up paths on a short timescale [69]. The authors of [61] for example present extensions to the GMPLS suite to support reservation of network resources in advance. Data beaming uses sequential scheduling implemented by a priority queue and can be modeled according to Kendall's notation as G/G/1-PRIO[1] queueing system. Hence, data beaming can be viewed as a circuit-on-demand and is in contrast to the concurrent transmission of multiple flows that compete for available bandwidth which is the philosophy of today's Internet. With concurrent transmissions, requests are served in parallel and the available resource is shared among all active transmissions. The bandwidth sharing can be modeled with the generalized processor

[1] Queueing node with general distributed arrival process, general distributed service process, one server, and priority service queueing discipline.

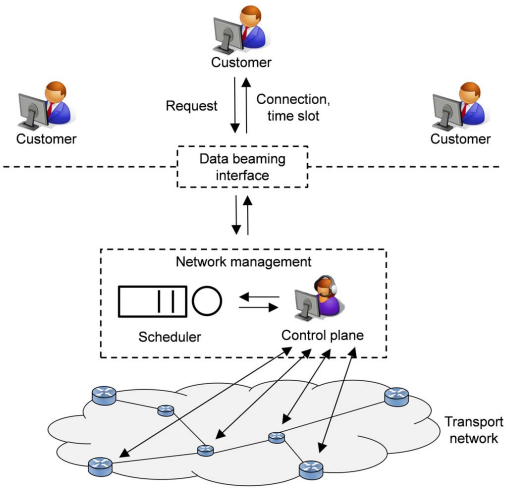


Figure 2.2: Overview of the data beaming architecture.

sharing (GPS) [70] queueing discipline and the resulting Kendall notation for concurrent transmissions is G/G/1-GPS[2].

Architecture

The data beaming architecture, which is shown in Figure 2.2, offers interested customers an interface to issue requests for a data beaming slot. In a data beam request, customers indicate the amount of data volume to be transferred, the source and destination site, certain bandwidth requirements, and an optional time window in which the data beam should be executed. Incoming data beam requests are passed to the network management, which consists of a scheduler and the control plane. Both entities work together to determine a transmission path through the network and a free time slot for each data beam. The returned path infor-

[2] Queueing node with general distributed arrival process, general distributed service process, one server, and generalized processor sharing queueing discipline.

mation may be the label of an existing MPLS label switched path or an existing GMPLS-managed lightpath through a GMPLS/MPLS transport network. The returned data beaming slot and the returned path guarantee that each data beam can be transmitted at the desired rate.

Subscription Model

Data beaming guarantees short transmission times at the expense of initial waiting times for a transmission slot. However, the processing times are reduced leading to similar overall sojourn times. The initial waiting times can be significant in case many customers want to transmit multiple data beams. To guarantee an upper bound to the waiting time, the offered load needs to be limited. To that end, we propose a subscription model. A customer may subscribe to the transport network provider and declare how many data beam requests will be issued over time, their bandwidth demands, their directions, and the tolerable waiting time for the transmission slot. Requests for data beams conforming with these subscriptions are served with high priority, i.e., they face only short waiting times until their data beams can be transmitted. Other customers without subscriptions may also indicate on-demand data beams. On-demand data beams and out-of-profile data beams are served only if no other data beams from in-profile requests are waiting for transmission.

Subscriptions are useful for customers and for providers and may be part of service level agreements. For customers they guarantee a maximum waiting time as long as the requests are in-profile. For providers they serve for planning so that their networks can be provisioned with sufficient resources. To define an upper limit of customers, providers have to consider the distinct paths between possible source and destination sites. The available bandwidth on these paths in combination with the maximum waiting time per customer is then necessary to determine the number of supportable customers. Subscriptions may be charged so that capacity investments are partly refunded by the declaration of data beams with guaranteed waiting times and partly by the actual use of the data beaming

service. More detailed subscriptions allow more cost-efficient network planning but give less flexibility to the customers. However, those including a fixed source and destination site for example, may be cheaper than more flexible subscriptions, e.g., those specifying only a source site. This is a motivation for customers to make a more detailed subscription.

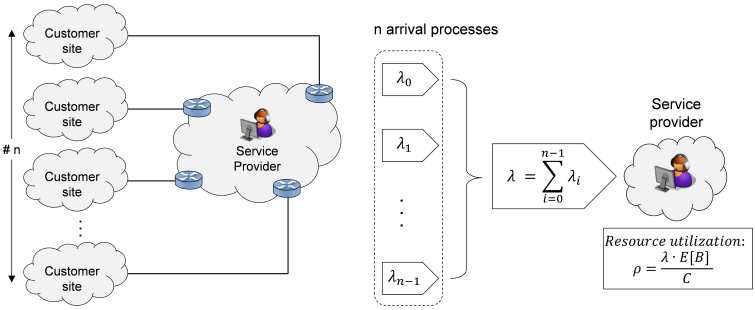
When requests for data beams conform with existing subscriptions, they must be served within the desired waiting time. To avoid extensive load, the feasibility of new subscriptions or the extension of existing subscriptions must be controlled and they must be explicitly admitted.

Simple Scheduler for Data Beaming

The scheduler for data beaming needs to respect the subscriptions of the customers. For simplicity reasons, the data volume offered by each customer is only limited by an average transmission rate r_i, $0 \leq i < n$. That means, if customers send traffic not faster than declared, their data beam requests should be served within short time. In contrast, if customers send bursts of requests or requests for large data beams, they must accept longer waiting times until they can transmit data.

For this evaluation, we simplified the considered scheduler to focus on its ability to ensure the subscriptions of the customers. We chose a priority queue that stores data beam requests for a single transmission resource until the transmission starts. In addition, the following algorithm determines the priority dates of requests x. Similar to weighted fair queuing, the algorithm uses virtual finish times $f(x)$ for that purpose, but it is also inspired by VirtualClock's ability to respect reserved rates [71]. For each customer i, the finish time of its last request is recorded in f_i which is initialized with $f_i = -\infty$. Each request is associated with a data volume $b(x)$. When a new request x arrives from customer i, the scheduler calculates the virtual finish time by

$$f(x) = \max(f_i, now) + \frac{b(x)}{r_i}, \tag{2.1}$$



(a) Illustration of the setup. *(b)* Load generation model.

Figure 2.3: Application scenario.

whereby now is the current time. Then, the customer-specific virtual finish time is set to $f_i = f(x)$ and the request is inserted into the priority queue using $f(x)$ as priority date. Whenever the transmission resource is free, the request with the lowest priority date is removed from the priority queue and transmitted.

2.3.2 Performance Modeling

We first describe a general application scenario and then we develop a performance model for data beaming (DB) and concurrent transmission (CT).

Application Scenario

Figure 2.3a shows a transport network interconnecting n customers at different sites. The customer of each site wants to transmit traffic to the other sites which requires a common resource. The common resource may be the transport network of a service provider or adjacent networks which federate to offer data beaming. For this first simulation study, we concentrate on the first case and assume a single provider network as connecting resource between customer sites (see Figure 2.3a). We further treat the service provider network as a single resource as the

objective is to present first performance trade-offs rather than a detailed protocol simulation in a realistic topology.

The inter-arrival time between consecutive transmission requests of customer i, $0 \leq i < n$, is modeled by identical and independently distributed (i.i.d.) random variables A_i, and the request sizes are modeled by i.i.d. random variables B_i. With $\lambda_i = 1/E[A_i]$, the overall inter-arrival time can be calculated by $\lambda = \sum_{0 \leq i < n} \lambda_i$.

To simplify our performance evaluation, we assume that all customers have the same mean request inter-arrival time $E[A]$ and mean request size $E[B]$ so that we can calculate the resource utilization by

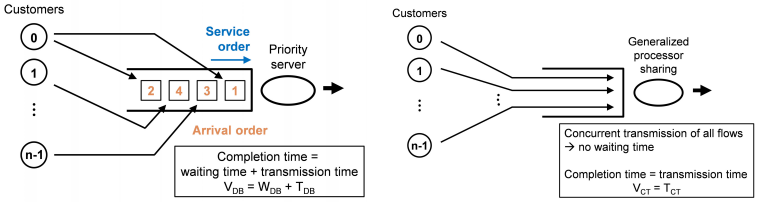
$$\rho = \frac{\lambda \cdot E[B]}{C}, \tag{2.2}$$

where C is the transmission capacity of the considered resource. In Figure 2.3b, we visualize the above described load generation model due to *n* arrival processes and the corresponding resource utilization at the service provider.

The capacity C and the average request size $E[B]$ give a lower bound on the mean transmission time which equals the mean transmission time $E[T_{DB}] = \frac{E[B]}{C}$ for data beaming. To make our study independent of assumptions of C and $E[B]$, we normalize all performance metrics in our study by $E[T_{DB}]$. The normalization allows the reader to adapt the quantities to any transmission size. In our experiments we choose a number of customers n and a resource utilization ρ, and adjust the mean inter-arrival time $E[A]$ accordingly.

Data Beaming: G/G/1-PRIO Model

We assume n customers of a data beaming service with identical subscriptions. The scheduling mechanism for data beaming is configured with $r_i = E[B]/E[A]$ which is the mean traffic rate of the flow and which is also the declared traffic rate in the subscription of the customer. When a customer sends a request to the data beaming interface, the scheduler calculates a priority date for



(a) Data beaming: G/G/1-PRIO. *(b)* Concurrent transmission: G/G/1-GPS.

Figure 2.4: Queuing models for concurrent transmission and data beaming.

the request according to Equation (2.1), and the request is inserted into the priority queue according to this priority date, which is then served in a FIFO manner. The service time for data beaming is given by $T_{DB} = B/C$ and the completion time $V_{DB} = W_{DB} + T_{DB}$ of a request is its waiting time W_{DB} plus the service time (see Figure 2.4a).

In the special case that the inter-arrival times A are exponentially distributed, the mean of the waiting time for an M/G/1-FIFO queue can be calculated with the Pollaczek-Khinchin mean value formula (Formula 1.82 in [72]) by

$$E[W] = E[T_{DB}] \cdot \frac{\rho}{1-\rho} \cdot \frac{1 + (c_{var}[B])^2}{2}. \tag{2.3}$$

This, however, is just an upper bound on the average waiting time for data beaming because the scheduler effects that short requests are served earlier than long request which decreases the mean waiting time $E[W_{DB}]$ for data beaming.

Concurrent Transmission: G/G/1-GPS Model

With concurrent transmission, flows are immediately served and fairly share the common resource. This is an optimistic approximation of bandwidth sharing using congestion control algorithms like TCP and stands for the current Internet.

Each flow is considered to adjust its sending rate so that the shared network resource is not congested. Bandwidth sharing using TCP relies on the fairness of each flow and does not need a control entity like data beaming.

We model the bandwidth sharing by the generalized processor sharing (GPS) [70] queuing discipline (see Figure 2.4b). It approximates the long-time average of TCP's bandwidth sharing for long-lived flows with equal round trip times. Requests are concurrently served and get an equal amount of the available bandwidth. The shared bandwidth extends the transmission time T_{CT} for concurrent transmissions to values that are a multiple of the short transmission time T_{DB} for data beaming. With concurrent transmission, the completion time V_{CT} of a request equals its transmission time T_{CT}.

If the inter-arrival times A are exponentially distributed, the transmission time T_{CT} for the resulting M/G/1-GPS queueing model can be analytically computed according to Formula 4.17 in [72] as

$$E[T_{CT}] = \frac{E[T_{DB}]}{1-\rho}. \tag{2.4}$$

2.3.3 Performance Comparison

In this section, we analyze the performance of data beaming and concurrent transmission for a single transmission resource. We explain the simulation setup and introduce the completion time V as performance metric. First, we perform experiments where all customers have the same model for transmission requests and then we study how an in-profile and an out-of-profile class of customers compete for the transmission resource.

Simulation Setup

We simulate data beaming and concurrent transmission using a flow-based model in OMNeT++ [73]. In both cases, we simulate transmission requests from n different customers. To model the inter-arrival time A of consecutive transmission

requests of a customer and the request size distribution B, we use a Gamma distribution as we can easily control its mean and coefficient of variation. For $c_{var} = 1.0$, the Gamma distribution becomes an exponential distribution which we use as default value in some cases. We conduct several experiments by choosing different values for $c_{var}[A]$ and $c_{var}[B]$ so that we can study the performance of data beaming under various scenarios. However, we avoid coefficients of variations of inter-arrival times smaller than $c_{var}[A] = 0.1$ as those systems tend to become quasi-periodic so that they require extremely long simulation runs to provide reliable results. This is not problematic for request sizes and we use a deterministic distribution for $c_{var}[B] = 0$, i.e., all requests have the same size.

Performance Metric

In the following, we consider the mean completion time $E[V]$ of transmission requests. In case of data beaming it is the sum $E[V_{DB}] = E[W_{DB}] + E[T_{DB}]$ of the mean waiting time and the mean transmission time. For concurrent transmission, it is just the mean transmission time $E[V_{CT}] = E[T_{CT}]$. We present the completion time as a multiple of the minimum mean transmission time which is equal to $E[T_{DB}] = E[B]/C$. Thus, the normalized mean waiting time $E[W_{DB}]$ for data beaming is the normalized mean completion time $E[V_{DB}]$ minus 1 and is hence implicitly given in each figure.

We assume that n customers send traffic over the single resource and that each of them has a subscription for a traffic rate of $\frac{E[B]}{E[A]}$. The obvious system parameters are the number of customers n and the resource utilization ρ. In addition, the coefficient of variation of the inter-arrival time $c_{var}[A]$ and the transmission request size $c_{var}[B]$ influence the completion times for data beaming and concurrent transmission. We investigate their impact in the following.

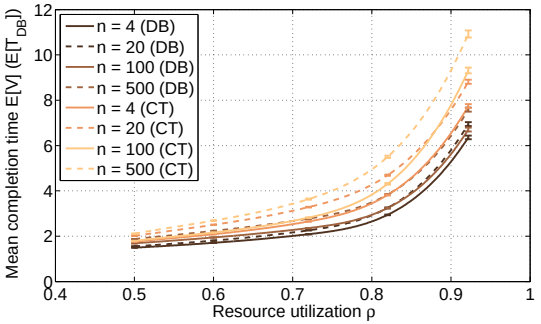


Figure 2.5: Completion time for DB and CT with $c_{var}[A] = 0.1$ and $c_{var}[B] = 1.0$.

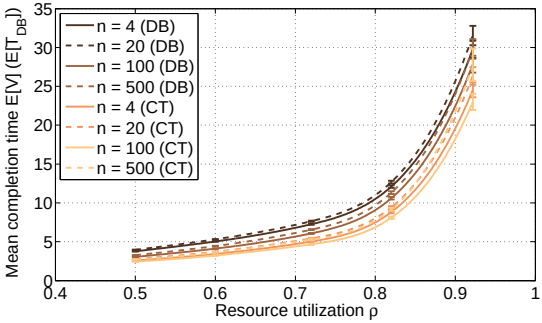


Figure 2.6: Completion time for DB and CT with $c_{var}[A] = 2.0$ and $c_{var}[B] = 1.0$.

Impact of Resource Utilization, Number of Customers, and Inter-Arrival Time Variability

In our first experiments, we set the coefficient of variation of the transmission request size to $c_{var}[B] = 1$ and set the one for the inter-arrival time to extreme values $c_{var}[A] = 0.1$ and $c_{var} = 2.0$. Figures 2.5 and 2.6 show the completion time depending on the resource utilization ρ and the number of customers n. In both cases, the completion time increases with increasing resource utilization and it is shorter for data beaming than for concurrent transmission. The transmission time for concurrent transmission $E[V_{CT}]$ is at least twice as large as the one for

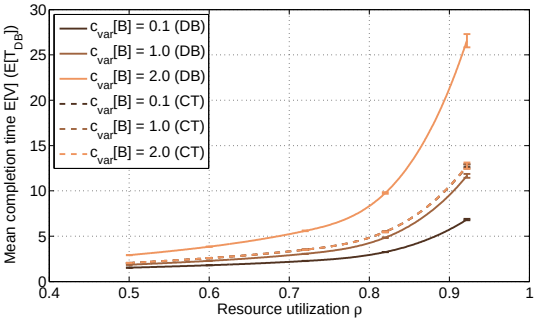


Figure 2.7: Completion time for DB and CT with $c_{var}[A] = 1.0$ and variable $c_{var}[B]$.

data beaming $E[T_{DB}]$ and can quickly become a multiple of it. The figures show that for $c_{var}[A] = 0.1$, the completion time slightly increases and for $c_{var}[A] = 2.0$ slightly decreases with the number of customers n, but the impact of the utilization ρ is much larger.

Impact of Request Size Variability

In another experiment we observed that the completion time is independent of the number of customers n for $c_{var}[A] = 1.0$. In that case, the inter-arrival times between requests of a single customer are exponentially distributed as we use a Gamma distribution. As a consequence, the inter-arrival times from all customers are also exponentially distributed. As we keep the resource utilization ρ constant, the mean of the inter-arrival time of the request arrivals multiplexed from all customers is the same for all n. As the exponential distribution has only a single parameter, the arrival process for all requests is the same for all n. Therefore, the simulation results are independent of n for $c_{var}[A] = 1.0$.

Figure 2.7 shows the completion times for data beaming and concurrent transmission depending on the resource utilization ρ for different coefficients of variations $c_{var}[B] \in \{0.1, 1.0, 2.0\}$ of the transmission request sizes. We observe that the completion time for concurrent transmission is independent of that value.

In fact, for the special case of exponentially distributed inter-arrival times, the completion time can also be analytically calculated by Equation (2.4) which is also independent of the coefficient of variation $c_{var}[B]$. We also observe that the completion time for data beaming increases with increasing $c_{var}[B]$ and it even exceeds the one for concurrent transmission for a large value of $c_{var}[B] = 2.0$.

For $c_{var}[A] = 1.0$ and $c_{var}[B] = 1.0$, concurrent transmission leads to the same mean waiting time as an M/G/1 queuing system. In Figure 2.7, the curve for the completion time with data beaming is lower than the one with concurrent transmission. This shows that the completion time for M/M/1 systems can be reduced by substituting the FIFO service order by the order proposed by our scheduler.

Overview of Impact of Transmission Request Variability

We systematically study the effect caused by the coefficients of variations $c_{var}[A]$ and $c_{var}[B]$ for $n = 100$ and $\rho = 0.7$. Figure 2.8 shows the completion time depending on the coefficient of variation of the inter-arrival time $c_{var}[A]$ for different coefficients of variation $c_{var}[B]$ of the transmission request size. Increasing variability of the inter-arrival time leads to longer completion times. For data beaming, more variable transmission request sizes also increase the completion time. But for concurrent transmission we observe that the completion time is identical as long as the coefficient of the inter-arrival time is at most $c_{var}[A] = 1.0$. If the coefficient of variation $c_{var}[A]$ is larger, the completion time decreases with an increasing coefficient of variation $c_{var}[B]$ of the transmission request size. This sounds counterintuitive, but can be explained as follows. If a transmission request is very long, many very short requests may arrive in the meantime, they are served in parallel, they complete quite quickly as they are short, and in particular earlier than the long transmission request. The average completion time may be short if many short transmission requests arrive. With increasing variability of the transmission request size, the effect of this example dominates the completion time.

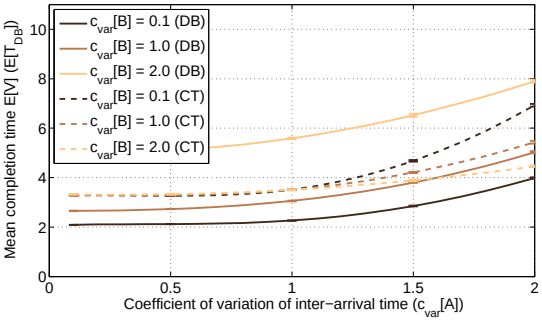


Figure 2.8: Completion time for DB and CT ($\rho = 0.7, n = 100$).

2.3.4 Efficiency of Subscription Model

In the following, we call customers in-profile if they send transmission requests smoothly over time so that their transmission rates deviate only little from their subscribed rate. Otherwise, we call them out-of-profile. We only need this qualitative description in the following to differentiate two customer types.

We have shown that data beaming leads to short waiting times when all customers are in-profile, i.e. when they have low $c_{var}[A]$ and low $c_{var}[B]$, and that data beaming leads to long waiting time when all customers are out-of-profile. Now we assume that a class of in-profile customers ($c_{var}[A] = 0.1, c_{var}[B] = 0$) and a class of out-of-profile customers compete for the transmission resources. We show that our proposed scheduler for data beaming can well enforce short waiting times for in-profile customers under various conditions. We use $n = 100$ customers in the following experiments and a resource utilization of $\rho = 70\%$.

Coexistence of In-Profile and Out-of-Profile Customers

First, we investigate a scenario with 50% in-profile customers and 50% out-of-profile customers. We consider out-of-profile customers that have constant request sizes ($c_{var}[B] = 0$) and vary the coefficient of variation of their inter-arrival times $c_{var}[A]$. Figure 2.9 shows that the completion time for in-profile

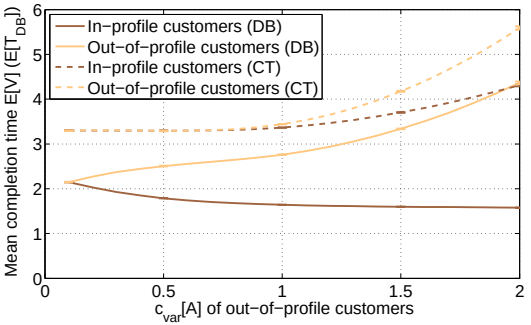


Figure 2.9: Completion time for DB and CT for 50% in-profile and 50% out-of-profile customers with variable $c_{var}[A]$ and $c_{var}[B] = 0$.

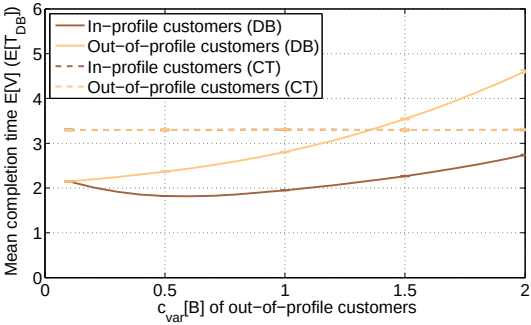


Figure 2.10: Completion time for DB and CT for 50% in-profile and 50% out-of-profile customers with $c_{var}[A] = 0.1$ and variable $c_{var}[B]$.

customers decreases with the variability of the traffic sent by out-of-profile customers and that the completion time for out-of-profile customers increases. This is due to the fact that in-profile customers have higher priority over more out-of-profile customers so that they get served faster when the transmission resource becomes free again. An important observation is that the completion time of in-profile customers is bound by two times the mean transmission time $E[T_{DB}]$ for data beaming. For concurrent transmission, the completion time is much larger and transmission requests from in-profile customers face similar waiting times as transmission requests from out-of-profile customers.

Now, we study out-of-profile customers that have low variability in the inter-arrival times of consecutive requests ($c_{var}[A] = 0.1$) and vary the coefficient of

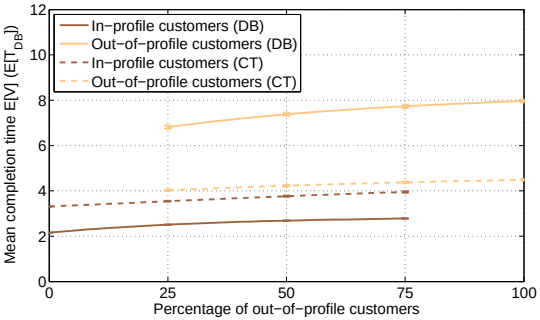


Figure 2.11: Completion time for DB and CT with varying percentage of out-of-profile customers ($\rho = 0.7, n = 100$).

variation of their request sizes $c_{var}[B]$. Figure 2.10 shows that with data beaming the completion time for both in-profile and out-of-profile customers increases with the variability of the size of the requests sent by out-of-profile customers. However, out-of-profile customers need to wait much longer and the completion time of requests from in-profile customers is bounded by 2.8 times the mean transmission time $E[T_{DB}]$. The reason for the increase of the completion time for all customers is the fact that the transmission resource takes longer to be released when long requests are served which is more likely with more variable request sizes. With concurrent transmission, the completion time is independent of the variability of the request sizes and it is the same for in-profile and out-of-profile customers. In particular, it is significantly larger than the completion time for in-profile customers with data beaming.

Varying Percentage of Out-of-Profile Customers

The last experiment focuses on varying percentage of out-of-profile customers. We model out-of-profile customers with a coefficient of variation of the transmission request inter-arrival time of $c_{var}[A] = 2$ and of the request size of $c_{var}[B] = 2$. Figure 2.11 shows the completion time for in-profile and out-of-profile customers for different percentages of out-of-profile customers. For in-profile data beaming customers, the completion time is not larger than 2.9

times the mean transmission time $E[T_{DB}]$ while it is almost three times larger for out-of-profile customers than for in-profile customers. This is different with concurrent transmission where the completion time for in-profile and out-of-profile customers does not differ a lot. Thus, data beaming subscriptions are useful for customers if they want to have short waiting times until transmission. This is a kind of priority service and may be charged. Moreover, the subscriptions give hints for resource provisioning to the network operator.

2.4 Evaluation of DCI Solutions

The preceding section presented our optimized architecture for VM migrations which satisfies the special requirements regarding the data transfer during the migration process. In the following, we take a closer look at data center interconnect solutions and analyze their address resolution scalability. We focus on VPLS and OTV as exemplary protocols as explained in Section 2.2.2. First, we present VPLS and OTV in detail and explain the ARP traffic handling. Second, we review ARP implementation characteristics and present our ARP model. Third, we apply our ARP model to VPLS and OTV and quantify the total address resolution traffic. In the last part, we discuss how an ARP proxy can reduce the total address resolution traffic for both approaches.

2.4.1 Virtual Private LAN Services

In this section, we give an overview of VPLS, explain how unicast traffic is handled in general and then describe the broadcast handling by means of an ARP resolution process between two nodes in different data center Ethernet networks.

Architecture Overview

VPLS is a standardized mechanism to connect Ethernet domains over a MPLS core. VPLS is implemented in *Provider Edges (*PEs) and the PEs are connected via a full mesh of MPLS tunnels among each other. The provider edges apply

data plane learning on all interfaces and learn the mapping from source MAC address to ingress interface. If the destination of an outgoing packet is not known, the packet is flooded via the full mesh to all other connected provider edges. To avoid loops in the full mesh of MPLS tunnels, a provider edge does not forward incoming packets from one MPLS tunnel to another MPLS tunnel which is known as "split horizon" rule.

Forwarding Procedure for Unicast Traffic

Considering the forwarding procedure for unicast traffic at provider edges, we distinguish between outgoing and incoming Ethernet frames.

Data Plane Learning: if a provider edge receives an Ethernet frame on one of its interfaces, it first adds or updates the entry for the source MAC address in its local MAC table. The table stores the mappings from MAC address to outgoing interface for a specific VPLS instance. The frame is then further processed depending on the communication direction.

Outgoing frames from internal nodes: for outgoing frames received via one of the local interfaces, the provider edge checks in its internal MAC table, whether there is an entry for the destination MAC address. If there is an entry in the table, the provider edge forwards the Ethernet frame on the associated MPLS tunnel. In case there is no entry, the Ethernet frame is broadcast on all MPLS tunnels belonging to this VPLS instance. Therefore, the provider edge replicates the packet and forwards it on the appropriate MPLS tunnels.

Incoming frames from external nodes: for incoming frames received via one of the MPLS tunnels, the provider edge also checks in its internal MAC table, whether there is an entry for the destination MAC address. If the entry points to another MPLS tunnel, the Ethernet frame is discarded to avoid a possible loop in the full mesh of MPLS tunnels. If the entry points to an internal interface, the MPLS header is removed and the frame is forwarded on the internal interface.

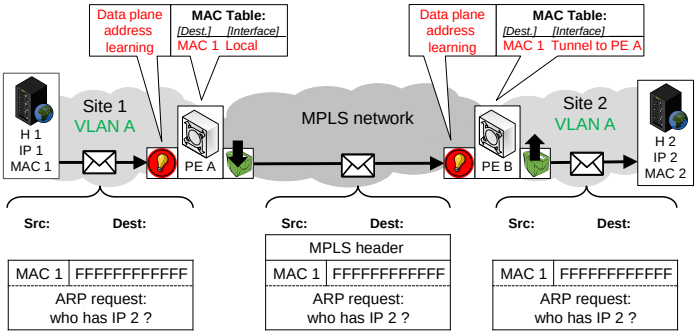


Figure 2.12: ARP broadcast request from H 1 to H 2.

In case there is no entry for the destination MAC address, the packet is broadcast on all interfaces except the interfaces pointing to an MPLS tunnel. This again avoids loops in the full mesh of MPLS tunnels.

ARP Traffic Handling

The following scenario describes the ARP traffic handling of VPLS by means of a communication example between two endhosts (H 1 and H 2) located in different customer sites, see Figure 2.12. Endhost H 1 has IP address IP 1 and a MAC address MAC 1. Endhost H 2 has IP address IP 2 and a MAC address MAC 2. H 1 knows the IP address of H 2 and the first step of the well-known ARP resolution is to get the MAC address for H 2. Therefore, H 1 sends an ARP request to discover the MAC address of H 2. The ARP request is sent to the broadcast MAC address FF:FF:FF:FF:FF:FF and the source address is the MAC address of H 1 (MAC 1). The frame arrives at PE A which then performs the VPLS forwarding process. First, PE A learns that MAC 1 can be reached locally and stores the appropriate entry in its MAC table. The destination MAC address is the broadcast MAC address, hence PE A floods the packet to all other PEs (PE B). PE B learns that MAC 1 can be reached via the MPLS tunnel to PE A and stores this information along with the MAC address in its MAC table. PE B then removes the

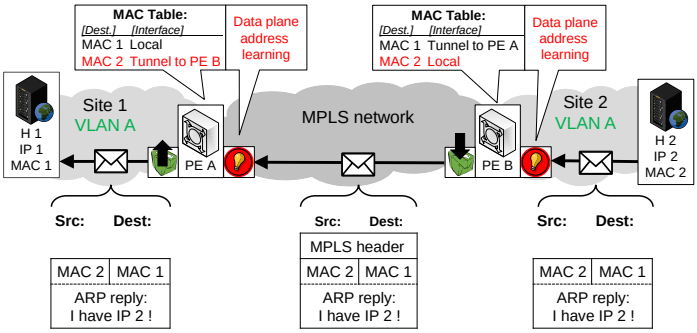


Figure 2.13: ARP unicast reply from H 2 to H 1.

MPLS label and floods the frame on its site-faced local interfaces. It does not flood the packet over MPLS tunnels because of the split horizon rule. Eventually, the broadcast frame arrives at H 2.

H 2 now responds to the ARP request with an ARP reply, see Figure 2.13. Therefore, H 2 sends a frame addressed to MAC 1 and uses its own MAC address MAC 2 as source address. The frame arrives at PE B, which learns that MAC 2 can be reached locally and stores this information in its MAC table. PE B already knows that MAC 1 can be reached via the MPLS tunnel to PE A and adds the appropriate MPLS header. The frame is then only forwarded to PE A, which receives the frame and learns that MAC 2 can be reached via the MPLS tunnel to PE B. PE A removes the MPLS header and forwards the frame according to the entry in its MAC table. Eventually, H 1 receives the frame.

2.4.2 Overlay Transport Virtualization

This section gives an overview of OTV, explains important building blocks, and shows how unicast traffic is handled in general. Further, the broadcast handling by means of an ARP resolution process between two nodes in different data center Ethernet networks is described.

Architecture Overview

OTV is designed to connect Ethernet domains over an IP core network. OTV is implemented in *Edge Devices (*EDs) and there has to be at least one edge device per participating Ethernet domain. Ethernet frames destined from one domain to another domain are tunneled between the edge devices of the different domains. To find the corresponding edge device for an outgoing frame, a mapping inside the edge devices is required. OTV utilizes a separate overlay control plane to distribute these mappings among the participating domains. In the following, we give further details regarding the different building blocks of OTV.

Transport Overlay

OTV establishes IP tunnels between the edge devices of participating Ethernet domains to transparently connect them over an IP core network. Each edge device is therefore configured with a globally reachable IP address. An edge device adds an additional outer IP header to outgoing Ethernet frames and uses its own IP address as source address. The destination address depends on the type of outgoing traffic. Unicast traffic is tunneled directly to the address of the destination's edge device. Broadcast traffic in contrast is sent via an *Any Source Multicast (*ASM) group and all participating edge devices join this ASM group. The ASM group is also used to convey the signaling messages for the overlay control plane. Broadcast frames or signaling messages are then encapsulated to the ASM group address and are delivered to all participating edge devices.

Composition of Mapping Tables

Edge devices learn the MAC-to-IP mappings for endhosts located in another Ethernet domain via the core-faced overlay interfaces and a separated overlay control plane. IS-IS [74] is used in this control plane to distribute MAC reachability information among all participating edge devices. Once an edge device sees a new MAC address on one of its site-faced internal interfaces, it distributes the mapping from the new MAC address to its own IP address in the overlay control plane

via an IS-IS *Link State Packet (*LSP). Other edge devices receive these LSPs and store the mappings in an internal MAC-to-IP mapping table.

Forwarding Procedure

When an edge device receives a frame destined to an endhost in another participating Ethernet domain, it first performs a lookup in its internal mapping table. The MAC address in the destination field of the outgoing Ethernet frame is resolved into the IP address of the appropriate edge device of the destination Ethernet domain. If there is an entry for the destination MAC address in the local mapping table, the Ethernet frame is tunneled to the address of the destination's edge device. If no entry is found, the frame is not flooded but discarded.

ARP Traffic Handling

In this example, we show how OTV handles ARP traffic which is generated due to a communication between two endhosts (H 1 and H 2) located in different customer sites, cf. Figure 2.14. Endhost H 1 has the IP address IP 1 and a MAC address MAC 1. Endhost H 2 has the IP address IP 2 and a MAC address MAC 2. H 1 knows the IP address of H 2 and the first step is to get the MAC address for H 2. Therefore, H 1 broadcasts an ARP request to discover the MAC address of H 2. The destination address of the ARP request is the MAC broadcast address FFFFFFFF and the source address is the MAC address of H 1 (MAC 1). ED A receives the ARP request and performs the following steps.

At first, the edge device has learned the MAC address of an endhost located in its own site and thus sends an IS-IS LSP, which contains the MAC address of H 1 (MAC 1) and its own IP address (IP A). This advertisement only happens if an Ethernet frame arrived on an internal site-faced interface and if this frame's source MAC address is not yet known by the edge device. If the source MAC address is already known, the edge device does not send out an advertisement. The next step for ED A is to send the ARP request to all other participating Ethernet sites. This is done by sending the Ethernet frame to the ASM group. Therefore,

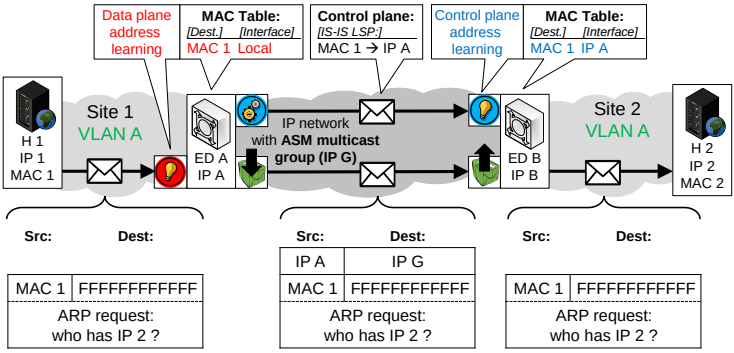


Figure 2.14: ARP broadcast request from H 1 to H 2.

the ED A adds an outer header that contains the IP address of the ASM group (IP G) in the destination address field and its own IP address (IP A) in the source address field. The packet is then forwarded, due to the multicast state in the core, to all subscribed edge devices of the ASM multicast group G.

The ED B of the customer site 2 now receives the IS-IS LSP with the mapping for MAC 1 (MAC 1 → IP A) and stores this mapping in its MAC-to-IP mapping table. Once ED B also receives the ARP request, it decapsulates the frame by removing the outer header and broadcasts the frame via its internal interfaces in its attached customer site (site 2). The ARP request eventually arrives at H 2 which recognizes its own IP address (IP 2) in the frame.

H 2 then responds with an ARP reply, cf. Figure 2.15. The reply is sent from H 2 to H 1 using MAC 2 as source address and MAC 1 as destination address. The ARP reply arrives at ED B of site 2. ED B has already learned the mapping for MAC 1 (MAC 1 → IP A) and adds an outer header containing IP A as destination address and its own IP B as source address. In addition, ED B also advertises the mapping for MAC 2 (MAC 2 → IP B) via an IS-IS LSP so that other edge devices learn that H 2 is located in the Ethernet domain behind ED B.

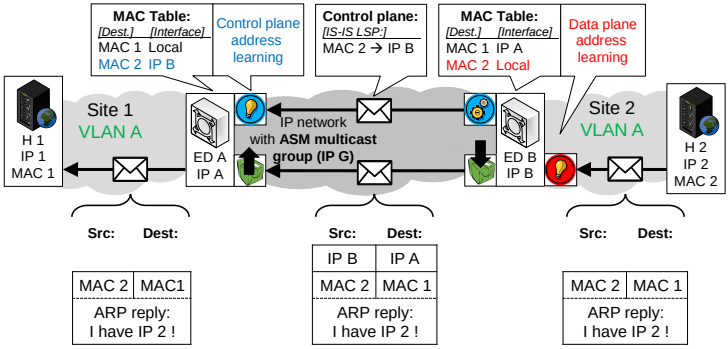


Figure 2.15: ARP unicast reply from H 2 to H 1.

The encapsulated ARP reply packet is unicast forwarded to ED A. ED A removes the outer header and forwards the ARP reply using MAC 1 to the destination H 1. H 1 eventually receives the packet and is now able to send further packets to H 2 using the MAC address MAC 2.

2.4.3 Analytical Modeling

In this section, we present an analytical model for the rate of ARP broadcasts per server in an Ethernet domain. Even though ARP is a standard protocol, we are not aware of models for the resulting traffic. Therefore, we first describe the behavior of state-of-the-art ARP implementations. Then, we introduce our assumed scenario and our analytical model.

ARP Implementation Characteristics

In this paragraph, we detail the ARP implementations in current operating systems that were the basis for the developed ARP model. We verified the following behavior for Linux (Ubuntu 10.10) and Microsoft Windows 7 Professional. The current implementation of the ARP kernel module both in Windows [75] and

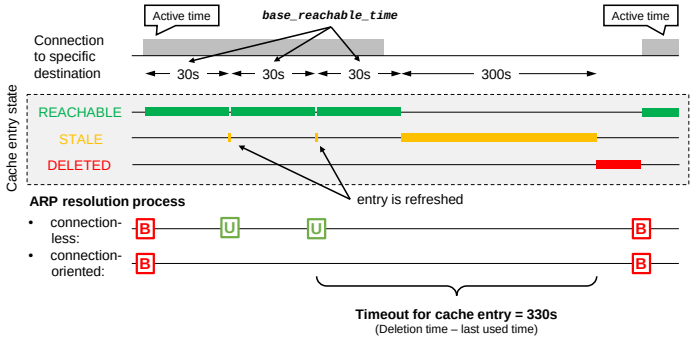


Figure 2.16: Visualization of state transitions for ARP cache entries.

Linux [76] follows RFC 4861 [77], which describes the network to link layer address resolution via the neighbor discovery protocol in IPv6.

If a source host initiates a connection to a destination IP address with unknown MAC address, ARP is used to obtain the MAC address. The source broadcasts an ARP request and the owner of the destination IP address sends an ARP reply in return, which contains the mapping from IP to MAC address. The source host stores this information in an ARP cache. The destination host also stores the mapping for the source hosts in its own cache, but it first checks whether this mapping is valid. For connection-less protocols like UDP, this induces an ARP resolution in the opposite direction. For connection-oriented protocols like TCP, the ACK packet is sufficient and no additional ARP resolution process is initiated.

The ARP cache inside both hosts is used to reduce the number of ARP broadcast requests. To avoid outdated entries, each entry is assigned with a timeout. Once the timeout expires, the entry is removed from the ARP cache. In addition, each entry has a certain state which indicates the validity of the entry before the entry is eventually deleted. A visualization of the different states and the corresponding transitions due to lookups by higher layer applications can be seen in Figure 2.16. A newly created entry gets the *REACHABLE* state and can be used by higher layer applications without a refresh of its validity.

If the entry is not used by higher layer applications for a random time between $base\_reachable\_time/2$ and $base\_reachable\_time \cdot 3/2$, the state of the entry is changed to *STALE*. The parameter $base\_reachable\_time$ is usually set per default to $30\,s$ [77]. *STALE* entries need to be refreshed if they are used again. For connectionless protocols like UDP, this again requires an ARP request. However, this ARP request is then sent per unicast and not per broadcast. For connection-oriented protocols like TCP, ACK packets of successful connections can be used to refresh the entry. Refreshed entries get the *REACHABLE* state again. The resulting ARP resolution processes are shown in the lower part of Figure 2.16. The rectangles with *B* indicate a broadcast ARP resolution process and rectangles with *U* indicate a unicast ARP resolution process. Eventually, if a *STALE* entry is not refreshed, it is deleted after a certain time span. The length depends on the configuration but a common value is $300\,s$.

In summary, for TCP, the above described behavior results in a timeout for cache entries of approximately $330\,s$. A new outgoing TCP connection induces an ARP broadcast request if this specific address has not been refreshed for $330\,s$. For UDP, the timeout for cache entries is $30\,s$ and entries can only be refreshed with an ARP request. Hence, an outgoing UDP connection induces an ARP request every $30\,s$. However, an ARP request for UDP is only broadcast if there is no entry in the ARP cache. Regarding the ARP broadcast requests, UDP and TCP thus show the same behavior (see lower part of Figure 2.16) and an ARP broadcast request to a specific destination is only sent if the address has not been used for $330\,s$. Nevertheless, in the following sections we only consider the TCP behavior because most traffic in large data centers usually uses TCP. However, the model can easily be adapted for UDP traffic.

Scenario and Assumptions

We assume a geographically dispersed data center with D locations and a maximum number of N connected servers per considered VLAN. For the numerical evaluations, N is set to 10000. This results in N/D servers for the VLAN per

data center location. Furthermore, we suppose that each server initiates flows to random destinations with a certain arrival rate. This workload model is of course simple and it neglects many details about the complex load distribution mechanisms inside a data center, but it is difficult to provide a better model that is still generally applicable. However, the measurements of the exchanged bytes between server pairs in [78] confirm that a random workload scenario is a reasonable assumption. We vary the arrival rate of flows per server λ_{server} between 10^{-4} and 10^4 flows per second, in order to consider a wide range of possible load situations. Regarding the flow duration, we assume short flows that are at least one order of magnitude smaller than the cache timeouts in hosts. This assumption is valid as according to [79], 99% of the flows in a data center transmit less than $100\,MB$, which results in a flow duration in the order of seconds.

Analytical Model: ARP Rate per Server

In this section, we model the ARP rate per server λ_{ARP} dependent on the rate of outgoing flows per server λ_{server}. We assume that the time A between two consecutive flows a server initiates towards another server is exponentially distributed with rate λ_{server}. According to [80], this is a reasonable approximation for the traffic within a data center. As there are $N-1$ other servers in the VLAN, the considered server contacts a specific other server only with a rate of $\lambda_{server}/(N-1)$. If the server does not find an entry in the ARP cache, it broadcasts an ARP request in the VLAN. This happens if the server initiated no other flow to that same destination for more than T_{host} time. Thus, the corresponding probability is

$$p_{ARP} = P(A > T_{host}) = exp(-\frac{\lambda_{server}}{N-1} \cdot T_{host}), \tag{2.5}$$

so that the overall rate of ARP messages issued by a server is

$$\lambda_{ARP} = \lambda_{server} \cdot p_{ARP}. \tag{2.6}$$

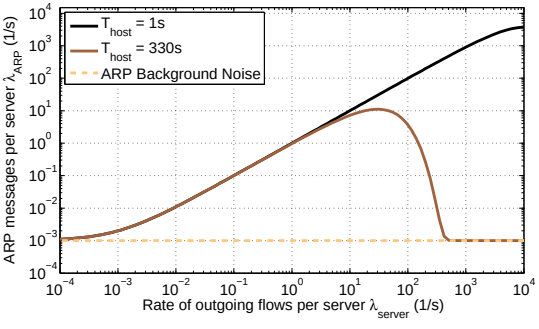


Figure 2.17: Rate of outgoing ARP messages per server.

In addition to the ARP rate due to outgoing flows, we also assume a certain base rate λ_{base} of outgoing ARP requests per server. This base rate may be caused by configuration or signaling protocols like DHCP, automated updates, or administration tasks. It can be seen as a kind of ARP background noise and we assume a rate of 10^{-3} ARP requests per second. This rate is added to Equation 2.6 and we then get a total rate of ARP broadcasts per server as

$$\lambda_{ARP} = \lambda_{server} \cdot p_{ARP} + \lambda_{base}. \tag{2.7}$$

In Figure 2.17, we plot λ_{ARP} against the rate of outgoing flows λ_{server} for $T_{host} = 330\ s$. We also present results of a hypothetical value of $T_{host} = 1\ s$ in order to show how our model behaves for very short timeout values. The x-axis shows the rate of outgoing flows per server λ_{server} and the y-axis shows the ARP rate per server λ_{ARP}. Both axis are logarithmically scaled. The ARP rate per server increases about linearly with the rate of outgoing flows per server λ_{server} for values below $\lambda_{server} < 10$ flows per second. It is almost identical because in that range, the server finds almost never a matching entry in its ARP cache as entries are deleted from the cache before the same destination is contacted again. For larger rates of outgoing flows per server, the ARP rate depends on the value for the timeout interval T_{host}. For $T_{host} = 330\ s$, the rate of ARP broadcasts rapidly decreases for rates λ_{server} larger than 10 flows per second. The smaller the timeout interval T_{host}, the later the ARP rate drops to the base rate λ_{base}.

Analytical Modeling of IS-IS Messages

In case of pure data plane learning like in VPLS, the Ethernet address resolution prior to each communication ensures that the necessary entries are learned in the MAC tables of the VPLS switches (cf. Section 2.4.1). Hence, no additional signaling messages are triggered in this case. However, in case of control plane address learning like in OTV, additional signaling messages are triggered by the Ethernet address resolution, and they have thus to be taken into account, too.

In this section, we present an analytical model for the rate of sent and received IS-IS signaling messages per OTV switch. We focus on the LSPs which carry the mapping information and neglect other signaling message like periodic hello messages as they constitute only a small amount. At first, we give a short overview of IS-IS and explain how it is used in OTV according to the available documentation. Then we describe a model that can be used to quantify the base receiving rate of IS-IS messages per OTV switch.

IS-IS Overview

IS-IS [74] is a link-state routing protocol and designed for use within an administrative domain. An IS-IS node uses Link State Packets (LSPs) to flood its reachability information to other participating network nodes. Each node then stores the received LSPs in a database and uses the Dijkstra algorithm to compute the shortest paths to all other nodes. Each LSP in the IS-IS database has a timeout interval *max-LSP-lifetime*, in the following denoted as Δ_{max}. A default value for this parameter is $1200\,s$ (cf. [74, 81]). To avoid expiration of cached entries and to keep them up to date, each node floods all its advertised reachability information every *LSP-refresh-interval* seconds (labeled Δ), which is per default set to $900\,s$ according to [74, 81]. If an entry is not refreshed within its *max-LSP-lifetime*, the entry gets deleted. To avoid excessive flooding, each node is only allowed to send 30 LSPs per second. This parameter is configured by the value *LSP-interval*, which is usually set to $33\,ms$ [74, 82].

IS-IS as Control Plane Protocol for OTV

Each OTV switch uses IS-IS LSPs to advertise the mappings from its IP address to the MAC addresses of local connected servers. These advertisements can be seen as a MAC reachability information. Other participating OTV switches listen on these advertisements and store the flooded LSPs in their LSP database which acts as mapping database. If a mapping is not refreshed by the advertising OTV switch in time, it is deleted from the mapping databases in all other OTV switches. Therefore, we suppose that each OTV switch refreshes its advertised information every *LSP-refresh-interval* seconds, i. e., once per Δ. The rate for these refreshes depends on the number of advertised LSPs and Δ.

Base Rate of IS-IS Messages

Each OTV switch is authoritative for the mappings of its local servers and advertises their MAC reachability once it sees a flow arriving from an unknown source address on one of its local interfaces. The LSP refresh rate depends on the number of entries in the mapping database of an OTV switch and the length of the timeout interval Δ_{max}.

In order to compute the number of entries, we again assume that the time A between two consecutive flows from a specific server is exponentially distributed with rate λ_{server}. An entry for this specific address is in the LSP database if and only if the inter-arrival time A is smaller than the LSP lifetime Δ_{max}. The corresponding probability is then

$$p_{entry} = P(A \leq \Delta_{max}) = 1 - exp(-\lambda_{server} \cdot \Delta_{max}). \tag{2.8}$$

The rate of IS-IS LSPs λ_{LSP} per individual entry is then

$$\lambda_{LSP} = \frac{p_{entry}}{\Delta}. \tag{2.9}$$

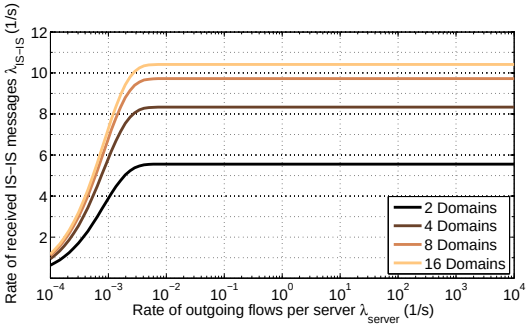


Figure 2.18: Rate of received IS-IS messages per OTV switch for different number of domains.

The total receiving rate of IS-IS LSPs per OTV switch is then Equation 2.9 times the number of remote servers

$$\lambda_{IS-IS} = \frac{N}{D} \cdot (D-1) \cdot \lambda_{LSP}. \tag{2.10}$$

In Figure 2.18, we plot λ_{IS-IS} according to Equation 2.10 and vary the number of domains D. For rates λ_{server} larger than 0.001 flows per second, we get the maximum rate of received LSPs, since the rate per server is sufficiently high so that there is always an advertised entry per server. The number of domains also influences the rate of received IS-IS messages as with increasing number of domains, more servers lie outside the own domain and hence more LSPs arrive from connected OTV switches.

The evaluation in Figure 2.18 was done using the default value of $900\,s$ for the parameter Δ, i. e., *LSP-refresh-interval*. In Figure 2.19, we plot the receiving rate again for four domains but for different values of the Δ.

With decreasing refresh interval Δ, the rate of received IS-IS messages increases rapidly and hence care has to be taken how to set the refresh interval. If the interval is too big, the LSP database may contain outdated entries but if the interval is too small, the network is flooded with IS-IS LSP updates.

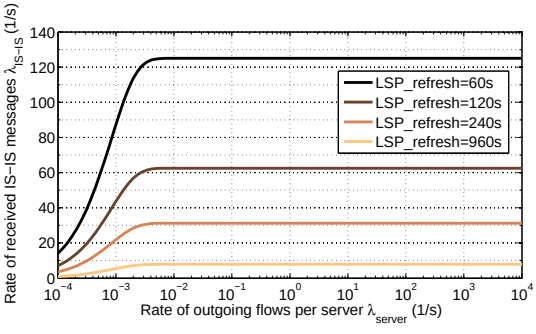


Figure 2.19: Rate of received IS-IS messages per OTV switch for four domains and different refresh intervals.

2.4.4 Address Resolution Performance

In this section, we combine the findings of the previous section to quantify the amount of received broadcast messages per VPLS and OTV switch that are caused by the address resolution mechanism.

Total Address Resolution Traffic for VPLS and OTV

In the following, we assume a benign environment in which ARP traffic is mainly triggered by the steady-state communication between the servers inside the data center. We thereby neglect specific situations such as broadcast storms caused by failures, which are more difficult to model.

Concerning control traffic, the main performance metric of interest is the rate of received broadcast messages. For a VPLS switch, the received broadcast messages comprise the ARP requests. In case of OTV, in addition to the ARP requests also the IS-IS LSPs have to be considered. Concerning the traffic originating within one domain, the total rate of ARP broadcast requests is the individual rate per server times the number of servers per location. The number of servers per location is N/D and hence the total rate of ARP broadcasts per domain is

$$\frac{N}{D} \cdot \lambda_{ARP}. \tag{2.11}$$

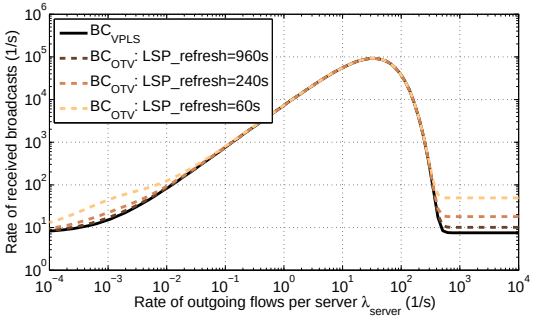


Figure 2.20: Rate of received broadcasts per VPLS and OTV switch (D=4).

These ARP requests are broadcast within the own location and between the different locations belonging to the same data center. Both VPLS and OTV switches receive the rate according to Equation 2.11 from each of their $D-1$ connected neighbors. Hence in total, a switch receives in both cases

$$\frac{N}{D} \cdot (D-1) \cdot \lambda_{ARP} \tag{2.12}$$

ARP requests per second. In addition, an OTV switch also receives the broadcast IS-IS LSPs. The total rate of received broadcasts per OTV switch is then the sum of the received ARP requests and the rate of received IS-IS LSPs.

In Equations 2.13 and 2.14, we summarize the findings for VPLS and OTV:

$$BC_{VPLS} = \frac{N}{D} \cdot (D-1) \cdot \lambda_{ARP}, \tag{2.13}$$

$$BC_{OTV} = \frac{N}{D} \cdot (D-1) \cdot (\lambda_{ARP} + \lambda_{LSP}). \tag{2.14}$$

Numerical Results

In Figure 2.20, we plot BC_{VPLS} and BC_{OTV} for $D = 4$ domains. In addition, we vary the parameter Δ (i. e., *LSP-refresh-interval*) between $60\,s$, $240\,s$, and

$960\ s$. The solid line denotes the broadcast messages received at an VPLS switch and the dashed lines show the corresponding values for an OTV switch. A VPLS switch only receives the ARP requests and hence the solid line only comprises the ARP broadcast requests. OTV switches in addition receive the broadcast IS-IS LSPs and hence the dashed lines also comprise the broadcast IS-IS LSPs.

For a low or very high load, i. e., for rates λ_{server} smaller than 0.1 or larger than 100, the rate of IS-IS LSPs constitutes a significant amount of the total number of received broadcast messages per OTV switch. Also, the smaller the parameter *LSP-refresh-interval*, the larger the amount of IS-IS LSPs.

However, between rates λ_{server} of 0.1 and 100, the amount of IS-IS LSPs compared to the number of ARP broadcast requests is rather small and has no significant impact on the overall control traffic. As a consequence, in this parameter range VPLS and OTV perform similarly and there is no significant difference in signaling overhead due to ARP traffic.

2.4.5 Improvement by an ARP Proxy

An ARP proxy is a well-known solution to improve the scalability of Ethernet address resolution. In this section, we model an ARP proxy and we show how such a proxy in the edge switches reduces the number of ARP broadcast requests between data center sites.

ARP Proxy Basics

ARP proxies are usually implemented in customer edge switches. They snoop ARP traffic and cache the mappings from IP to MAC address seen in the ARP reply packets. The ARP proxy sees the ARP replies from hosts outside its own domain as these ARP replies pass through its interfaces. The cache inside the ARP proxy can thus be seen as an aggregate of the ARP caches of the servers in the local domain. If a server in that domain asks for an already cached IP address, the ARP proxy generates an ARP reply locally rather than broadcasting the ARP request to other domains. As a result, the ARP proxy reduces the number of ARP

broadcast requests between the different domains. A more detailed description of an ARP proxy can be found for example in [83].

Entries in an ARP Proxy Cache

In the following, we describe a model how to estimate the number of entries inside an ARP proxy. First of all, we need to calculate the outgoing rate of ARP requests λ_{dest} that arrive at an edge switch for an individual address from the servers within this domain. Since there are in total N servers in this distributed data center, λ_{dest} can be calculated by dividing the total rate (cf. Equation 2.11) by N:

$$\lambda_{dest} = \frac{1}{D} \cdot \lambda_{ARP}. \tag{2.15}$$

The number of mapping entries inside the ARP proxy cache depends on the timeout T_{cache} for ARP cache entries inside the proxy. For the numerical results presented in the following, we assume the same value as for the ARP cache timeout in the servers ($T_{cache} = 330\,s$). To calculate the number of entries inside the cache, we first need the probability p_{entry} that a specific remote address is stored in the cache. A mapping for a specific remote address is in the cache if and only if the time A between two consecutive ARP requests for this address is smaller than the timeout interval T_{cache}. If we again assume that the time A is exponentially distributed with rate λ_{dest}, the corresponding probability is

$$p_{entry} = P(A \leq T_{cache}) = 1 - exp(-\lambda_{dest} \cdot T_{cache}). \tag{2.16}$$

The total number of mapping entries is then

$$n_{proxy} = \frac{N}{D} \cdot (D-1) \cdot p_{entry}. \tag{2.17}$$

In Figure 2.21, we plot the number of mapping entries inside the proxy ARP cache for $D = 4$ domains. As a reference, we also depict the number of entries in the ARP cache of one individual server. This value can be calculated similarly to

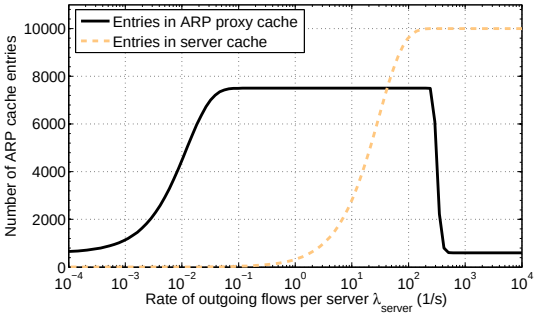


Figure 2.21: Number of entries inside ARP proxy cache and server ARP cache.

Equations 2.16 and 2.17 by replacing λ_{dest} by $\frac{\lambda_{server}}{N-1}$ and by multiplying with $(N-1)$ instead of $\frac{N}{D} \cdot (D-1)$. This is because a server cache contains entries for both the remote and the local servers. The number of entries inside server ARP caches is thus

$$n_{server} = (N-1) \cdot (1 - exp(-\frac{\lambda_{server}}{N-1} \cdot T_{host})). \tag{2.18}$$

For a low load, i. e., for rates λ_{server} smaller than 0.1, the number of entries inside the ARP proxy cache slowly increase until they reach the maximum of $\frac{N}{D} \cdot (D-1)$ entries. In this parameter range, the increasing rate λ_{server} leads to an increasing rate λ_{dest} at the ARP proxy (cf. Equation 2.7 and 2.15) which then leads to an increasing number of cache entries in the ARP proxy (cf. Equation 2.16 and 2.17). For a medium load, i. e., for rates λ_{server} larger than 1 but smaller than 100, the number of cache entries inside the server cache increase because of the increasing rate λ_{server} (cf. Equation 2.18). Hence, the number of sent ARP broadcasts per server decreases and the cache entries for remote servers shift from the ARP proxy cache to the server cache. For a high load, i. e., for rates λ_{server} larger than 100, the ARP caches inside the servers nearly contain all possible N destination addresses and hence no ARP broadcasts are sent anymore. The small number of cache entries for rates larger than 100 come from the assumed ARP background noise λ_{base} (cf. Equation 2.7).

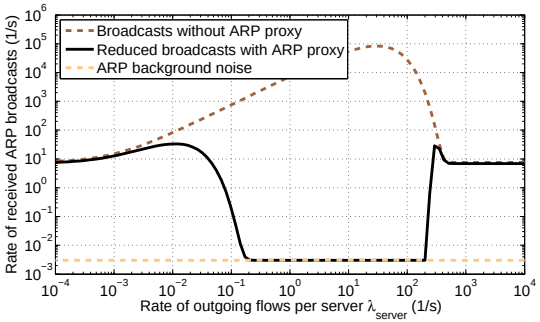


Figure 2.22: Rate of received ARP broadcasts with and without ARP proxy.

ARP Broadcast Reduction with ARP Proxy

Now, we apply the model for the ARP proxy to the receiving rate of ARP broadcasts at a customer edge switch. The ARP proxy of a site broadcasts an ARP request to the other sites only of it cannot find a matching entry for that request in its ARP cache. Hence, the probability $p_{broadcast}$ is complementary to the probability p_{entry} that there is an entry inside the ARP proxy cache. The reduced receiving rate is then

$$BC_{receiving} = (1 - p_{entry}) \cdot \frac{N}{D} \cdot (D - 1) \cdot \lambda_{ARP}. \tag{2.19}$$

In Figure 2.22, we plot the receiving rate with and without ARP proxy mechanism for $D = 4$ domains. The dashed line shows the receiving rate without ARP proxy and the solid line shows the receiving rate with ARP proxy. Again, we assume some kind of ARP background noise per domain.

According to Figure 2.22, there is a significant reduction of the control traffic for values of λ_{server} between 0.01 and 100. In this interval, nearly all remote destination mappings are stored in the ARP cache and hence no inter-domain ARP broadcasts are necessary, except for ARP background traffic. For lower and higher rates, not all entries are cached in the ARP proxy and hence the probability to broadcast an ARP request increases. For these rates, the ARP proxy is not as

efficient as for the interval between 0.01 and 100. In summary, our analytical model confirms that an ARP proxy is an effective method to reduce the control traffic between data center sites, and that it may make sense to deploy such a proxy in edge devices.

2.5 Lessons Learned

The objective of this chapter was to detail challenges in the area of physical infrastructure providing and to discuss possible solutions. In particular, we took a closer look at the interconnection of data center PIPs. The interconnection is required for mechanisms like service component mobility which typically comprises the migration of VMs between locations of one or several data center PIPs.

First, we analyzed the specific requirements of the migration process with respect to the data transfer and the underlying network. We showed that the migration process benefits from a short transmission time and presented our optimized data beaming architecture for server migrations. The data beaming architecture features a subscription model and a simple scheduler. Customers indicate transmission requests and receive time slots for high-speed data transfers after some time. In general, it is useful for applications that require high data rates but can wait some time until transmission starts.

In the current Internet, flows are transmitted concurrently and receive only a fraction of the physical bandwidth. Our simulations showed that the completion time of transmission requests is in most cases shorter for data beaming than for concurrent transmission. However, the main benefit of data beaming is that its transmission time takes only a fraction of the one of concurrent transmission.

The subscription model for data beaming helps to provision the network with sufficient resources. A scheduler enforces that transmission requests from customers who conform with their subscriptions are earlier served than transmission requests from customers who exceed their subscriptions or who have no subscriptions at all. Our simulations showed that the waiting time for in-profile customers is low, also in the presence of out-of-profile customers. The completion time for

in-profile users is also clearly shorter with data beaming than with concurrent transmission. Thus, subscriptions for data beaming offer a priority service to customers and may be charged. Overall, data beaming enables the network provider to offer new high-speed services to customers based on its flexible network infrastructure like GMPLS without giving the customer control over its network.

With respect to the network requirements, we explained that Layer 2 connectivity is required between source and destination data center. This is necessary so that the hypervisor at the destination node can inform switches and routers about the new location of the migrated VM. However, connecting a large number of Ethernet nodes to the same broadcast domain may result in scalability issues, in particular with respect to the ARP broadcast traffic. As this may limit the scalability of data center interconnect solutions, we studied the control traffic caused by address resolution for two selected technologies, i.e., OTV and VPLS.

We proposed an analytical model for the ARP traffic between data center locations that takes into account the number of hosts and connected sites. The model quantifies the amount of outgoing ARP broadcasts per server as a function of the outgoing flow rate. With a cache timeout of 330 s, the rate of ARP broadcasts per second linearly increases until the maximum is reached at a rate of 30 outgoing flows per second, resulting in 10 ARP broadcasts per server and second.

With this model, we quantified the ARP traffic for VPLS and OTV. As OTV uses IS-IS as additional control plane protocol, we also included a model which quantifies the number of sent LSPs as a function of the LSP refresh timer.

According to our numerical results, we showed that despite of the different realization concepts, the two address resolution mechanisms result in a similar signaling load for rates between 0.1 and 100 outgoing flows per server and second. For a low or very high load however, i.e., for rates smaller than 0.1 or larger than 100, the rate of IS-IS LSPs constitutes a significant amount of the total number of received broadcast messages per switch. As a consequence, in this parameter range, VPLS causes less signaling overhead due to ARP traffic than OTV.

In addition to the quantification of the signaling overhead, we studied how an ARP proxy can improve the overall scalability by reducing the ARP broadcast

traffic. The highest reduction with respect to the received ARP broadcasts can be achieved for medium load, i.e., for rates between 0.1 and 100 flows per server and second. Overall, our proposed models can be used to estimate the address resolution traffic for data center interconnect solutions and to assess the effective reduction due to an installed ARP proxy.

3 Endpoint Mobility

Enabling service component mobility via VM migrations between virtual networks of different data center locations requires transparent Layer 2 connectivity so that hypervisors are able to announce the new location of migrated VMs via gratuitous ARP announcements. These announcements are broadcast within the created extended subnet so that all Layer 2 nodes update their stored information about the migrated VM. The extended subnet spans the different data center locations and comprises the transparently connected virtual networks. Due to the rigid current routing architecture, migrating VMs within an extended subnet leads to challenges with respect to ingress path, i.e., the location at which packets enter the extended subnet. After the migration process is finished between old and new location, packets are still first forwarded to the old location since IP routing entries for the IP of the migrated VM still point to that location. The underlying problem is that the current IP address both serves as identifier on transport layer and as routing locator on network layer. Existing transport connections are bound to the IP address and in addition, the current network attachment point is defined by the IP prefix to which the IP address belongs. Hence, to avoid the detour via the old location for the traffic to the migrated VM, a new flexible routing architecture is required that decouples identification and location information of today's IP naming and addressing architecture.

The *Locator/ID Separation Protocol (*LISP) has received the most attention and is standardized in an own IETF working group [84]. However, ingress path optimization via LISP is only possible if customer access networks or mobile endpoints are LISP-enabled. The *LISP Mobile Node (*LISP-MN) extension proposes modified network stacks for mobile nodes and enables them to roam be-

tween LISP and non-LISP networks. Hence, LISP in combination with LISP-MN enables seamless service component and endpoint mobility with optimized ingress path. A demonstration of such a scenario was shown at the VMWorld 2012 conference [85].

There are still several open challenges with respect to LISP-MN in such a scenario. First, the mobility related forwarding of LISP-MN is not always optimal and we propose a set of improvements to optimize the forwarding behavior. The optimizations aim at several issues on control and data plane and either reduce the signaling delay or the encapsulation overhead due to applying LISP. One such improvement is the usage of a local mapping service inside local domains to reduce the delay due to mapping lookups. Another severe challenge is that mobile nodes behind NAT boxes are not supported by LISP-MN. Hence, we present a NAT traversal mechanism to restore connectivity behind NAT boxes. Finally, the different handover mechanisms proposed by LISP-MN have not been studied so far for a video streaming scenario. To solve this, we have implemented LISP in a simulation framework to evaluate both the efficiency of our proposed improvements as well as the handover performance for the video streaming use case.

Large parts of this chapter are taken from [6, 7]. In addition, the chapter contains material and results from [9]. The chapter is structured as follows. First, we detail the ingress path optimization challenge for extended subnets and explain LISP along with its extensions. Second, we present related work in the area of future Internet routing and mobility support architectures. Third, we discuss the identified disadvantages related to the LISP-MN extension and present our proposed improvements as well as the NAT traversal mechanism. Finally, we study the efficiency of our proposed improvements, evaluate the handover performance, and conclude the chapter with pointing out the lessons learned.

3.1 Background

LISP is the most prominent future Internet routing protocol that enables ingress path optimization after VM migrations. The LISP protocol family comprises sev-

eral building blocks that address different use cases like interworking with non-LISP networks or mobility. Hence, we first give a short introduction to LISP and its building blocks and then explain how ingress path optimization after VM migrations is enabled with LISP.

3.1.1 LISP-enabled VM Migrations

LISP [38] is an implementation of the Loc/ID split and divides the IP address range into two different subsets. *Endpoint Identifiers (*EIDs) identify end-hosts on a global scale and are used to forward packets locally inside LISP domains. LISP domains are edge networks that are connected via LISP gateways to the core of the Internet, where globally routable addresses are used to forward packets. The globally routable addresses of LISP gateways are called *Routing Locators (*RLOCs). The communication between LISP nodes inside the same LISP domain does not change according to LISP. However, the communication between LISP nodes in different domains requires tunneling between the different LISP gateways. The gateways either act as *Ingress Tunnel Router (*ITR) or as *Egress Tunnel Router (*ETR). ITRs tunnel packets to other LISP gateways which then act as ETRs. Figure 3.1 shows a packet flow sequence for the communication between two LISP clients located in different LISP domains. ITR A receives packets addressed to EID 2 from an end-host in its own LISP domain. It keeps the *Inner Header (*IH) untouched and adds a UDP header addressed to the default LISP data port *4341* and an *Outer Header (*OH) with its RLOC (RLOC A) as source and RLOC B as destination address so that the packets are globally routable.

This procedure requires a mapping lookup to learn the appropriate RLOC (RLOC B) for the destination EID (EID 2). LISP does not mandate a specific mapping service but instead introduces *Map Servers (*MSs) and *Map Resolvers (*MRs) [86]. These two entities form an interface which facilitates the operation of LISP with different mapping systems. ETRs register the EID-to-RLOC mapping for all attached LISP nodes at their associated MS on the default LISP signal-

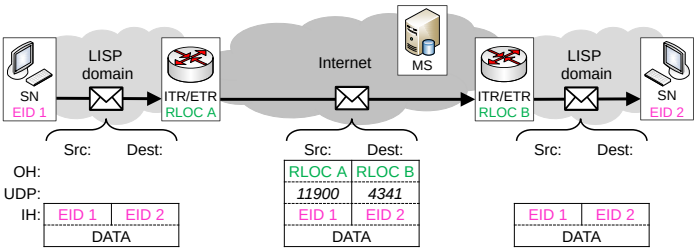


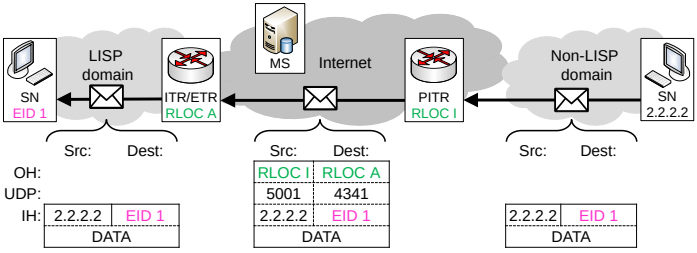
Figure 3.1: Packet flow sequence with LISP.

ing port *4342*. The EID-to-RLOC registration process and security mechanisms to protect the registration against for example spoofing attacks are described in Section 4.2 of [86].

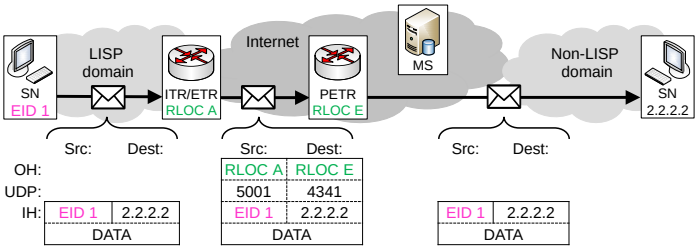
MSs listen on port *4342* and once the MS receives the registration, it distributes the mapping information within the mapping service so that MRs find the authoritative MS for a specific EID. ITRs query MRs for the RLOC of a specific EID. The MR initiates a map-request which is forwarded via the mapping service to the authoritative MS. Again, port *4342* is used for the map-request message. The MS responds with a map-reply message which contains the valid locator set for the queried EID. If a map-request contains an EID for which no locator is registered, the mapping service returns a negative map-reply. To reduce communication overhead, this query may also be answered from a local cache at the ITR [87]. MRs and MSs can either be deployed in separate nodes or inside ITRs and ETRs. The current most prominent mapping system is *LISP Alternative Topology (*LISP+ALT) [88].

Interworking with the non-LISP Internet

To communicate with nodes in the non-LISP Internet, additional interworking mechanisms are required which are proposed in the *LISP Interworking (*LISP-IW) architecture [89]. Figure 3.2 shows the packet flow sequence for a communication between *Stationary Nodes (*SNs) located in LISP and non-LISP



(a) Packet flow via PITR



(b) Packet flow via PETR

Figure 3.2: LISP interworking packet flow sequence.

networks. Normal IP nodes usually resolve the DNS name of a LISP node into an EID and use it as destination address. However, EIDs are not globally routable and thus the border router of the non-LISP domain discards the packets because of missing forwarding entries for EIDs. To solve this problem, LISP-IW proposes additional boxes called *Proxy-ITRs (*PITRs), which are located outside edge domains and advertise highly aggregated EID prefixes into BGP. This way, packets addressed to EIDs become globally routable and are forwarded to one of the PITRs. PITRs perform the same traffic processing as ordinary ITRs, i.e., they query the mapping system for an RLOC of the destination EID and encapsulate packets towards the returned RLOC (see Figure 3.2a).

In the reverse direction, LISP packets destined to non-LISP nodes are not encapsulated by ITRs and the EID remains in the source address field of outgoing packets. Since the EID is not part of the upstream providers address range, such packets might be dropped when the provider does source address filtering to ensure that outgoing packets carry only addresses from its own address range. In this case, LISP uses *Proxy-ETRs (*PETRs). LISP gateways encapsulate packets destined to non-LISP nodes and send them to a preconfigured PETR outside their own domain (see Figure 3.2b). The PETR decapsulates the packet and sends it to the destination node in the non-LISP Internet. This way, LISP bypasses the source address filtering of upstream providers. PETRs can also be used to connect LISP domains which use a different IP version than their upstream provider.

Overall, proxy LISP gateways enable communication initiation from non-LISP domains to LISP domains. Hence, we assume in the following that proxy LISP gateways are used for interworking between LISP and non-LISP domains.

Mobility Extension

LISP Mobile Node (LISP-MN) [39] introduces mobility and allows LISP nodes to roam into other domains. *Mobile Nodes (*MNs) have a permanent EID which is used for identification but not for forwarding. In contrast to non-LISP nodes and SNs in LISP domains, MNs have upgraded LISP-MN networking stacks. When a MN roams into a network, it receives a care-of-address (e.g., via DHCP) under which it is locally reachable in the destination domain and registers it as locator in the mapping system. When the MN roams into a non-LISP network, the obtained care-of-address is globally reachable and serves as RLOC for the MN. When the MN roams into a LISP domain, the obtained care-of-address is only site-locally reachable and serves just as *Local Locator (*LLOC). The term “LLOC” was not proposed in the LISP-MN extension [39], but we use it to facilitate the distinction between site-locally and globally routable locators (LLOCs, RLOCs). All LLOCs of a LISP domain are pre-registered in the mapping system together with the RLOCs of the domain’s ETRs. LISP-MN assumes that a MN forms a separate

LISP domain and implements ITR/ETR functionality for incoming and outgoing traffic except for DHCP traffic (see [39, Sect. 6]). To send traffic, a MN has to encapsulate outgoing traffic to some ETR or PETR, i.e., it must be configured with the RLOC of a PETR. To receive traffic, the traffic must be tunneled to the MN from some ITR, PITR, or another MN.

LISP-MN implicitly expects enhanced functionality for normal ITRs and PITRs to communicate with MNs in other LISP domains. When a packet is sent from a LISP domain to a MN in another LISP domain, the ITR receives the outbound packet addressed to the EID of the MN. It queries the mapping system with the destination EID of that packet and encapsulates the packet to the returned locator which is the LLOC of the corresponding MN. Then, the ITR queries the mapping system again with the returned LLOC and encapsulates the packet with the returned locator which is the RLOC of the ETR of the destination LISP domain (see [39, Sect. 9]). Thus, two lookups are required. As it is not possible to infer the locator type (RLOC, LLOC) from the returned mapping, ITRs and PITRs must always perform two mapping lookups as they do not know a priori whether the packet is destined to a MN. If the corresponding node is not a MN in a LISP domain, the second lookup yields a negative map-reply and the packet is encapsulated just once. In contrast to ITRs and PITRs, MNs query the mapping system and encapsulate packets only once.

Mapping Cache Update Mechanisms

If the mapping of a LISP stationary or mobile node changes due to a certain event, e.g., roaming into a different domain, the cached mappings in ITRs or proxy-ITRs of communication partners need to be updated. This is necessary so that ongoing connections can survive the roaming event.

The *Solicit Map-Request (*SMR) mechanism uses a special map-request message which is sent from the mobile node to ETRs of the most recent communication partners. After a roaming event, the mobile node sends map-request messages with the solicit bit set to all locators in its mapping cache. In the mapping

cache, the mobile node stores the RLOCs of ETRs, to which the mobile node has recently sent a packet to. The ITRs, which receive the solicit map-request message, first determine whether there is an entry for the EID in their mapping cache. If there is a mapping, they update it by querying the mapping service. This mechanism requires additional signaling messages after the roaming event but the cached mappings of communication partners are quickly updated. The actual timespan between a handover event and the completion of the mapping updates in the cache depends on the connection from the ITR to the map server of the mobile node. If the connection delay to the map server is rather small, this mechanism allows near real-time applications.

The *piggybacking* mechanism uses the ability to add mapping data to map-request messages. After a roaming event, the mobile node invokes the solicit map-request mechanism and adds its mapping data to the solicit map-request message. Like in the former mechanism, this message is then sent to all ETRs of communication partners. If the receiving ITRs have an entry for that mapping in their mapping cache, they update this mapping with the piggybacked mapping data from the solicit map-request message. In addition, appropriate security mechanisms have to be taken into account to avoid malicious insertion of false mapping data in the caches of ITRs. This cache update mechanism requires additional signaling messages and by piggybacking the mapping information, it avoids the lookup for the piggybacked mapping at ITRs of communication partners. Hence, the timespan between handover event and updated mapping cache entries is reduced compared to the plain solicit map-request mechanism.

In case of asymmetric paths between the roaming mobile node and its communication partner, the former two mechanisms are not applicable. A scenario with asymmetric path is the communication with non-LISP domains via proxy-ITRs and proxy-ETRs. To solve this issue, each mobile node keeps track of proxy-ITRs which have recently sent packets to the mobile node in a separate proxy-ITR cache. The cache contains the locators of the proxy-ITRs. After a roaming event, the mobile node sends the solicit map-request messages also to these locators. Hence, also the mappings in the mapping caches of proxy-ITRs get updated.

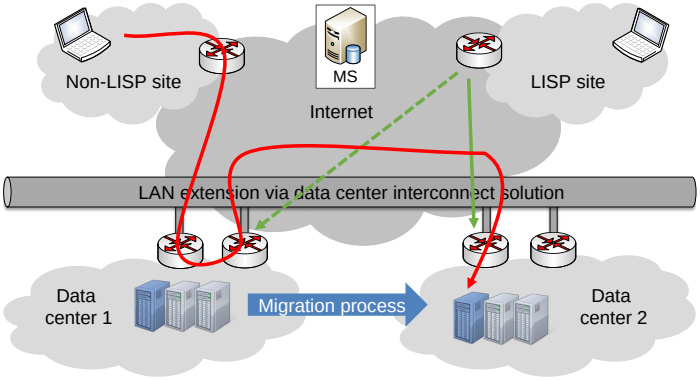


Figure 3.3: Ingress path optimization for extended subnets using LISP [90].

Ingress Path Optimization after VM Migration

In case of a multi-homed data center, two or more ingress paths are available for incoming traffic. By assigning a certain IP address, the data center provider can influence the ingress path for the traffic to a particular VM. Further, shortest path routing in the Internet or WAN ensures that incoming packets from customer networks take the shortest path to the destination VM. This path optimality however is not given anymore if VMs are migrated between data center locations that are transparently connected on Layer 2. In this case, the IP address of the migrated VM does not change so that existing transport connections survive the migration process. This way however, the traffic the to the migrated VM is still first forwarded to the old location because of the unchanged IP address. The corresponding data center edge switch however recognizes that the VM has moved and forwards the traffic to the new location. Overall, this results in triangle routing between the customer network, the old, and new data center location. With LISP-enabled data center and customer networks however, a suboptimal ingress path can be avoided while still maintaining transport connection survivability after a migration event. The migrated VM keeps its assigned EID and only the mapping

from EID to RLOC is changed in the mapping system. In addition, all involved LISP gateways are notified so that existing mapping cache entries are refreshed and existing connections use the new changed mapping. To show the difference between non-LISP and LISP networks, a scenario with two inter-connected LISP-enabled data center networks as well as non-LISP and LISP customer networks is shown in Figure 3.3. A VM is migrated from data center 1 to data center 2 while mobile nodes in the non-LISP and LISP site maintain active transport connections to the migrated VM. From the non-LISP site's perspective, the IP address of the migrated VM does not change and hence, the packets are still sent to data center 1 where they are forwarded to data center 2. The LISP site however notices the changed mapping and the corresponding LISP gateway directly sends the packets to the new location without the detour via the old location. Hence, LISP in combination with DCI solutions enables seamless service migration and optimal ingress path as long as both communication partners are LISP-enabled.

3.1.2 Mobility Challenges

In a scenario where LISP and LISP-MN are used to enable ingress path optimization to avoid triangle routing, several challenges are not addressed until now. These challenges are related to the mobility extension itself and must be solved so that seamless endpoint mobility is supported. In Figure 3.4, we sketch these challenges for a handover scenario between LISP and non-LISP network.

The overall scenario is a mobile node that receives a video stream from a video streaming server hosted in data center 1. At an arbitrary instance in time, the mobile node changes its access network and roams into a non-LISP network that is behind a NAT gateway. The mobile node then needs to inform its communication partners, i.e., the LISP-gateways at data center 1, about the roaming event. The LISP-gateways then query the mapping service to obtain the new mapping. Once the new mapping is available at the LISP-gateways, the video stream is sent to the new location of the mobile node and the handover is accomplished.

A first challenge in this scenario is related to the control and data plane of

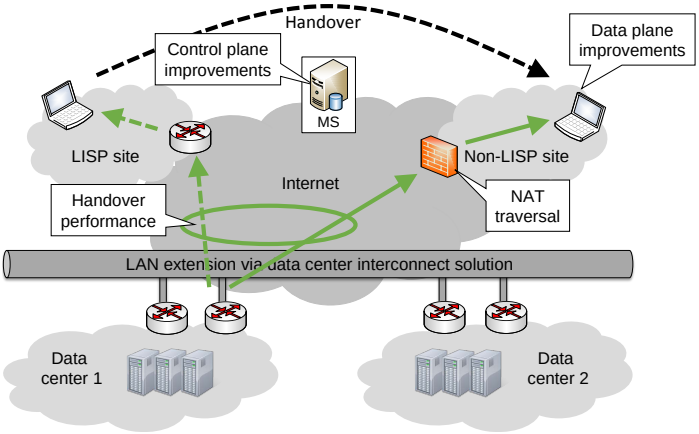


Figure 3.4: Addressed challenges related to LISP-MN.

LISP-MN. Depending on the specific communication scenario, LISP-MN causes increased communication delay as well as encapsulation overhead in comparison to the basic LISP architecture. We propose a set of improvements that reduce the communication delay and encapsulation overhead. Another important point is to restore connectivity behind the NAT gateway. The plain LISP-MN architecture does not comprise a mechanism to traverse NAT gateways and hence a second challenges is to enable NAT traversal. Finally, the LISP-MN architecture features several mechanisms to update communication partners about changed mapping information and depending on the used mechanism, the handover duration changes. So the last challenges is to evaluate the handover performance by comparing the different update mechanisms.

In the remainder, we present a detailed analysis of the forwarding structure of LISP-MN and highlight identified disadvantages that are addressed by our set of improvements. In addition, we explain in detail how NAT traversal is possible for a single LISP-gateway and why this is not practical for MNs.

Mobility Related Forwarding Analysis

The following forwarding analysis comprises various combinations of MNs and SNs located in either LISP or non-LISP networks. To that end, we consider MNs in LISP or non-LISP domains that communicate with MNs or non-LISP nodes in non-LISP domains or with SNs or MNs in the same or another LISP domain. In total, nine different scenarios are considered and the most complex ones are further illustrated with packet flow sequences.

1) A MN in a non-LISP domain communicates with another MN in a non-LISP domain: The MN addresses a packet towards the EID of the other MN, and encapsulates and sends the packet towards the globally routable RLOC for this EID. The same procedure applies for the reverse direction.

2) A MN in a non-LISP domain communicates with a SN in a LISP domain: The MN addresses a packet towards the EID of the SN and encapsulates the packet towards the RLOC for the SN's EID. The packet is forwarded to the ETR of the destination LISP domain where it is decapsulated and then forwarded to the SN (see Figure 3.5). In the reverse direction, the SN addresses a packet towards the EID of the MN and forwards it to a default ITR. The ITR encapsulates the packet towards the RLOC for the MN's EID and sends it to the MN.

3) A MN in a non-LISP domain communicates with a MN in a LISP domain: The MN in the non-LISP domain addresses a packet towards the EID of the MN in the LISP domain. It encapsulates the packet towards the LLOC of the other MN's EID and sends it. The packet is carried towards a PITR which encapsulates the packet again towards the RLOC of the ETR of the destination LISP domain and sends it. The packet is carried to that ETR which decapsulates the packet, and then forwards it to the MN in the LISP domain (see Figure 3.6). In the reverse direction, the MN in the LISP domain addresses the packet towards the EID of the MN in the non-LISP domain. It encapsulates it towards the other MN's RLOC and the packet is forwarded to a LISP gateway. After a mapping lookup which is negative, the LISP gateway tunnels the packet to its configured PETR to hide the LLOC as source address. The PETR decapsulates the packet and forwards it to

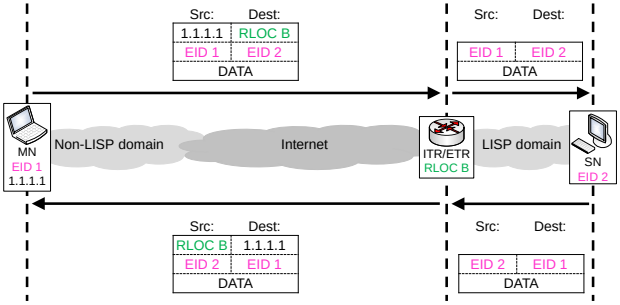


Figure 3.5: Scenario 2: A MN in a non-LISP domain communicates with a SN in a LISP domain.

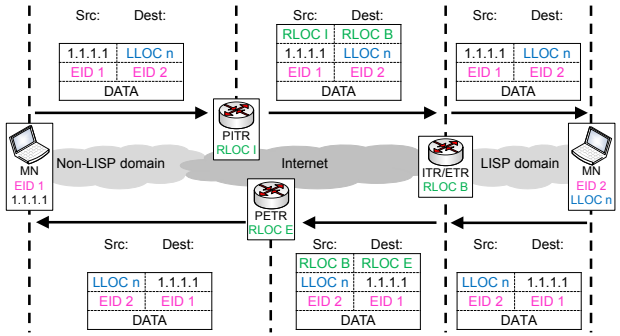


Figure 3.6: Scenario 3: A MN in a non-LISP domain communicates with a MN in a LISP domain.

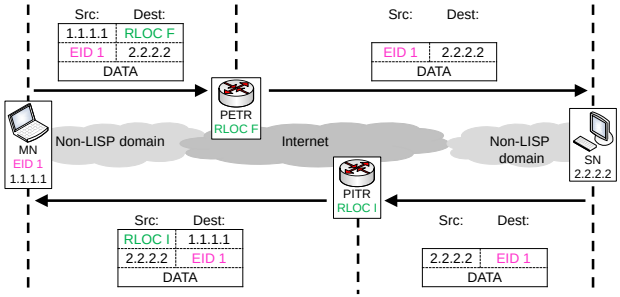


Figure 3.7: Scenario 4: A MN in a non-LISP domain communicates with a non-LISP node.

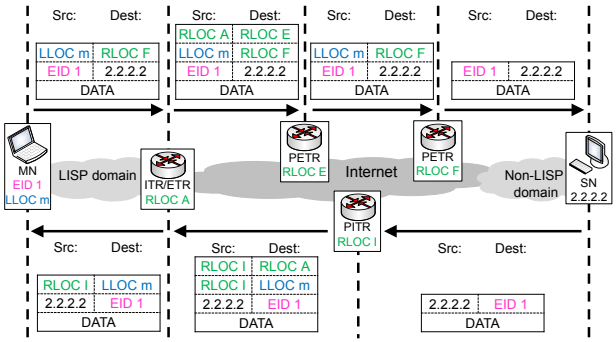


Figure 3.8: Scenario 5: A MN in a LISP domain communicates with a non-LISP node.

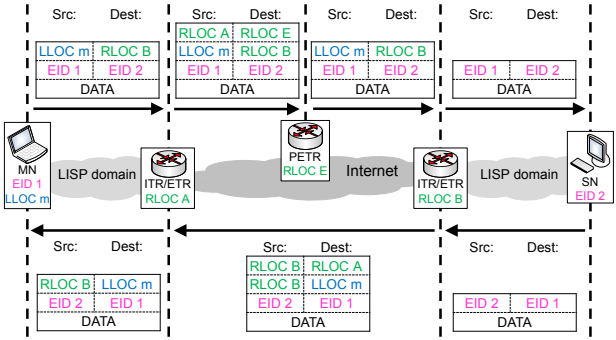


Figure 3.9: Scenario 6: A MN in a LISP domain communicates with a SN in another LISP domain.

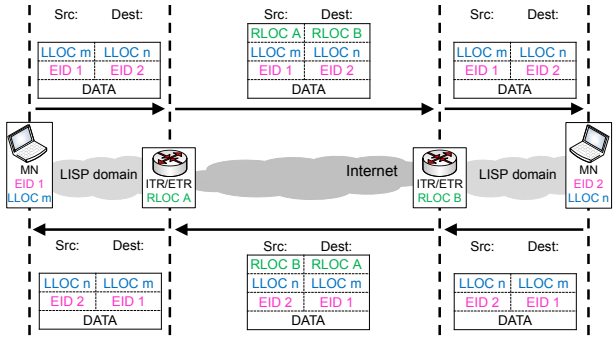


Figure 3.10: Scenario 7: A MN in a LISP domain communicates with a MN in another LISP domain.

the MN in the non-LISP domain. In both cases, triangle routing occurs due to the use of a PITR and a PETR.

4) A MN in a non-LISP domain communicates with a non-LISP node: The MN in the non-LISP domain addresses a packet towards the IP address of a non-LISP node. As there is no locator for that address, it encapsulates the packet towards the RLOC of its configured PETR and sends the packet. The PETR just strips off the outer header and the packet is forwarded to the non-LISP node (see Figure 3.7). In the reverse direction, the non-LISP node addresses a packet towards the EID of the MN and sends it. The packet is carried to a PITR. The PITR encapsulates it to the RLOC of the MN and sends it to that node. In both directions we observe triangle routing via the PETR or the PITR.

5) A MN in a LISP domain communicates with a non-LISP node: The MN in the LISP domain addresses a packet towards the IP address of a non-LISP node. As there is no locator for that address, it encapsulates the packet towards the RLOC F of its configured PETR and sends the packet. The ITR receives the packet and sees the LLOC in the source field. Therefore, it also encapsulates the packet towards the RLOC E of its configured PETR and sends it. The packet is first carried to PETR E which decapsulates it, then to PETR F which decapsulates it again, and eventually the non-encapsulated packet is carried to the non-LISP node (see Figure 3.8). In the reverse direction, the non-LISP node addresses a packet towards the EID of the MN. The packet is forwarded to a PITR. The PITR first encapsulates the packet towards the LLOC of the MN and then towards the RLOC for that LLOC. The packet is carried to the ETR which decapsulates it and passes it on to the MN in the LISP domain. In the forward direction we observe "quadrangle" routing via the PETRs of the ITR and the PETR of the MN while in the reverse directions we observe triangle routing via the PITR.

6) A MN in a LISP domain communicates with a SN in another LISP domain: The MN addresses a packet towards the SN's EID and encapsulates the packet to the SN's RLOC. The packet is sent to the ITR. The ITR sees an LLOC in the source field and tunnels the packet to its PETR. The PETR decapsulates the

packet and forwards it to the ETR of the destination domain. The ETR decapsulates the packet and it is forwarded to the SN (see Figure 3.9). In the reverse direction, the SN addresses a packet towards the EID of the MN and forwards it to its default ITR. The ITR first encapsulates the packet towards the LLOC for the MN's EID and then to the RLOC for this LLOC. The packet is carried to the ETR of the MN's domain, which strips off the outer encapsulation header and sends the packet to the MN.

7) A MN in a LISP domain communicates with a MN in another LISP domain: The MN addresses a packet towards the EID of the other MN and encapsulates it towards the LLOC of the other MN. The packet is carried to the ITR which queries the RLOC for the LLOC, encapsulates the packet accordingly, and sends it. The ETR decapsulates the packet and forwards it to the destination node (see Figure 3.10). The reverse direction works likewise.

8) A MN in a LISP domain communicates with a SN in the same LISP domain: The MN addresses a packet towards the SN's EID and encapsulates it towards the RLOC for that EID. The packet is sent to the ETR with that RLOC. This ETR decapsulates the packet and forwards it to the SN. If the SN is multihomed, it might have another ETR. The MN should ensure to choose the ETR that it has in common with the SN. In the reverse direction, the SN addresses a packet towards the MN's EID. The packet is forwarded to the default LISP gateway which first encapsulates it towards the LLOC of the MN and then to the RLOC for that LLOC. The packet is sent to the ETR with that RLOC which is possibly but not necessarily the same LISP gateway and the ETR decapsulates the outer decapsulation header. If the default LISP gateway is the same node as this ETR, the second encapsulation and the decapsulation can be omitted which is shown in the figure. Eventually the packet is forwarded to the MN. In the forward direction we observe triangle routing via the ETR of the LISP domain. In the reverse direction we witness either triangle routing via the default ITR or even "quadrangle" routing via the default ITR and the chosen ETR.

9) A MN in a LISP domain communicates with a MN in the same LISP domain: The MN addresses a packet towards the EID of the other MN in the same LISP

domain. It encapsulates the packet towards the LLOC for that EID and sends it. The packet is carried directly to the corresponding MN. The reverse direction works likewise.

Identified Forwarding Disadvantages

The above described scenarios present examples where the forwarding structure of LISP-MN is not optimal and leaves room for optimizations. Hence, in the following, we explicitly highlight the identified disadvantages and explain, which scenarios are affected by which disadvantage.

Unnecessary double mapping lookups: the LISP-MN extension requires double encapsulation by (P)ITRs when they receive traffic from SNs or non-LISP nodes towards MNs in LISP domains. Hence, two mapping lookups are needed (see return direction in scenarios 5 and 6). However, as the (P)ITR cannot know a priori whether a second lookup returns another RLOC, it must always perform two mapping lookups for all packets unless the first mapping lookup already returns a negative map-reply.

Avoidable detour via proxy gateways: under some conditions, PETRs and PITRs are needed as intermediate boxes for decapsulation and encapsulation (see scenarios 3 – 6). As PETRs and PITRs cause path stretch, i.e., increased communication delay due to triangle routing, their use should be avoided if possible. Typically, the preconfigured PETR of an ITR is close to the ITR but the preconfigured PETR of a MN is far away from the MN and the corresponding node. Therefore, avoiding the use of PETRs is especially attractive for communications where MNs are involved.

Unneeded detour via ETR in same domain: local communications between MNs and SNs in the same LISP domain requires detour via local ETR. This leads to triangle routing (see scenario 8) and causes path stretch. Although the absolute path stretch is probably negligible, relaying the traffic via the local ETR causes increased load at the ETR which should be avoided.

Unnecessary double encapsulation: traffic which is sent towards a MN residing in a LISP domain carries two encapsulation headers until it reaches the ETR of the destination LISP domain (see scenarios 3, 5 – 7). This may exceed the *Maximum Transfer Unit (*MTU) on one of the traversed links and leads to packet fragmentation. Therefore, it is also an undesired property.

NAT Traversal Challenge

Besides the challenges regarding the forwarding structure of LISP-MN, another important aspect is to enable MNs behind NAT gateways. MNs behave like lightweight LISP domains and implement LISP gateway functionality. The ETR functionality of LISP gateways is by definition addressable by globally routable RLOCs. This requires that the external address of the NAT device is registered in the mapping service, and that two static routes are installed in the NAT.

ITRs send LISP data messages over UDP/IP tunnels, and always use port *4341* as destination port and a random source port. Outgoing LISP data messages can be sent through the NAT but response packets are discarded at the NAT as the destination port does not match the previously used source port. Thus, it is necessary to install a static route in the NAT that forwards the external port *4341* to the internal port *4341* of the ETR. All incoming packets that arrive on port *4341* at the NAT are then statically forwarded to port *4341* at the ETR (see Figure 3.11). The static route enables the ETR behind a NAT to receive LISP data traffic. However, only one static route can be installed for a specific port at a NAT gateway and hence, only one ETR can be deployed behind a NAT by means of a static route. When ETRs behind a NAT also implement map server functionality, they must be able to receive map-requests, which arrive at port *4342*. Hence, this requires an additional static route (see Figure 3.11). Theoretically, static routes could support the deployment of an ETR behind a NAT, but it would be limited to a single ETR. This restriction might be acceptable for ITR/ETR gateways but not for MNs. Thus, an additional mechanism is required to support more than one MN behind a NAT.

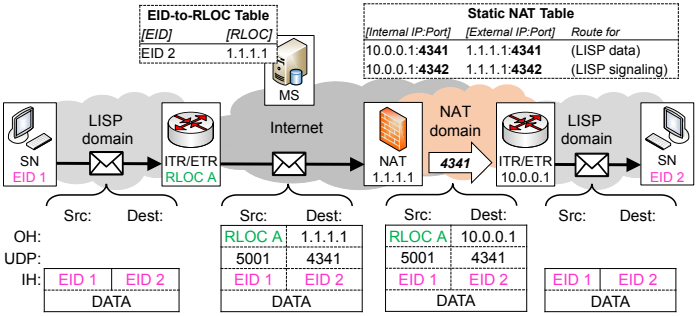


Figure 3.11: LISP gateway behind a NAT.

3.2 Related Work

In this section, we present a brief overview of other mechanisms that aim at making the rigid routing architecture more flexible. First, we review future Internet routing architectures and their mobility features and second, we compare the most promising mobility extension with dedicated mobility architectures.

3.2.1 Future Internet Routing

The current inter-domain naming and addressing architecture experiences scalability problems [35] and is not flexible enough to enable virtual machine migration between different locations of a data center. Hence, several new scalable routing architectures have been developed in the IRTF RRG working group which also support virtual machine migrations between subnets with the required flexibility. Most of the approaches are based on the *Locator/Identifier Split (*LOC/ID Split) [35] that helps to improve the routing scalability [102]. It uses special *identifiers (*IDs) to denote end-hosts or services. These IDs are not routable in the Internet core. Instead, a routable *Locator (*LOC) is added to packets to send them over the Internet. The current LOC for an ID is returned by a

Building block	GLI-Split	HIP	ILNP	Ivip	LISP
Status of Standardization	None	RFC ([91–94])	RFC ([95–98])	Draft ([99–101])	RFC ([38, 86, 89])
Core edge separation	Host- and network-based	Host-based	Host-based	Network-based	Network-based
Modification of host stack	Optional, required for mobility	Yes	Yes	No	Optional, required for mobility
Implementation of separation	Address rewriting	Extension header	Address rewriting	Tunneling	Tunneling
Mapping system	Local and global mapping system	DNS	DNS	Fast-push mapping distribution network	Interface for different mapping systems
Addressing format	IPv6 addresses	128-bit HIT and IPv6 addresses	IPv4 or IPv6 addresses	IPv4 or IPv6 addresses	IPv4 or IPv6 addresses
Interworking support	Intrinsic feature, no extra entities	Proxy gateways	Fallback to legacy mode	Proxy gateways	Proxy gateways, or NAT-based approach
Mobility requirements	Modified stack in MN and CN	Modified stacks, rendezvous server	Modified stack in MN and CN	Modified stack in MN and proxy gateways	Modified stack in MN and proxy gateways
Additional infrastructure	GLI gateways, mapping system	Modified DNS	Modified DNS	Ivip gateways, mapping system	LISP gateways, mapping system

Table 3.1: Comparison of future Internet Routing architectures and their most important building blocks.

special mapping service that stores an ID-to-LOC-mapping for each ID in the Internet. The exact implementation of the LOC/ID Split is dependent on the specific routing architecture and in Table 3.1, we classify the most prominent architectures according to selected building blocks.

The *Global Locator, Local Locator, and Identifier Split (*GLI-Split) architecture as well as the *Host Identity Protocol (*HIP) and the *Identifier Locator Network Protocol (*ILNP) implement a host-based LOC/ID Split where the translation between ID and LOC is done in the host's network stack. All three architectures are based on IPv6 but use different kind of mapping systems. HIP and ILNP store the mappings in a modified DNS whereas GLI-Split uses a hierarchical mapping system with a local and a global part. Mobility or virtual machine migration is supported by all three approaches and requires modification of the network stack inside mobile nodes. In addition to the host-based approach, the GLI-Split architecture also supports a network-based approach where site border routers perform the translation between ID and LOC. This variant improves backwards compatibility and does not require host network stack modifications. A pure network-based approach is implemented in the *Internet Vastly Improved Plumbing (*Ivip) architecture and by LISP. In both cases, modified site border gateways perform the translation between ID and LOC and use tunneling to add the required LOC information to outgoing packets. Mobility is also supported and requires host network stack updates and proxy gateways in the Internet core. A more detailed overview of the various routing architectures can be found in [103]

All presented approaches support mobility and virtual machine migrations but besides the future Internet routing architectures, there are other dedicated mobility support solutions available which should also be considered. Hence in the following, we also give a brief overview to these solutions and compare them with LISP-MN in Table 3.2. To provide a transparent comparison, we identified critical aspects like required changes to the existing architecture, additionally required infrastructure, or signaling as well as packet overhead. All approaches are then compared according to these aspects.

3.2.2 Mobility Support Architectures

*Mobility Support in IPv6 (*MIPv6) is a protocol extension to the current naming and addressing architecture that can be used to support mobile nodes in IPv6 networks. The mobility functionality is implemented in updated host network stacks and both nodes of an ongoing connection need to implement MIPv6 so that advanced mechanisms like route optimization can be used. While the MN is in a foreign network, a so called *Home Agents (*HAs) acts as anchor point on the data plane in the home network and tunnels packets to the MN. This however may lead to an increased communication delay as packets do not take the shortest path between MN and *Correspondent Node (*CN). To avoid this, the so-called route optimization mechanism can be used to send packets directly between MN and CN. This however requires that both nodes are MIPv6 capable. There are various extensions available to MIPv6 which are designed for specific use cases like fast handovers with reduced packet loss [107] or a hierarchical approach that reduces the amount of signaling messages [108]. In addition, network-based approaches called *Proxy-MIPv6 (*PMIPv6) [106] and *Network Mobility (*NEMO) [105] are introduced that require no host updates and implement the mobility functionality inside network elements. NEMO introduces mobile routers to provide mobility to a whole network of mobile devices without requiring host network stack updates. PMIPv6 in contrast provides mobility to nodes within a local network and only allows roaming events between subnets of the same mobility domain. A more detailed overview of mobility support architectures can be found in [109].

One major disadvantage of MIPv6 and its various extensions is that they require a detour on the data plane via a fixed entity, i.e. the HA. This detour is also known as triangular routing and increases the communication delay between MN and CN as packets do not take the shortest path between MN and CN. In contrast to MIPv6, LISP-MN was designed to avoid triangular routing on the data plane for most of the communication scenarios. In addition, LISP-MN fits well into the LISP future Internet routing architecture and reuses many of its components for the mobility support. An implementation of LISP-MN for Linux, Android, and

Building block	LISP-MN	MIPv6	NEMO	PMIPv6
Status of Standardization	Draft ([39])	RFC ([104])	RFC ([105])	RFC ([106])
Required infrastructure	Proxy gateways	Home agent	Mobile router, home agent	Access gateway, mobility anchor
Mobile node modification	Network layer	Network layer	Not required	Not required
Correspondent node modification	Not required	Required for route optimization	Not required	Not required
Detour via intermediate node	Required for interoperability	Required until route optimization	Permanent detour via home agent	Only intra-domain roaming events
Data plane packet overhead	Extra IP, UDP, and LISP header	IP tunnel (MN-HA) Ext. header (MN-CN)	IP tunnel between HA & mobile router	IP tunnel (MAG-LMA)
Signaling after roaming event	Mapping update, Cache update	Binding update, CN registration	Binding update	Binding update
Rendezvous infrastructure	Map server	Home agent	Home agent	Mobility anchor

Table 3.2: Comparison of future Internet mobility architectures.

OpenWRT is also available [110]. Hence, LISP with its mobility extension LISP-MN is the most promising approach to provide scalable routing and mobility in a future Internet architecture. Hence, the overall objective of this chapter is to address open issues that arise when using LISP as enabling protocol for endpoint and service mobility.

3.3 Improvements to LISP Mobile Node

Earlier in this chapter, we described how LISP enables endpoint and service mobility and what are open challenges in such a scenario. Hereinafter, we propose a set of improvements to the LISP-MN extension with respect to the control and data plane. In addition, we present a mechanism that allows NAT traversal for MNs behind NAT gateways.

3.3.1 Control Plane Improvements

An important aspect for LOC/ID-Split based architectures like LISP is the indirection between EID and RLOC and the required lookup in the mapping system to get the corresponding RLOC for an EID. Subsequently, we present two improvements with respect to the information stored in the mapping system and to the mapping lookup process.

Locator Types

In Section 3.1.1, we introduced LLOCs and RLOCs as two different locator types for the sake of simpler readability, but LISP-MN does not take advantage of this differentiation. We now suggest that the locator type is stored as accompanying information together with the mapping entries in the mapping system, returned in map-replies, and stored in map-caches.

The globally reachable IP addresses of LISP gateways are registered in the mapping system as RLOCs for EIDs and care-of-addresses inside that LISP domain. When a MN roams into a new network, it obtains a new care-of-address

(e.g., by DHCP) under which it is then reachable. It registers this address in the mapping system as a locator for its EID. We propose that it also stores the locator type as "LLOC" in the mapping system. Then, it queries the mapping system with that address. If a negative map-reply is returned, the MN is in a non-LISP domain and the care-of-address is an RLOC. Therefore, the MN changes the locator type for its EID-to-locator mapping in the mapping system to "RLOC". If RLOCs are returned for the requested care-of-address, the MN is in a LISP domain and the care-of-address is in fact an LLOC so that nothing needs to be changed.

ITRs, PITRs, and MNs take advantage of the locator type information in the mappings. If they encapsulate packets towards RLOCs, they can send them immediately without querying the mapping system again. This avoids unnecessary double mapping lookups by ITRs, PITRs, and MNs when the destination address of a packet is not a MN in a LISP domain.

A MN-bit was proposed in [39, Sect. 8] in the context of multicast. It indicates in the mapping whether the node for which the locator is returned is a MN. This is similar to the locator type but not the same because MNs in non-LISP domains do not have LLOCs. To save extra bits in the mappings, the MN-bit may be used instead of the locator type. This avoids double lookups by ITRs, PITRs, and MNs when the destination of a packet is a SN, but if the destination is a MN in a non-LISP domain, the second unnecessary lookup cannot be avoided.

Local Mapping System

In addition to the locator type bit, we propose a local mapping system that helps a MN to send traffic to a SN in the same LISP domain without triangle routing over the SN's ETR. Each LISP gateway knows the routable EIDs of all SNs in its domain, e.g., by configuration. Furthermore, it keeps a local EID-to-LLOC table for all MNs in its domain. It returns these mappings when queried by registered MNs. This constitutes the local mapping system. We explain how the EID-to-LLOC table is populated. If a MN roams into a LISP domain, it receives an LLOC, queries the global mapping system with that LLOC, and records the

returned RLOCs as configured LISP gateways. Then, it registers its EID together with its obtained LLOC at all configured LISP gateways. The LISP gateways use soft state to store this information in their EID-to-LLOC tables so that stale information is purged after short time when MNs have left the LISP domain without logging off properly.

When a MN in a LISP domain wants to send an outgoing packet, it first queries one of its configured LISP gateways with the destination address of the packet for an EID-to-LLOC mapping. The gateway returns one of the following three responses: (1) the EID belongs to a SN in the same domain, (2) the EID belongs to a MN in the same domain and the LLOC is delivered, or (3) a node with the requested destination address is not in the same domain. In the first case, the MN sends the packet without encapsulation. In the second case, the MN encapsulates the packet towards that LLOC and sends it. In the third case, the MN queries the global mapping system with the destination address of the packet, encapsulates the packet if the map-reply was positive, and sends the packet. In this third case, the lookup latency of first packets is increased by the query to the LISP gateway. However, this adds only little delay because the LISP gateway is near and extra delay is not very critical in this particular situation as the packet waits at its source node. As a result of this optimization, the MN sends traffic directly to another SN in the same domain and triangle routing via the SN's ETR is avoided.

Also the behavior of ITRs should be changed. Before querying the global mapping system, they should query their local mapping system when a packet causing a cache miss is to be sent. If the map-reply from the local mapping system is positive, the packet is encapsulated with the returned LLOC and sent. This avoids potential quadrangle routing when a SN sends traffic to a MN in the same LISP domain (see scenario 8 in Section 3.1.2).

3.3.2 Data Plane Improvements

Besides the described modifications on the control plane, we also present a set of improvements related to the LISP data plane. These improvements reduce the

packet overhead by omitting additional LISP headers and reduce the communication delay by avoiding the proxy gateways when communicating with non-LISP networks.

Filter Check and Direct Communication

A MN tunnels packets to its PETR to hide non-routable source addresses from the provider edge router which performs source address filtering. A recent study observed that in 31% of the investigated scenarios, ISPs did not block spoofed source addresses [111]. In such networks, tunneling traffic to the PETR is not necessary. This provides some optimization potential to reduce path stretch and latency.

We propose that when a MN roams into a new network, it should find out whether outgoing packets sent with its EID as source address are blocked by the ISP. To discover this, the MN may ping the RLOC of its configured PETR without encapsulating packets. If the PETR responds, the provider edge router does not filter packets having the EID of the MN as source address. Therefore, the MN can communicate directly with non-LISP nodes. Hence, when the MN later queries the mapping system with the destination address of an outgoing packet and the map-reply is negative, the MN can send the packet directly without tunneling it to its PETR. This improvement can reduce the path stretch for scenario 4 and scenario 5. However, this mechanism is only effective if the ISP of the network hosting the MN does not perform source address filtering.

Location-Aware MNs

The basic idea in the remainder is that the MN should find out whether it is currently hosted in a LISP domain or in a non-LISP domain. To detect the difference, the MN queries the mapping system with its assigned care-of-address. If a negative map-reply is returned, the MN is in a non-LISP domain and the care-of-address is an RLOC; otherwise, when an RLOC is returned, the MN is in a LISP domain, and the care-of-address is an LLOC. We call a MN having that in-

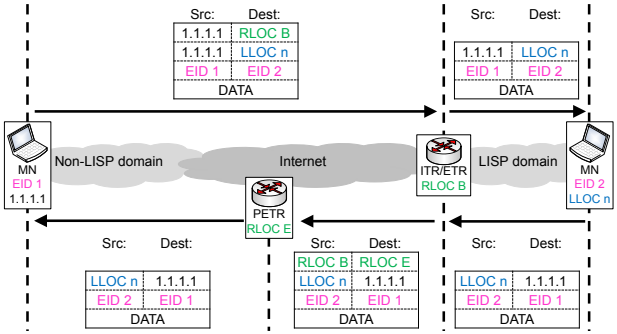


Figure 3.12: Scenario 3: MNs in non-LISP domains perform double mapping lookups and double encapsulation if needed.

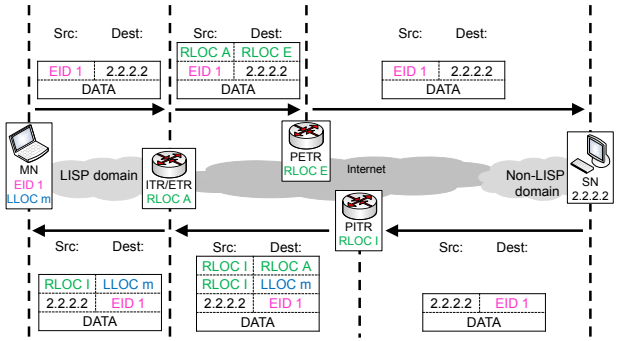


Figure 3.13: Scenario 5: MNs in LISP domains send packets to non-LISP nodes without encapsulation.

formation location-aware. A location-aware MN in a non-LISP domain can avert that its sent traffic is routed via a PITR if it communicates with a MN in a LISP domain (see scenario 3 in Figure 3.6). To that end, the MN always performs a double mapping lookup and double encapsulation if possible just like an ITR or PITR. Figure 3.12 shows that traffic is then carried directly from the MN in the non-LISP domain to the ETR of the destination LISP domain so that triangle routing via a PITR is prevented. Furthermore, a location-aware MN in a LISP domain can avoid that its sent traffic is routed over the PETR of the MN if it communicates with a non-LISP node (see scenario 5 in Figure 3.8) although the

ISP of the LISP domain performs source address filtering. To that end, the MN sends packets directly instead of encapsulating them to its PETR when the mapping lookup for the destination address of a packet returns a negative map-reply. Figure 3.13 shows that the ITR then tunnels the traffic to its PETR which then forwards it directly to the non-LISP destination node. This averts triangle routing via the PETR of the MN, which would otherwise most likely add a significant path stretch. The remaining path stretch through triangle routing over the PETR is likely to be small because the involved PETR is located near the ITR.

Avoiding Double Encapsulation Headers

Packets addressed to MNs in LISP domains have two encapsulation headers. The outer encapsulation header carries the destination RLOC which is used for forwarding in the Internet except for the destination domain. The inner encapsulation header carries the destination LLOC which is used for forwarding in the destination domain. We propose a mechanism to avoid this double encapsulation. In the Internet except for the destination domain, packets addressed to MNs in LISP domains are encapsulated with the destination RLOC in the destination field. In the destination domain, they are encapsulated with the destination LLOC in the destination field. Our proposal prerequisites the local mapping system presented in Section 3.3.1.

We change the behavior of the MN slightly. If it queries the global mapping system and receives an LLOC, it queries the global mapping system again with the obtained LLOC to get its RLOC. Then it encapsulates the packet only to the RLOC. When an ITR receives a packet, it performs the modified steps according to the behavior described in Algorithm 1. PITRs work in the same way but cannot consult a local mapping system and check for registered LLOCs. When an ETR receives a packet destined to itself, it decapsulates it. Then, it queries the local mapping system for an LLOC of the destination address. If an LLOC is returned, the ETR encapsulates the packet towards that LLOC. Eventually, the ETR forwards the packet into its LISP domain.

Algorithm 1: Modified ITR forwarding behavior.

```
   input : packet with srcAddress and destAddress
1  Loc1 ← queryLocalMS(destAddress)
2  if Loc1 ≠ Null then
3      # Loc1 is LLOC
4      encapsulate packet to Loc1, send it
5  else
6      Loc2 ← queryGlobalMS(destAddress)
7      if Loc2 == Null then
8          # negative map-reply ⇒ destAddress is routable
9          if srcAddress is LLOC in ITR's domain then
10             # see reverse direction in Fig. 3.14, forward direction in
11             # Fig. 3.16, and Fig. 3.17
12             substitute srcAddress with ITR's RLOC, send packet
13         else
14             # srcAddress is EID, see Fig. 3.15
15             encapsulate packet to PETR, send it
16     else if typeof(Loc2) == LLOC then
17         # see reverse direction in Fig. 3.16
18         Loc3 ← queryGlobalMS(Loc2)
19         encapsulate packet to Loc3, send it
20     else
21         # Loc2 is already RLOC, see Fig. 3.1
22         encapsulate packet to Loc2, send it
```

These changes have no impact on the encapsulation and forwarding structure in scenarios 1, 2, 4, 8, and 9 which do not suffer from double encapsulation headers. Scenarios 3, 5, 6, and 7 still suffer from double encapsulation either in the forward or reverse direction even if the changes of the previous sections are applied. Figures 3.14–3.17 show the encapsulation and forwarding structure with the proposed changes of this section. We compare the structure with and without our new mechanism and explain how double encapsulation is prevented.

We consider scenario 3 and compare Figure 3.14 and Figure 3.12. On the forward path, there is no double encapsulation as only the encapsulation header with

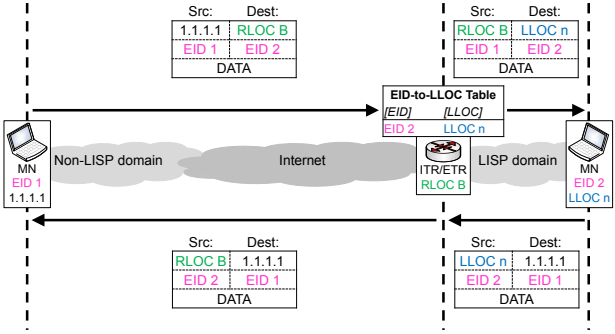


Figure 3.14: Scenario 3: A MN in a non-LISP domain communicates with a MN in a LISP domain.

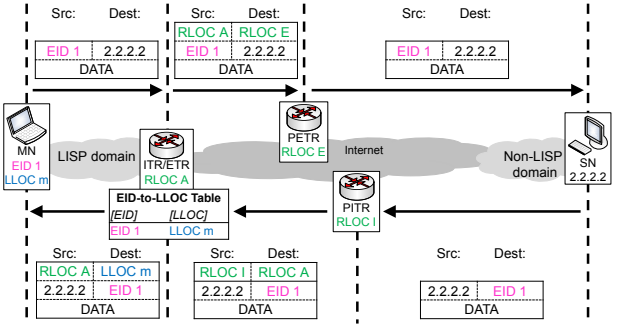


Figure 3.15: Scenario 5: A MN in a LISP domain communicates with a non-LISP node.

the destination LLOC is added at the ETR. On the reverse path, an additional encapsulation header is averted because the ITR substitutes the source LLOC by its own RLOC and avoids thereby tunneling packets to its PETR. We consider scenario 5 and compare Figure 3.15 and Figure 3.13. On the reverse path, double encapsulation is avoided by adding the encapsulation header with the destination LLOC only at the ETR instead at the PITR. We consider scenario 6 and compare Figure 3.16 and Figure 3.9. On the forward path, an additional encapsulation header is not necessary since the ITR substitutes the source LLOC by its own RLOC and avoids thereby tunneling packets to its PETR. On the reverse

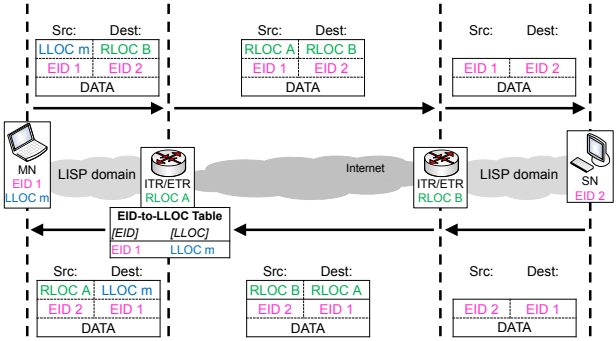


Figure 3.16: Scenario 6: A MN in a LISP domain communicates with a SN in another LISP domain.

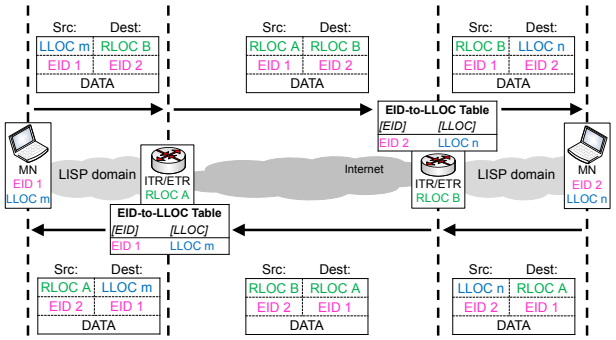


Figure 3.17: Scenario 7: A MN in a LISP domain communicates with a MN in another LISP domain.

path, double encapsulation is averted by adding the encapsulation header with the destination LLOC only at the ETR instead at ITR. We consider scenario 7 and compare Figure 3.17 and Figure 3.10. On the forward and on the reverse paths, an additional encapsulation header is avoided because the ITR substitutes the source LLOC by its own RLOC and avoids thereby tunneling packets towards its PETR. Moreover, double encapsulation is avoided by adding the encapsulation header with the destination LLOC only at the ETR instead at the ITR.

We have shown that double encapsulation can be avoided for communication with MNs. In addition, the use of PETRs is minimized and thereby path stretch

is reduced. However, our proposal has some disadvantages. When ITRs change the source address of a packet, this corresponds to combined decapsulation and encapsulation which might be difficult to implement. ETRs need to add encapsulation headers whereas without our additions they just used to decapsulate traffic. Whenever the global mapping system is queried and returns an LLOC, the LLOC is no longer added as destination address to packets which seems inefficient. This unnecessary indirection in the mapping system may be avoided by registering only RLOCs for both SNs and MNs.

3.3.3 NAT Traversal Mechanism

When mobile nodes are behind a NAT and implement only the standard LISP-MN architecture, they can send traffic to other nodes, but cannot receive traffic from them. When roaming into the NAT domain, they receive a private care-of-address, and register it at their associated map server. When sending traffic, the MN queries the mapping system on destination port *4342* without LISP encapsulation so that the map-reply is able to return to the MN. The MN encapsulates data packets towards the obtained RLOC using the care-of-address as source address. The address is modified in the headers of outgoing packets when crossing the NAT. When an ITR tries to send a packet to a MN behind a NAT, the mapping system returns a private address as RLOC so that packets cannot be forwarded correctly after encapsulation and get dropped. Thus, the packets never reach the MN behind the NAT.

The NAT traversal functionality is collocated with the same box that also implements the map server and the PETR for the MN. In the following, we call a modified map server that implements the NAT traversal mechanism a *NAT Traversal Router (*NTR). When a MN roams into a network, it obtains a care-of-address and registers it as RLOC for its EID at its preconfigured NTR. If the NTR recognizes that the MN is behind a NAT, the IP address of the NTR is registered as RLOC for the EID of the MN in the mapping system. Thus, when traffic is sent to MNs behind a NAT, (P)ITRs tunnel it to NTRs instead of to the care-of-

address of the MNs. The NTR has sufficient information to relay that traffic to the MNs and the traffic traverses the NAT due to the context established during the registration. This essentially constitutes a tunnel between the NTR and the MN which is used to bypass the NAT gateway. Due to the tunnel between the NTR and the MN, our NAT traversal mechanism works with every type of NAT, even with symmetric NATs, and is able to cope with several layers of NAT gateways.

Subsequently, we explain how MNs behind a NAT register at NTRs, and how NTRs relay packets destined to registered MNs. We then describe the applicability of our NAT traversal mechanism to stationary LISP domains behind a NAT. Finally, we discuss some deployment considerations and security concerns.

Registration Process

When a MN roams into a network, it receives a care-of-address from the local DHCP service and sends a map-register message to the map server using destination port *4342* without any LISP encapsulation. In contrast to the current behavior in LISP-MN, our NAT traversal proposal requires that source port *4341* is used (the reason is explained later in this section). The collocated NTR compares the reported care-of-address with the source address of the register message. If they are the same, the MN is not behind a NAT and the address is registered as RLOC for the EID of the MN in the mapping system. If the two addresses differ, the newly proposed NAT traversal concept for MNs behind NATs is used. We explain it using the packet flow sequence in Figure 3.18. A MN with EID 1 has roamed into a private network and obtained the care-of-address 10.0.0.1. It sends a register message containing this address to port *4342* at the NTR with RLOC N. The intermediate NAT gateway translates the source IP:port 10.0.0.1:*4341* into 203.0.113.3:*11341* and stores this as context for outgoing packets with destination IP:port RLOC N:*4342*. The NTR detects that the care-of-address 10.0.0.1 differs from the source address of the register message (203.0.113.3) and, therefore, it stores its own IP address (RLOC N) as RLOC for EID 1 in the mapping system. In addition, the NTR records the source address and port of the register

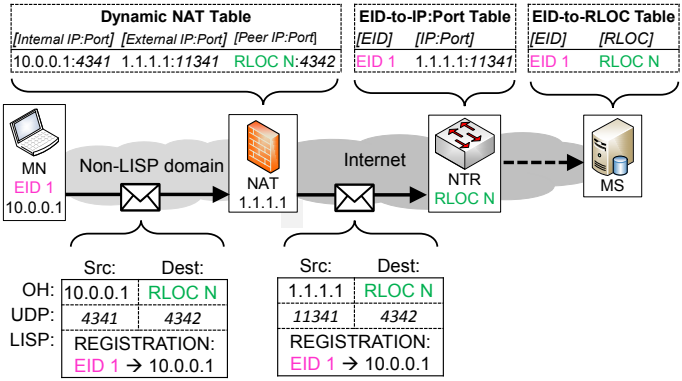


Figure 3.18: Registration process.

message (203.0.113.3:*11341*) with the EID (EID 1) in an EID-to-IP:port table. The NTR requires this IP:port to relay packets to the MN behind the NAT. The private address (10.0.0.1) is not stored at the NTR and only used by the NTR to detect whether the source address of the register message differs from the registered care-of-address. To make the mapping system robust against stale information, an expiration timer is associated with registered EID-to-RLOC mappings. The same may be applied to the EID-to-IP:port table inside the NTR. However, in this context, the expiration timer should be set to small value so that the established context in the NAT gateway is also refreshed in time.

Relaying Process

When traffic is sent to MNs behind a NAT, (P)ITRs tunnel it to the NTR at which the MNs have registered. This is depicted in Figure 3.19. An NTR relays such traffic as follows. It strips off the LISP and UDP header, uses the destination EID (EID 1) in the IH of the packet to look up the IP:port 203.0.113.3:*11341* in the EID-to-IP:port table, and encapsulates the packets to this IP:port combination using its own IP address and port *4342* as source IP:port combination (RLOC

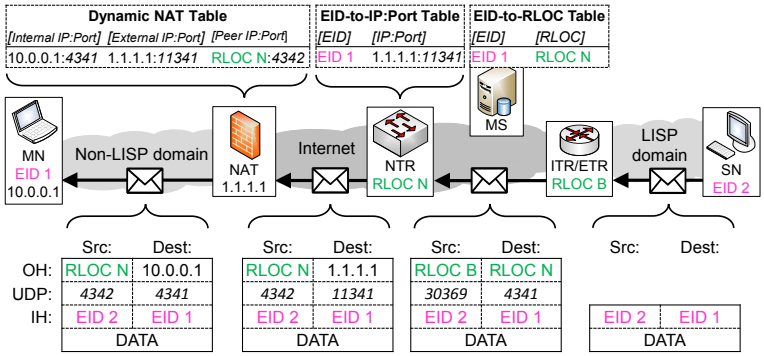


Figure 3.19: Incoming flow for a MN behind a NAT.

N:*4342*). The NAT gateway recognizes the destination IP:port and translates it accordingly which is 10.0.0.1:*4341* in our example. Eventually, the translated packet reaches the MN on the correct port *4341* for incoming LISP-encapsulated traffic. The correct port number is achieved by requiring MNs to send map-register messages to the map server using source port *4341*. Regarding the behavior of a MN, this constitutes the only difference between our proposal and the original LISP-MN architecture. Choosing another source port for the registration process would require that the MN has to listen on that port for LISP data traffic in case it is behind a NAT. By using the LISP data port *4341*, we avoid this issue and the MN has not to be aware of the NAT.

Applicability to Stationary LISP Domains

A LISP gateway behind a NAT can be made reachable from the outside by a static route from the NAT gateway to the LISP gateway (see Section 3.1.2). However, this works only for a single LISP gateway per NAT gateway. A slight adaptation of our proposed NAT traversal mechanisms allows to operate a large number of LISP gateways behind a NAT which might be a significant advantage. The EID ranges for stationary LISP nodes are configured with the LISP gateway. The LISP

gateway registers all configured EID ranges with the NTR and the NTR registers its own RLOC in the map server for these EID ranges. As a consequence, the stationary LISP nodes in the LISP domain behind the NAT are reachable from the outside through the NTR which forwards incoming traffic to the respective LISP gateway. For LISP domains receiving high data rates, care must be taken since all incoming traffic is relayed over the NTR. The number of supportable users in these LISP domains is limited only by the number of simultaneous outgoing connections that can be supported by the NAT device.

Deployment Considerations

In the description of our NAT traversal mechanism, we assumed that the NTR is collocated with the map server of the MN. However, it is also possible to run the NTR functionality in a separate box. This maybe important when the provider of the map server does not want to relay high data rates. The NTR in this case relays signaling traffic between the MN and its map server and data traffic between communication partners of the MN and the MN itself. The NTR infrastructure is then completely decoupled from the map server infrastructure.

Security Concerns

The presented NAT traversal allows nodes in the Internet to contact MNs behind a NAT gateway which is the intention of the proposal. If the NAT is used as part of a firewall, external nodes can easily circumvent this security feature and contact MNs. This is a general concern of all NAT traversal mechanisms. Moreover, any type of traffic can reach the MN behind a NAT/firewall because of tunneling. This may be improved by making the NAT/firewall aware of this mechanism using deep packet inspection for incoming LISP traffic.

Modified NAT Traversal

The proposed NAT traversal mechanism adds new functionality to map servers and does not introduce additional complexity at MNs. While this may be impor-

tant for low power nodes, a more sophisticated version of a NAT traversal mechanism may be interesting for MNs with sufficient resources. The mechanism described in the following requires updates to the MN stack but avoids sending registration messages that carry a private IP address. This avoids a possible problem with NAT gateways that utilize an *Application Layer Gateway (*ALG) to modify private IP addresses inside the payload of outgoing packets. Under these conditions, the previously described NTR would not be able to determine whether the MN is behind a NAT.

In the modified mechanism, an upgraded MN first sends a message to its preconfigured NTR and asks for its external address. The NTR responds with the address seen in the source address field of the message from the upgraded MN. The MN receives the response message and compares the returned external address with its own care-of-address. If the external address matches its care-of-address, the MN infers that it is not behind a NAT and contacts its preconfigured map server to register its care-of-address as current locator. Otherwise, both addresses differ and the MN concludes that it is behind a NAT and uses the proposed NAT traversal. This improved version of the NAT traversal mechanism avoids problems with NAT devices that use an ALG to modify private addresses inside the packet payload but requires changes to the MN stack. However, the implementations of the MN stack are not yet deployed and hence the proposed modification may be an interesting tradeoff between complexity and increased robustness of our proposed NAT traversal mechanism.

3.4 Evaluation of LISP Performance

Subsequently, we assess the efficiency of our proposed improvements to the LISP-MN architecture and investigate the LISP-MN handover performance for different handover scenarios. We start with a short introduction to the implemented and used LISP simulation framework.

3.4.1 LISP Simulation Framework

The simulation model of the LISP architecture and its extensions was implemented in the OMNeT++ simulation framework. OMNeT++ [73] is a discrete event simulation environment with a modular, component-based architecture. We built our model on top of the INET framework [112] which extends OMNeT++ with Internet-related protocol implementations and several application models.

Initially, our implementation was based on design ideas of OpenLISP. OpenLISP [113] is an open-source implementation of the LISP protocol running in the kernel of the FreeBSD Operating System. However, the OpenLISP implementation does not cover all extensions of the LISP architecture which are described in the LISP standard documents [8, 38, 39, 86, 89]. Therefore, we implemented our LISP model mainly based on the standard documents. Starting with the INET framework, we incrementally integrated the basic LISP architecture, map server interface, interworking, mobility, and eventually NAT traversal.

The *LISPRouter* node is our model of a LISP gateway and the central part of the LISP architecture. In principle, the *LISPRouter* node acts as ordinary router but with extended functionality as it implements the LISP forwarding behavior. To simplify the implementation, the *LISPRouter* node may both act as ITR and ETR. This way, it is more convenient to build either single-homed or multi-homed domains with several LISP gateways. Depending on the desired scenario, the *LISPRouter* nodes may then be configured as ITR, ETR, or both. Figure 3.20 shows a sketch of the architecture of a *LISPRouter* node. Boxes with dashed border represent unchanged INET modules, boxes with bold border changed INET modules, and boxes with regular border and gray-colored background new modules. The important new modules of a *LISPRouter* node are the *MapResolver* module, the *LISPRouting* module, and the *MappingCache* module. The communication between the *LISPRouting* module and the *MapResolver* module is done through INET's *NotificationBoard*. While the *LISPRouting* module is mainly responsible to encapsulate and decapsulate the LISP data traffic, the *MapResolver* module is responsible to send and receive LISP control traffic, e.g. performing the

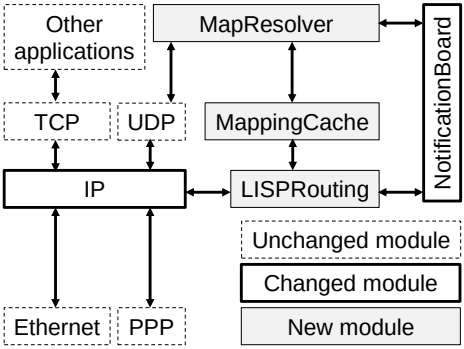


Figure 3.20: Architecture of a LISP gateway.

registration process or the mapping lookup. We modeled the *MapResolver* module as a UDP application because the LISP signaling traffic is sent over UDP.

The *MappingCache* module in between the *MapResolver* module and the *LISPRouting* module is used to cache the returned mappings after a mapping lookup. During a mapping lookup, the *MappingCache* module may either store the pending packets or drop them. After a mapping entry inside the *MappingCache* module has been used, its timer may either be refreshed or not. The last option may for instance be important for mobile nodes where the mapping regularly changes. The different options are configuration parameters and hence enable an easy setup of scenarios with different *LISPRouter* behavior.

To include the LISP mobile node architecture in our simulation model, we extended the *LISPRouting* module. Further, the *MapResolver* module was made aware of care-of-address changes. The architecture of a mobile node can be seen in Figure 3.21. After a roaming event, the *DHCPClient* module first starts the DHCP discovery process once the wireless interface is associated with an access point. Once the DHCP process is finished and the wireless interface got a new care-of-address, the registration process is started by the *MapResolver* module. Again, this event is signaled between the *InterfaceTable* and the *MapRe-*

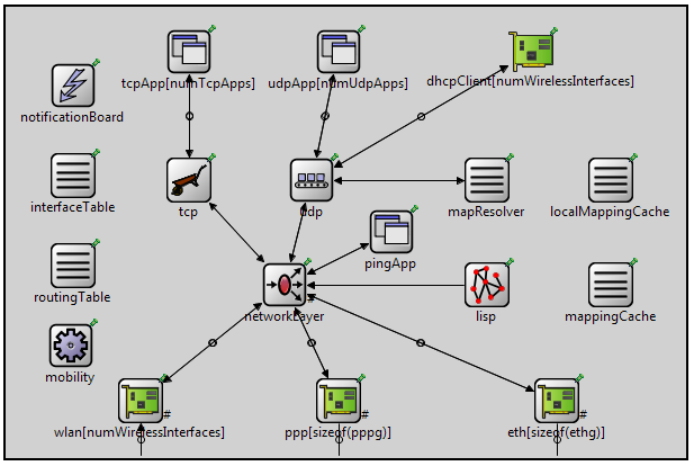


Figure 3.21: MN host stack in OMNeT++/INET.

solver via the *NotificationBoard*. During the registration process, the *MapResolver* sends a map-register message to the configured *MapServer* node. The *MapServer* acknowledges the map-register message with a map-notify message. Once the *MapResolver* receives this message, it starts the configured update process of remote caches of communication partners. This completes the handover process. The also visible *localMappingCache* module is part of the proposed local mapping service and stores the mapping for nodes within the local domain.

3.4.2 Efficiency of Local Mapping Service

We have already shown qualitatively how the path stretch and packet overhead due to additional LISP headers can be reduced by applying our proposed improvements to the LISP-MN architecture. These modifications either avoid the detour via proxy LISP gateways in the network or omit unnecessary additional headers. The efficiency of the local mapping service however has not been shown

so far and in the following, we quantify in which scenarios the local mapping services reduces the mapping lookup and which scenarios do not benefit from a local mapping service.

As performance metric, we consider the signaling delay that is caused due to the mapping lookup performed by the MN. In the evaluated scenario, a MN is located in a LISP domain with local mapping service and opens connections to internal and external nodes. All internal nodes are registered in the local mapping service and the mappings for external nodes are only available via the global mapping service. The probability to contact an external node is denoted with p_{ext} in the remainder. The mean lookup times in the local and global mapping service are denoted with d_{lms} and d_{gms} respectively. Hence, the mean lookup time $E[l_{unmod}]$ for the unmodified LISP-MN architecture without usage of the local mapping service can be written as

$$E[l_{unmod}] = d_{gms}. \tag{3.1}$$

For the modified LISP-MN architecture with local mapping service, the mean lookup time $E[l_{mod}]$ is

$$E[l_{mod}] = d_{lms} + p_{ext} \cdot d_{gms}. \tag{3.2}$$

To make the results of our study independent of assumptions of d_{lms} and d_{gms}, we normalize all performance metrics by d_{lms} and vary the ratio $r_{ms} = d_{gms}/d_{lms}$ in the range [5, 20]. Further, we investigate the influence of the probability to contact an external node p_{ext} in the range [0, 1]. Finally, we calculate the proportion P_{lms} in percent according to Equation 3.3 to show the improvement due to our proposed local mapping service.

$$P_{lms}[\%] = \frac{E[l_{mod}]}{E[l_{unmod}]}[\%] = \frac{1 + p_{ext} \cdot r_{ms}}{r_{ms}}[\cdot 100\%]. \tag{3.3}$$

In Figure 3.22, we present the results as contour plot to show the scenarios that benefit from the local mapping service. The contour plot shows the isolines for

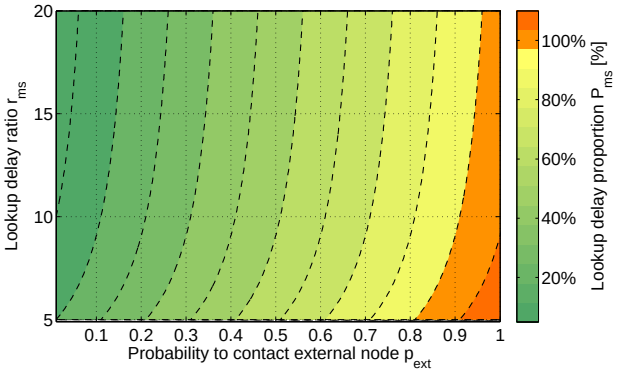


Figure 3.22: Contour plot showing improvement due to local mapping service.

P_{lms} and the corresponding isoline values are shown in the colorbar on the right-hand side. For scenarios where the value is higher than 100%, our proposed local mapping service does not reduce the mapping lookup time but increases it. This is true for scenarios where the probability p_{ext} is higher than 0.96. In this parameter range, the lookup delay ratio r_{ms} does not matter and P_{lms} is always higher than 100%. For lower values of p_{ext} however, the influence of r_{ms} increases and the value of P_{lms} drops below 100% which means that the proposed local mapping service leads to an reduction of the mapping lookup delay. For values of p_{ext} smaller than 0.8 then, there is always a reduction as the lookup delay ratio r_{ms} does not matter anymore and hence, P_{lms} is always below 100%. The contour plot helps to qualitatively investigate the parameter ranges of p_{ext} and r_{ms} for which our proposed local mapping service leads to a reduction. However, for a quantitative estimate, we plot the empirical CDF for P_{lms} over all scenarios in Figure 3.23. The x-axis shows the different values of P_{lms} and the y-axis shows the corresponding empirical cumulative distribution values. It can be seen that our proposed local mapping service leads to a reduced lookup delay for about 90% of the scenarios. For the other 10% of the scenarios, we get an increased lookup. However, the increase is in the most cases less than 10%. Overall, our proposed local mapping service leads to a significant improvement compared to the basic LISP-MN architecture without local mapping service.

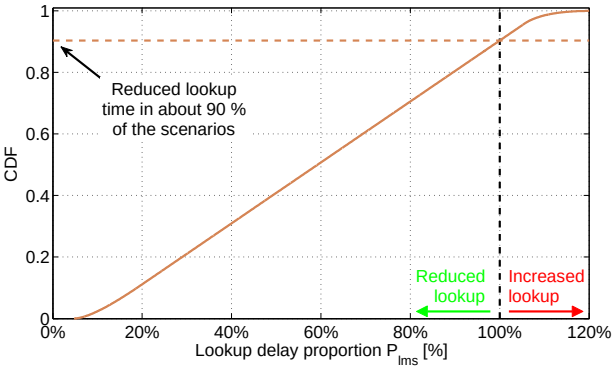


Figure 3.23: CDF plot showing improvement due to local mapping service.

3.4.3 Mobility Handover Performance

Another critical point besides the increased communication delay due to mapping lookups is the handover performance during a roaming event between different types of domains. In the remainder, we use our developed LISP simulation framework to investigate the handover performance under different scenarios.

Simulation Setup

The architecture of the simulation setup can be seen in Figure 3.24. The objective is to evaluate the handover performance of a LISP MN (*lispMobileNode*) which roams between a LISP domain, a non-LISP domain, a non-LISP domain behind a NAT gateway, and another LISP domain. The MN moves from the left-hand side to the right-hand side and back. This way, all possible six handover combinations of source and destination domain type are covered (see Figure 3.24).

During its movement, the MN with the UDP application module *UDPVideoStreamCli* requests a video stream from either the stationary node (*lispDomain3_SN*) inside the LISP domain or from the stationary node (*nonLispDomain2_SN*) inside the non-LISP domain. Both nodes may act as video stream server and have a *UDPVideoStreamSvr* UDP application module. The *UDP-*

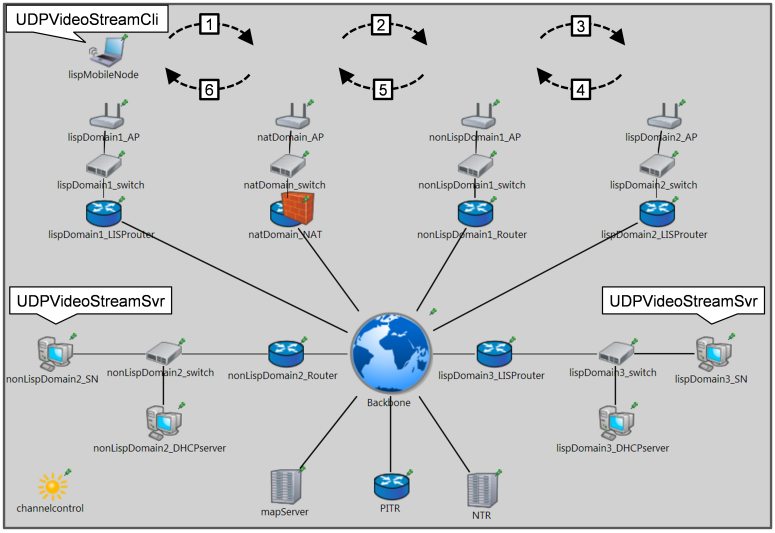


Figure 3.24: MN moving from left-hand side to right-hand side and back.

VideoStreamSvr module sends the video stream with a constant bit rate to the requesting *UDPVideoStreamCli* module inside the *lispMobileNode* network node.

The wireless network is modeled as IEEE 802.11g WLAN. The link delays for all intra-domain link delays are set to 1 ms whereas the inter-domain link delays are set to 10 ms. In addition, the link delays to the *mapServer*, to the *proxyITR*, and to the *NTR* are set to 50 ms delay. The higher link delays for these nodes were chosen to better see the differences between the analyzed handover scenarios.

Effect of Multi-Homing

As first evaluation, we investigate how the video stream from LISP and non-LISP domains is influenced during the handover. We further show how a MN with several interfaces could use multi-homing to reduce the handover delay.

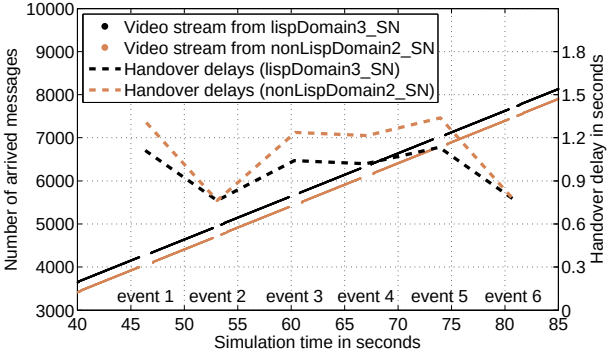


Figure 3.25: Single-homed MN.

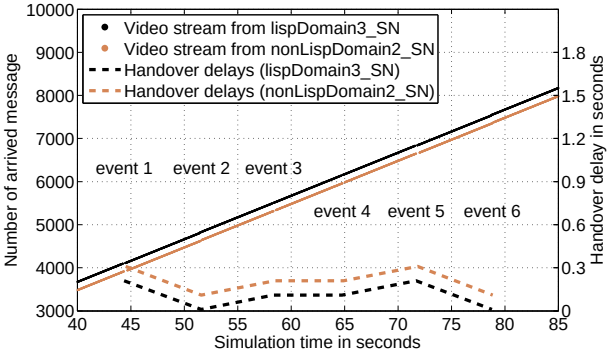


Figure 3.26: Multi-homed MN.

As remote cache update mechanism, we use the solicit map request mechanism described in Section 3.1.1 for the communication with the LISP domain. For the communication with the non-LISP domain, the extended solicit map request mechanism that is also described in Section 3.1.1 is used.

Figure 3.25 shows the effect of the handover for a single-homed MN. The black solid line denotes the video stream from the *lispDomain3_SN* node and the brown solid line denotes the video stream from the *nonLispDomain2_SN* node. The video stream from the *lispDomain3_SN* node was requested earlier than the stream from the *nonLispDomain2_SN* node. Hence, the black line appears on top of the brown line. In addition, the black and brown dashed lines

show the corresponding handover delays for the six described events in a different scaling. We see that for both scenarios, the video stream is interrupted during the six handover events but in all cases, the video stream continues after the MN has registered its new mapping and has updated the remote caches in either the *lispDomain3_LISProuter* or the *proxyITR*. The duration of each handover is between 0.8 and 1.3 seconds and there is 100 percent packet loss for all scenarios within this time.

Figure 3.26 now shows the results for a MN which uses multiple interfaces to decrease the handover delay. We see that the handover delay is significantly reduced compared to the single-homed case. As soon as the MN detects a new access point on its other currently not connected interface, its starts the association process with that new access point. Once this is done, it requests an address via DHCP and registers this address as its current locator at its map server. During the LISP signaling process, the MN is connected to both access points. Hence, packets via the old access point still reach the MN. Once the new mapping is available at either the *lispDomain3_LISProuter* or the *proxyITR*, the packets arrive at the MN via the new access point.

Influences on the Handover Delay

During the handover in the single-homed case, several elements influence the total handover delay. At first, it takes some time until the MN recognizes that it has moved out of the wireless range of its associated access point. In our model, each access point announces its presence with beacon frames which are transmitted every 100 ms. If a MN does not receive a beacon frame for 350 ms, it infers that it has moved out of the range of its access point. At this point, it starts the scanning for other wireless access points again. For our model, we use the passive scanning method with a *maxChannelTime* of 300 ms. During this timespan, the MN listens for other beacon frames on a certain channel. This is done for all possible channels. In our model, we allow only one channel. Hence after one *maxChannelTime*, if the MN has discovered another access point, it starts

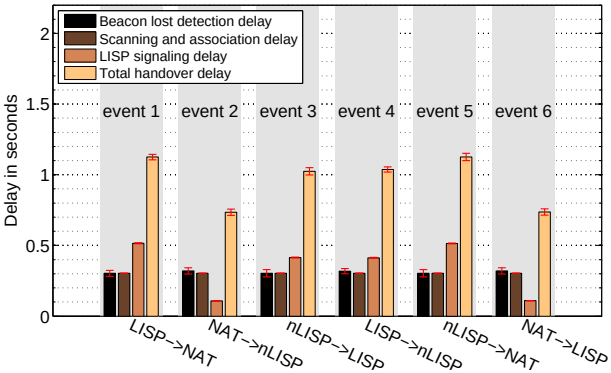


Figure 3.27: Video stream from LISP domain with solicit map-request update mechanism.

the association process with that access point. Once it is associated, it starts the DHCP discovery and requests a DHCP lease. After it has successfully obtained a lease, it starts the LISP signaling process. This process comprises the registration of the new mapping and the update of remote caches of communication partners.

In Figures 3.27 and 3.28, we show the mean and 95% confidence interval for the individual delays during a handover event for the single-homed scenario and for all six possible handover events. Again, these events occur when the MN in Figure 3.24 moves from the left-hand side to the right-hand side and back. To calculate the mean and the 95% confidence interval, we have conducted each experiment eight times

The beacon lost detection delay, and the scanning and association delay take as expected about 300 ms. The larger confidence interval for the beacon lost detection delay is due to the exact moment within one beacon frame window (100 ms), when the MN leaves the wireless coverage. The other delays are not directly influenced by random numbers and hence, the confidence interval for those is rather small. Comparing the delays of the different events, only the LISP signaling delay differs. Roaming into the NAT domain has the longest LISP signaling delay because the packets need to be relayed via the NTR to traverse the NAT (events 1 and 5). This relaying process adds an additional delay of 100 ms. This is not

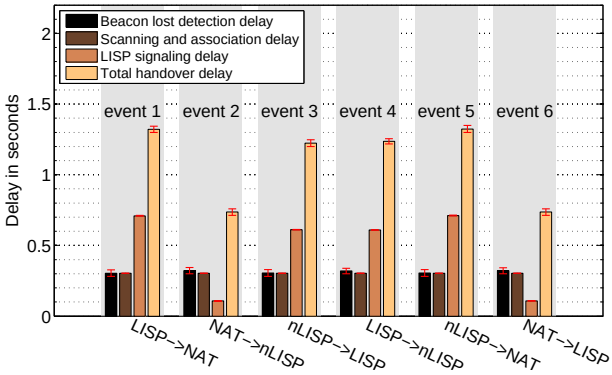


Figure 3.28: Video stream from non-LISP domain with solicit map-request update mechanism.

necessary during handover events between LISP and non-LISP domains (events 3 and 4). Hence, these LISP signaling delays are 100 ms lower. Finally, roaming from a NAT domain has the lowest LISP signaling delay (about 100 ms) as the NTR serves as anchor point (events 2 and 6). In this case, it is sufficient to update the mapping at the NTR. Once this is done, the NTR relays the packets to the new domain. The relaying process via the NTR is done until the new mapping is also available at the *lispDomain3_LISProuter* node. At this point, the packets are sent directly as the NTR is not necessary anymore.

For the video stream from the *nonLispDomain2_SN* node, we see a similar behavior except that the detour via the proxy-ITR prolongs the LISP and the total delay for all scenarios except the scenarios where the MN roams from the NAT domain to another domain (events 2 and 6). In this case, there is no detour via the proxy-ITR and hence, the delay is the same for both scenarios.

Different Remote Cache Update Mechanism

In Figures 3.29 and 3.30 we show the individual delays for the video stream from *lispDomain3_SN* node and *nonLispDomain2_SN* node respectively when the piggybacking mapping data mechanism as described in Section 3.1.1 is used.

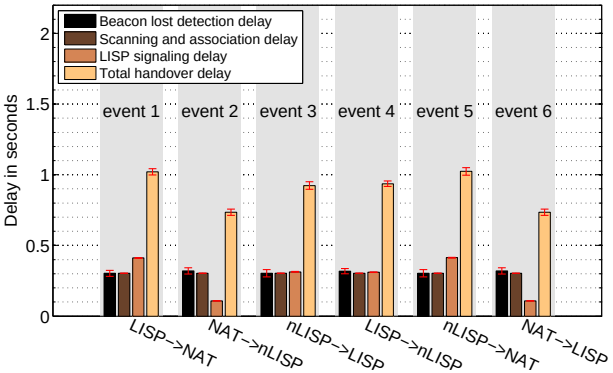


Figure 3.29: Video stream from LISP domain with piggybacking update mechanism.

In the first scenario, the piggybacking mechanism avoids one mapping lookup (100 ms) at the *lispDomain3_LISProuter* node for all handover events except for those where the MN roams from the NAT domain to another domain (events 2 and 6). This is because in this case, the lookup does not affect the handover at all as the traffic is relayed via the NTR. In the second scenario, the piggybacking mechanism also avoids one mapping lookup at the *proxyITR* which results in 200 ms lower handover delay. Again, this is only true for handover events, where the NAT domain is the source domain of the handover event (events 2 and 6). In this case again, no detour via the proxy-ITR is required and hence no mapping lookup is involved.

3.5 Lessons Learned

The objective of this chapter was to address open challenges that arise in a scenario where LISP with its mobility extension LISP-MN is used to enable seamless service and endpoint mobility. A detailed forwarding analysis revealed that LISP-MN includes several shortcomings related to unnecessary mapping lookups, path stretch due to triangle routing over proxy LISP gateways, and

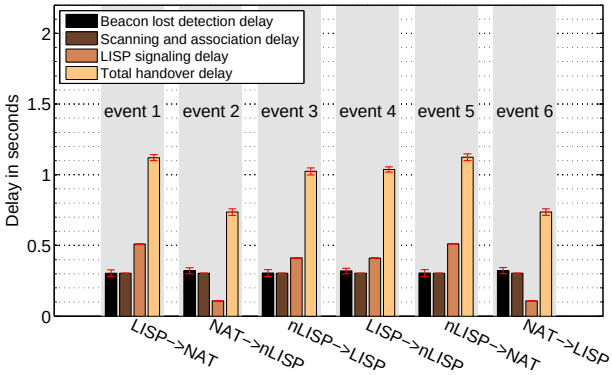


Figure 3.30: Video stream from non-LISP domain with piggybacking update mechanism.

double encapsulation headers. However, these shortcomings can be mitigated by tweaking the forwarding and encapsulation behavior of (P)ITRs and by introducing a modified mapping service that comprises a local and a global component. The actual improvement due to the local mapping service depends on the fraction of communications to external nodes and the ratio between local and global mapping lookup delay. For our investigated parameter range, we demonstrate that the local mapping service leads to a reduced lookup time in about 90% of the scenarios. Further, the local mapping service can be used in combination with an adapted encapsulation and decapsulation behavior of ITRs and ETRs to avoid double encapsulation between LISP domains. In general, our proposed improvements constitute a tradeoff between implementation complexity and communication delay and packet header overhead.

In addition to the developed control and data plane improvements, we have shown that MNs behind NAT gateways are not supported by the basic LISP-MN architecture because MNs try to register private and only locally routable care-of-addresses in the mapping system. Our proposed mechanism makes the MN globally reachable and allows transparent communication with other nodes in the Internet in spite of NAT. While the primary motivation for the presented NAT traversal was to make MNs behind NATs reachable in the Internet, it can also be

used to make several LISP domains reachable behind a NAT. Our proposed NAT traversal mechanism was adopted by the LISP working group and further implementation details for production deployments are now discussed there [114].

As last contribution, we developed a LISP simulation framework that includes all necessary LISP extensions and can be used to evaluate the LISP performance under several conditions. In particular, we investigated the handover performance for MNs in a video streaming scenario and showed that the handover delay is significantly reduced in case MNs utilize more than one wireless interface. Further, we presented a detailed evaluation of the handover delay for various roaming events and compared different cache update mechanisms. The longest handover delay can be observed if the MN roams into domains behind a NAT gateway, uses the SMR mechanism to update remote mapping caches, and receives a video stream from non-LISP networks. However, by using the piggybacking update mechanism, this handover delay is reduced.

Overall, the improvements to LISP-MN reduce the communication delay and packet overhead and the proposed NAT traversal mechanism restores connectivity behind a NAT. Finally, the developed simulation framework provides a valuable tool to evaluate the LISP protocol under various conditions.

4 Video Quality Monitoring

In the previous chapter, we discussed why a flexible new routing architecture like LISP is required to achieve minimal path stretch in scenarios where service component and endpoint mobility are supported. Both mechanisms enable an adaptation of the used paths between an offered service, i.e., video streaming and the consuming users. The adaptation with LISP is realized by changing the entries inside the mapping service. Hence, the mapping system constitutes the control component, as introduced in Section 1.1.1. However, to make evidence-based decisions, a suitable monitoring component is required that is able to monitor the user-perceived quality for specific applications, e.g., video streaming.

The challenge concerning video quality monitoring in the network is to provide a solution that is both accurate with respect to the estimated user-perceived quality as well as scalable with respect to the deployed monitoring agents in the network. There are several monitoring solutions available that try to infer the video quality from technically measurable parameters. The underlying idea of these approaches is to include available information from the coded video stream to provide an approximation of the perceived video quality. Such approximations however introduce errors, since context and content information may not be reflected sufficiently by such a solution. Other approaches decode the video stream to include detailed knowledge about the video content in the approximation of the user-perceived video quality. Such approaches provide the necessary accuracy but involve monitoring agents that decode the video stream and are hence not scalable since high computational expense is induced per video stream.

To address this challenge, we present a video quality monitoring solution that uses precomputed information about frame losses to improve the accuracy of

the monitoring in the network. This precomputed information is distributed to monitoring agents in the network so that content and codec unaware agents are supported. This design both ensures scalability as well as accuracy. However, including all possible frame loss combinations per GOP introduces a large number of frame loss scenarios and hence, excessive computing power is required. To further improve the scalability of our approach, higher frame loss scenarios are approximated by adding the distortion of single frame loss scenarios. Hence, only the distortions for single frame loss scenarios need to be precomputed. This approach however reduces the accuracy compared to a full reference approach.

Hence, we evaluate the accuracy of our solution by comparing it with a full reference approach for different frame loss scenarios. As an example, we apply the *Structural SIMilarity (*SSIM) metric [45] as *Video Quality Assessment (*VQA) metric to precompute the frame distortions. Further, we consider different high definition test video sequences and GOP structures and investigate the influence on the accuracy of our proposed approximation. In addition, to prove the rationality of our solution, we compare it for different RTP loss scenarios with the current state-of-the-art in network-based video quality monitoring. This comparison involves the correlation with the full reference SSIM metric as well as the percentage of wrongly classified GOPs.

Large parts of this chapter are taken from [13, 14]. Its remainder is structured as follows. First, we give necessary background information about video coding with the currently most used standard H.264/AVC and describe the research challenges for the developed monitoring solution. Second, we present related work in the area of video quality estimation methods in general and video quality monitoring in the network in particular. Third, we detail our proposed monitoring solution and explain the different building blocks. Fourth, we evaluate the accuracy of our proposed solution in various frame loss scenarios and compare our approach for different RTP loss scenarios with the state-of-the-art in video quality monitoring. Finally, we conclude the chapter with the lessons learned.

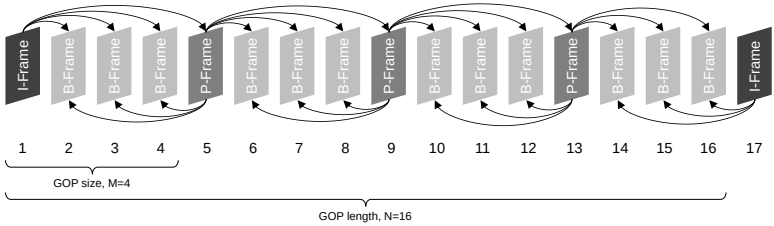
4.1 Background

Subsequently, we introduce necessary details about video coding with H.264/AVC and explain, how the resilience of coded video streams against packet loss can be influenced. Thereafter, we detail open research question with respect to our proposed monitoring system. Finally, we present the different test video sequences that were used throughout the evaluation presented in this chapter.

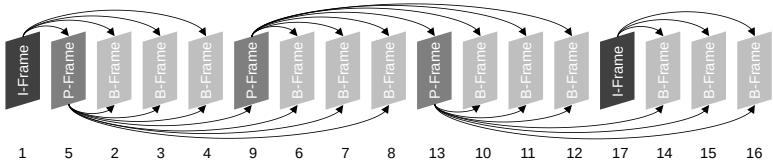
4.1.1 Video Coding with H.264/AVC

The currently most used standard for the distribution of digital high definition video is the 10th part of H.264/MPEG-4 [115], also known as *Advanced Video Coding (*AVC). AVC is a block-based motion-compensated coding framework that offers a high flexibility. It can be easily adapted to serve different application domains comprising different video bitrates and video resolutions as well as a variety of networks and architectures like Internet streams, mobile streams, and video broadcast. The motivation for the development was the increasing demand for better coding efficiency due to growing popularity of high definition video.

Video coding in general is necessary to reduce the video bitrate and the storage requirement of high definition video. An uncompressed two hour video with full high definition content for example would otherwise consume more than 500 GB [116]. However, there is a lot of redundant data within a single frame or between different frames that enables efficient video compression. Within a frame, spatially close pixels often share similar color and this spatial redundancy can be used to compress the video data. Between subsequent frames, there are also large parts which are often very similar, e.g., due to camera panning. This temporal redundancy can be used to compress the video data by transmitting only the difference information instead of the entire frame. This mechanism is called motion compensation and the difference information is stored in so-called motion vectors. Depending on the contained information and the referenced frames, three different frame types are distinguished.



(a) Frames in display order.



(b) Frames in decoding order.

Figure 4.1: Frame types, GOP structure, and display versus decoding order [116].

I-frames (intra coded frames): fixed reference frames that are independent of other frames and can be decoded without requiring other frames.

P-frames (predictive coded frames): comprise motion compensated difference information stored in motion vectors and reference either the preceding I- or P-frame.

B-frames (bidirectionally predictive coded frames): are similar to P-frames and contain motion compensated difference information but reference both the preceding and the following I- or P-frame.

The order of the frames in a coded video stream is specified with the so-called *Group of Pictures (*GOP) structure. Each GOP usually starts with an I-frame and contains several P- and B-frames. The two values *GOP size M* and *GOP length N* define the exact amount of P- and B-frames within a GOP. The distance between

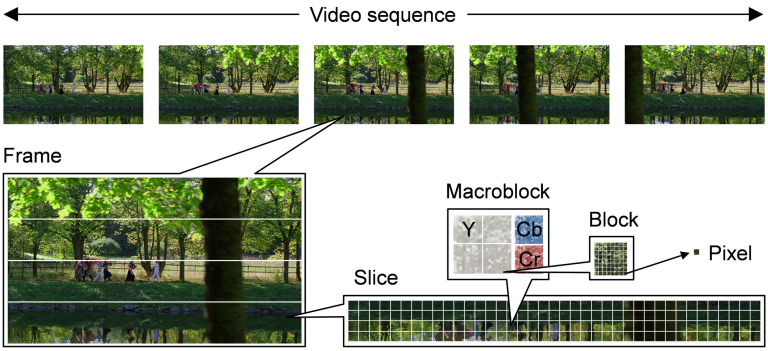


Figure 4.2: Elements of a video sequence with H.264/AVC [116].

two anchor frames, i.e., I- or P-frame is denoted as *GOP size M* and the distance between two I-frames is called *GOP length N*. Figure 4.1a shows an example of a GOP structure with *M=4* and *N=16*. The numbers below each frame denote the frame index and the arrows above and below specify the information flow. An arrow from P- to B-frame indicates that the information from the P-frame is required by the B-frame. The frames are arranged in the order at which they are played out by the video player during playback, i.e., the display order. However, to decode the coded video stream, this order is not feasible as required P-frames arrive later than the referring B-frames. This can be seen by arrows in left direction in Figure 4.1a. Hence, the order at which frames are stored in the bitstream and at which they arrive at the decoder is different to the display order. The corresponding decoding order for the example in Figure 4.1a can be seen in Figure 4.1b. Now, P-frames arrive before B-frames to ensure that all reference frames are decoded before the frames that depend on them. Hence, there are no arrows in left direction anymore and the information only flows from the left to the right.

The AVC framework comprises two different layers, i.e, the *Video Coding Layer (*VCL) and the *Network Abstraction Layer (*NAL). The VCL is responsible

for the encoding of the uncompressed video data. It partitions the frames into *Macroblocks (*MBs) and searches for similar blocks to apply motion compensation. The partitioning of frames and the different resulting elements are shown in Figure 4.2. A frame is first divided into different slices which then comprise several MBs. The MBs can be further divided into sub-blocks which eventually contain the pixel information. The pixel information inside a MB is grouped in separate blocks for luminance (Y) and chrominance (Cb and Cr) information. To further reduce the amount of stored color information, chroma sub-sampling is used. The most common color structure 4:2:0 comprises four 8x8 luminance blocks and two 8x8 chrominance blocks (see Figure 4.2).

The NAL then formats the VCL representation of the video and adds header information required by storage media or lower layer transport protocols such as RTP [117]. The RTP payload format for H.264 video for example is described in [118]. So called *Network Abstraction Layer Units (*NALUs) serve as container and comprise either a single slice or a group of slices. The NALUs are then encapsulated in RTP packets and a single packet either contains only one NALU or aggregates multiple NALUs. Fragmentation of larger NALUs across several RTP packets is also possible. The way the NALUs are created, encapsulated, and sent over the network mainly influences the resilience of the video stream against packet loss.

4.1.2 Research Challenges

The main objective addressed in this chapter is to evaluate our proposed network-based video monitoring solution that infers the user-perceived quality from packet loss measured in the network. The innovative concept of our approach is to precompute distortion values for certain loss scenarios, which are then distributed to monitoring agents in the network. These agents simply monitor the seen video streams and assess packet and more importantly frame loss on a per GOP basis. The user-perceived quality is then calculated by looking up the corresponding precomputed distortion values belonging to the detected lost frames. However,

due to complexity reasons, our approach does only precompute the distortion values for one frame loss scenarios and approximates higher loss scenarios with the distortion values computed for the one frame loss scenarios. This approximation however possibly reduces the accuracy of our approach in comparison to a full reference metric that uses both the undistorted and the distorted frame to calculate the video quality. Hence, one major research question addressed in the following chapter is to identify whether the introduced approximation significantly reduces the accuracy of the proposed method. As an example, we evaluate the accuracy using the SSIM metric as objective video metric to calculate the distortion values since this metric is known to correlate well with human perception.

Another important aspect involves other available monitoring solutions that include different technically measurable parameters to infer the user-perceived video quality. Depending on the amount and type of included information, the complexity and the accuracy of these solution changes. Some approaches simply monitor lost packets and assess the video quality based on predefined loss thresholds. Other more complex approaches decode the video streams in the network and assess the video quality based on the decoded content information like motion vectors, prediction residuals, and coding modes as well as concealed motion vectors. Our approach in contrast is a good tradeoff between accuracy and complexity since the precomputation enables content and codec unaware monitoring agents that simply need to lookup distortion values for precomputed loss scenarios. Hence, another research question addressed in the remainder is how our idea performs in comparison to other approaches that include different kinds of information.

4.1.3 Test Video Clips and GOP Structures

To address the research questions mentioned above, we consider several kinds of test sequences and GOP structures so that our evaluation reflects a wide area of different streaming configurations. First, we present the considered video sequences and second, we explain the chosen GOP structures.

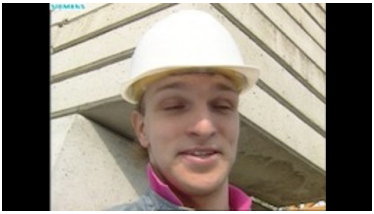
Table 4.1: Considered test video sequences.

Sequence	Resolution	SI	TI	Length
Foreman	QCIF (176x144) CIF (352x288)	Medium	Medium	$\frac{300 frames}{30\ fps} = 10\ s$
Soccer	4CIF (704x576)	Medium	Medium	$\frac{600 frames}{60\ fps} = 10\ s$
OTC	720p (1280x720) 1080p (1920x1080)	Low	Low	$\frac{500 frames}{50\ fps} = 10\ s$
DTO	720p (1280x720) 1080p (1920x1080)	Medium	Medium	$\frac{500 frames}{50\ fps} = 10\ s$
PJ	720p (1280x720) 1080p (1920x1080)	High	High	$\frac{500 frames}{50\ fps} = 10\ s$

Test Video Sequences

The following videos were chosen to represent different horizontal and vertical resolutions ranging from *QCIF* to *1080p*. In addition to different spatial resolutions, we also consider sequences with different amount of spatial and temporal information for the *720p* and *1080p* resolution since the focus is on high definition video streaming. Temporal information includes the motion between consecutive frames while spatial information includes the amount of details per single frame, as introduced in ITU-T Recommendation P.910 [119]. All considered video sequences are publicly available via the Xiph.org test media website [120].

The videos are listed in ascending order according to the spatial resolution and according to the amount of spatial and temporal information for video sequences with equal spatial resolution. Table 4.1 gives an overview over the considered sequences and the respective spatial resolution as well as spatial and temporal information. The considered sequences are *Foreman*, *Soccer*, *Old Town Cross (OTC)*, *Ducks Take Off (DTO)*, and *Park Joy (PJ)*. Screenshots for the five videos are shown in Figure 4.3. *Foreman* and *Soccer* have an aspect ratio of $4:3$ whereas the HD content has a ratio of $16:9$.



(a) Foreman.

(b) Soccer.

(c) Ducks Take Off.

(d) Old Town Cross.

(e) Park Joy.

Figure 4.3: Screenshots of video test sequences.

Foreman: shows a foreman in front of a newly constructed building. After a few seconds, there is a panning shot showing the entire construction site (see Figure 4.3a).

Soccer: displays a soccer match on artificial lawn between two teams during soccer practice (see Figure 4.3b).

Old Town Cross: is a panning shot of a big city filmed from an elevated position (see Figure 4.3d).

Ducks Take Off: shows a group of ducks swimming in the water. After a while, they take off and cause water ripples propagating across the lake (see Figure 4.3c).

Park Joy: displays a park with people in the background and trees passing by in the foreground (see Figure 4.3e).

Assessed GOP Structures

In addition to the influence of different types of videos, we also investigate how well our proposed monitoring solution behaves for different GOP structures. For that purpose, we have analyzed the GOP structures currently used for live streaming of video content by two prominent German IPTV broadcasters, i.e. the German Telecom and the German public service broadcasters (ARD/ZDF). The German Telecom offers an IPTV service called T-Entertain which can be booked in addition to the DSL Internet connection. Entertain runs in a separate VLAN and provides access to various TV channels in high and standard definition quality. The GOP structure for Entertain in HD quality is $M = 8, N = 64$, whereas M denotes the distance between P-frames and N the distance between I-frames. Hence, for Entertain HD, there are 7 B-frames after an I- or P-frame and between two I-frames, there are 63 P- and B-frames. This kind of GOP structure reduces the amount of transmitted information but the encoding and playout order of frames is different which increases the complexity at decoder side. For the ARD/ZDF live TV stream in contrast, the GOP structure is different. There are no B-frames at all and also the length of the GOP is variable and not fixed. A possible explanation for the variable GOP length is that the length is adaptive to the video content to reduce the video bit rate. Such an approach is for example proposed in [121]. Hence, the GOP structure for the ARD/ZDF life stream can be written as $M = 1, N = variable$. This kind of structure has an increased video

Table 4.2: Considered GOP structures.

Label	Size	Length	Structure
IBP	M=4	N=16	IBBBPBBBPBBBPBBB
IPP	M=1	N=16	IPPPPPPPPPPPPPPP

bitrate compared to the structure with B-frames but the encoding and playout order of frames is the same. For the evaluation, we use adapted GOP structures considering the high definition video test sequences. The GOP length for both structures has been set to 16 frames and is not dependent on the video content anymore so that both structures are comparable. For the structure with B-frames, we have reduced the GOP size to 4 which results in 3 B-frames in between I- and P-frames. The resulting modified GOP structures can be seen in Table 4.2. For convenience, the structures with and without B-frames are denoted as *IBP* and *IPP* structures respectively.

4.2 Related Work

In this section, we first describe related work in the field of video quality assessment in general and second, in the field of video quality monitoring in networks in particular.

4.2.1 Video Quality Estimation Methods

There are various methods available to assess the user-perceived quality for video applications. These methods are usually classified according to whether a subjective or objective QoE measure is produced. Estimating the subjective QoE usually involves well-defined test environments and settings under which subjects rate the quality of impaired video sequences. The tests are designed so that the ratings can be mapped to for instance numerical values from 1 (bad quality) to 5 (excellent). The average user rating for each specific test configuration then

yields the *Mean Opinion Score (*MOS) value [122]. The obtained MOS value from such controlled tests reflects the user's perception and hence has high significance. However, due to the varying quality awareness of human observers, multiple subjects are required to participate in a subjective study [123]. According to [124], at least 15 subjects should participate so that significant results can be obtained. On the one hand, these tests produce highly relevant ratings but on the other hand, they need to be conducted manually in a controlled environment and hence are time-consuming and costly. Thus, a reasonable tradeoff between significance and complexity is to use the subjective data as base for objective video quality algorithms that automatically predict the visual quality of a video clip. Metrics that yield objective video quality can be further classified into three categories due to the required amount of reference information [125, 126]:

Full-Reference (FR) metrics assess the video quality of a test video by comparing it frame-by-frame with the original video.

Reduced-Reference (RR) metrics require the test video and a set of parameters derived from the original video to compute a measurement.

No-Reference (NR) metrics require only the test video and hence, have to make assumptions about the original video.

The full reference metrics are typically publicly available and hence, can be used to compute the quality of a transmitted video clip. Examples are the *Peak Signal to Noise Ratio* (PSNR), the *Structural SIMilarity* (SSIM) metric, and the *Video Quality Metric* (VQM).

4.2.2 Video Quality Monitoring in the Network

Different quality monitoring mechanisms of video over IP networks have been investigated in research. The most simple mechanism is to define a packet loss threshold for the IPTV service and assume the video quality as acceptable as long as the threshold is not exceeded. This technique does not take any video

and content information into account. While a lost packet will produce a large error in regions with medium motion, it may produce no sizable error in regions with low motion. The mechanism introduced by Reibman et al. [127] focuses on no reference methods which estimate the video quality on network level and, if possible, on codec level. The estimation on codec level includes for instance spatio-temporal information and effects of error propagation. Tao et al. [128] propose a relative quality metric, rPSNR, which allows the estimation of the video quality against a quality benchmark provided by the network. The introduced system offers a lightweight video quality solution. Naccari et al. [129] introduce a no reference video quality monitoring solution which takes spatio-temporal error propagation as well as errors produced by spatial and temporal concealment into account. The results are mapped to SSIM and compared to results gained by computing the SSIM of the reference video and the distorted video. All these video quality monitoring mechanisms work on no reference or reduced reference metrics for estimating the video quality. A brief overview of current research questions within the area of IPTV monitoring can be found in [41].

4.3 Proposed Monitoring Solution

The considered scenario for our proposed monitoring solution is a virtual network that has been created by combining the virtual resources of several PIPs to host an IPTV streaming service (see Figure 4.4). A VNO controls and manages the network and is responsible for a good service quality at customers. Hence, the VNO deploys monitoring agents at critical locations in the network so that the monitoring can detect the location of possible performance problems. The agents either run directly on intermediate nodes or on dedicated monitoring nodes that receive mirrored traffic from intermediate nodes like routers or switches. With traffic or port mirroring, the data traffic is duplicated during the forwarding process and sent to preconfigured nodes which perform predefined traffic analysis or monitoring tasks. Due to emerging technologies like OpenFlow, this may even be achieved on a per-flow and hence per video stream basis. The IPTV service

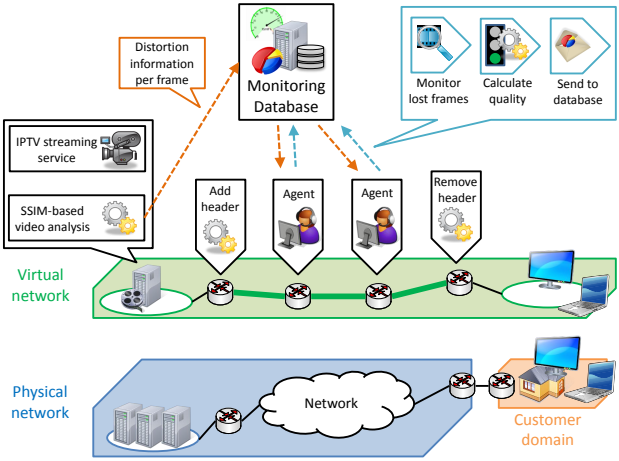


Figure 4.4: Possible realization of proposed video monitoring solution.

is connected to a cloud environment that is able to perform a fast SSIM-based video analysis on a per GOP basis. The SSIM-based video analysis is required to compute the distortion for loss scenarios where exactly one frame is lost. In addition, the inter-frame dependencies within a GOP are extracted. This information is then distributed via a central monitoring database to the monitoring agents in the network. The agents monitor the multicast video streams and map monitored packet losses to the video quality and send this information to the central monitoring database. To track the transmitted frames, the monitoring agent can use deep packet inspection to discover the necessary information from the video frame header. However, this may constitute performance problems as a lot of streams may pass through the monitoring agent. Another possibility is to provide information via an additional shim header between transport and application headers. It includes the frame index and the number of packets per frame. The header is added at the edge close to or by the video streaming servers and is re-

moved at the border to the customer domain. The agents then monitor the seen packet indexes in the shim header. If there is a gap in the frame index, the agent considers that frame as lost and calculates the service degradation on consumer side.

4.3.1 Precomputation of Distortion

Our proposed monitoring solution uses detailed knowledge about the video to calculate how much influence a specific lost packet and hence lost frame would have on the service quality for consumers. We apply a cloud computing based live analysis of the streamed video and generate distortion information for loss scenarios where exactly one frame within a GOP is lost. The distortion values are computed according to the SSIM metric and we define the distortion as the dissimilarity of two frames. Besides the SSIM metric, several other methods, e.g. the video quality metric (VQM) are possible and we have chosen the SSIM metric as it offers a fast computation and good correlation with human perception. For each frame within a GOP, the video analysis generates a loss scenario where only this specific frame is dropped and the resulting distortion on all frames within that group d_{GOP} is investigated. To that end, we directly compare the undistorted image f_{Good} with the distorted image f_{Bad} via the SSIM method and hence obtain, how different the undistorted and distorted image are. The SSIM metric yields values between *0* and *1* and the distortion value per frame d_{Frame} is defined according to Equation 4.1.

$$d_{Frame} = 1 - SSIM(f_{Good}, f_{Bad}). \tag{4.1}$$

The distortion value per single frame d_{Frame} hence has a maximum of *1* which means two completely different pictures. However, only I-frames are completely independent of other frames and constitute fixed pictures. All other frame types are dependent on other frames and if these frames are lost, the dependent frames cannot be decoded and must also be considered lost. Hence, a single frame can have a much higher distortion value in case a lot of other frames are depen-

Algorithm 2: Update process of d_{GOP}.

```
   input1: distortion of lost frame (d_Frame)
   input2: current distortion per GOP (d_GOP)
1  if lost frame not dependent on other frames then
2  |  d_GOP += d_Frame
3  else
4  |  if required frame is also lost then
5  |  |  ignore distortion value for this frame
6  |  else
7  |  |  d_GOP += d_Frame
```

dent on this frame. To normalize the distortion per group d_{GOP}, we divide it by the number of frames per group. To get the dependencies between the frames in a GOP, we also investigate in the above emulated loss scenarios which other frames are also distorted in the currently considered GOP if a specific frame is lost. In total, the precomputation of distortion values offers a high potential for parallel computing, i.e. processing the different loss scenarios on separate computing nodes within a computing cloud. The introduced lag due to the preprocessing depends on the degree of parallelization and the number of frames per GOP and due to our current experience, we assume that this process is feasible within less than 1 second.

4.3.2 Calculation of Video Distortion

The distortion value is calculated per GOP and once the agent sees the next GOP in the stream, the old distortion value of the former GOP is sent to the monitoring database and the value is reset to 0 for the next group. For each lost frame per group, the monitoring agent updates the distortion value d_{GOP} according to Algorithm 2. First, the monitoring agent checks whether the lost frame is dependent on other frames. If the lost frame is not dependent, the agent looks up the distortion value for the lost frame d_{Frame} and adds this value to the distortion value of the currently considered GOP (d_{GOP}) and the update process is fin-

ished. If otherwise the lost frame is dependent on other frames, the agent needs to check whether these frames are also lost. If the currently considered frame requires another frame which is also lost, the distortion of the current frame is already included and can be ignored. In this case, no update of the d_{GOP} value is required. If in contrast the required frame is not lost, the distortion of the currently considered lost frame is not yet included and hence, the distortion d_{Frame} is added to the d_{GOP} value.

4.3.3 Mapping to Video Quality

After the distortion per GOP d_{GOP} has been calculated, the monitoring needs to map the distortion to a proper metric showing the actual video quality. For our approach, the distortion is mapped to the MOS value according to [130] and then to the video quality according to [41]. There the authors have shown via subjective tests for web services that 90% of the users already accept a fair video quality (MOS 3). A distortion per GOP $d_{GOP} \leq 0.12$ corresponds to MOS values equal or larger than MOS 3. Hence, our monitoring solution rates GOPs with a distortion $d_{GOP} \leq 0.12$ as good or accepts the video quality and rates GOPs with a distortion $d_{GOP} > 0.12$ has bad or rejects the video quality.

4.4 Impact of Approximation on Accuracy

Subsequently, we perform an exhaustive study with respect to the accuracy of our monitoring approach. We focus on high definition 1080p video content as this is currently the quasi standard in video streaming. To be able to include a wide variety of loss scenarios, we conduct the following evaluation for entire frame loss scenarios. Therefore, we first investigate which frame loss scenarios are relevant and which higher scenarios cannot deliver a good video quality anymore. Second, we assess the influence of the different video types and GOP structures and show, how our metric can be optimized for certain GOP structures. Third, we further investigate the error due to our approximation and finally, we show the sensitivity

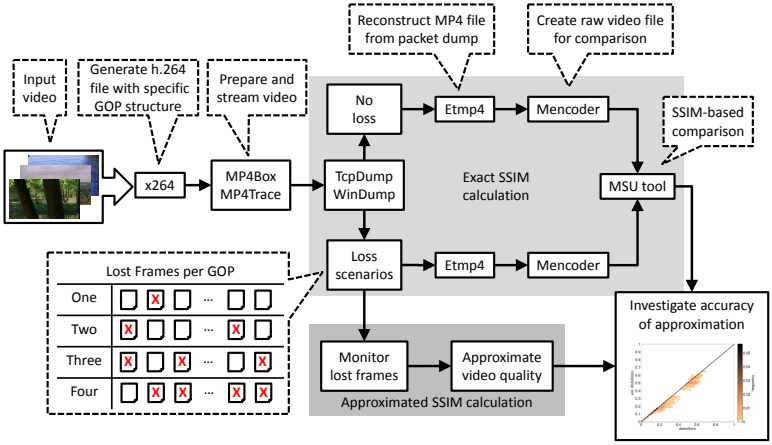
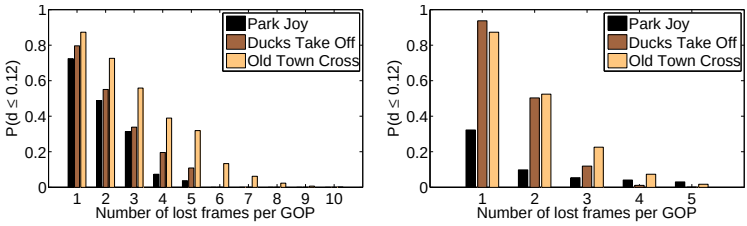


Figure 4.5: Flow chart showing the different steps during the evaluation.

of our approach with respect to the acceptance threshold for the video quality.

4.4.1 Evaluation Methods

For the evaluation of the accuracy of our proposed monitoring solution, we compare our approximated SSIM values with the exact SSIM values for different frame loss scenarios. The overall setup as well as the different steps during the evaluation can be seen in Figure 4.5. In the first step, the x264 tool [131] is used to create a h.264 file with a specific GOP structure (see Table 4.2) for the three different input videos. In the second step, we use the MP4Box tool [132] and the MP4Trace tool from the EvalVid framework [133] to create a video stream which is dumped in the next step using either TcpDump [134] or WinDump [135]. To evaluate different frame loss scenarios, we create a lossy dump file in the next step by removing certain frames from the dump file. This way, we generate different dump files for loss scenarios where exactly one, two, three, or four frames are lost per GOP. This results in $\binom{16}{i}$ scenarios in case exactly i frames are lost



(a) IBP structure. *(b) IPP* structure.

Figure 4.6: Number of GOPs with distortion value less than 0.12.

per GOP. Evaluating all frame loss combinations results in a high number of scenarios which requires excessive computing power. Hence, we limit our evaluation to at most five frame loss scenarios and show that higher loss scenarios are not required to demonstrate the accuracy of our approach. After the different dump files have been created, we use the Etmp4 tool from the EvalVid framework [133] to reconstruct the MP4 file from the dump file and create raw video files by using the MEncoder tool [136]. In the last step, the MSU tool [137] is used to compare the lossy video file with the original video file by calculating the exact SSIM values. These values are then used as reference and we evaluate the induced error due to our approximation of the SSIM metric. The information about lost frames is used by our proposed monitoring solution to calculate the approximated SSIM values (see lower part of Figure 4.5). These values are then compared with the exact SSIM values calculated by the MSU tool.

4.4.2 Relevant Frame Loss Scenarios

In this subsection, we investigate which loss scenarios lead to a large fraction of GOPs where the video quality is still acceptable. For these scenarios, our proposed monitoring solution needs to be accurate. For loss scenarios where a large fraction of GOPs has a high distortion and hence a very bad quality, accuracy is not that important. According to Section 4.3.3, our mapping rates the video

quality of a GOP acceptable if the distortion per GOP $d_{GOP} \leq 0.12$. The resulting percentage of GOPs with a distortion less than *0.12* for the three different videos is shown in Figure 4.6. Figure 4.6a shows the results for the *IBP* structure and Figure 4.6b shows the corresponding results for the *IPP* structure. In both figures, the x-axis shows the number of lost frames per GOP and the different colored bars denote the three different videos. Concerning the *IBP* structure in Figure 4.6a, we see that the fraction of GOPs with acceptable quality decreases for the higher loss scenarios. This is in line with the expectations because the more frames are lost, the worse is the overall video quality per GOP. However, there are strong differences between the different types of video which can be explained due to the amount of spatial and temporal information. The *Park joy* video sequence has the highest amount of information and is hence more susceptible to frame loss than the other two videos which have a higher number of acceptable GOPs in all loss scenarios.

Even if five frames are lost within a GOP, about 32% of the GOPs for the *Old Town Cross* test sequence still have an acceptable video quality. However, for six and seven frame loss scenarios, the percentage of GOPs with good quality drops to 13% and 6% respectively. Hence, also for the *Old Town Cross* video, it is sufficient to consider loss scenarios where at most five frames are lost per GOP because for higher loss scenarios, the probability to classify the GOPs wrong is very small as nearly all GOPs have bad video quality. Concerning the *IPP* structure in Figure 4.6b, there is a similar observation but the fraction of acceptable GOPs is much lower for all videos in the higher loss scenarios. Due to the absence of B-frames, the overall importance per frame is higher and hence, this structure is more susceptible to frame loss than the *IBP* structure. If we directly compare the number of accepted GOPs for the five frame loss scenario, we see that the *IPP* structure in the right figure has a much lower fraction as the *IBP* structure on the left figure. Overall, also for the *IPP* structure, it is sufficient to consider only loss scenarios where at most five frames are lost per GOP. For higher loss scenarios, our approach indicates an unacceptable video quality with a high probability.

4.4.3 Qualitative Evaluation of Accuracy

To show the accuracy of our developed monitoring solution, we plot the approximated distortion values against the distortion values calculated by the exact SSIM metric. The evaluation does only consider loss scenarios where exactly two, three, four and five frames are lost. One frame loss scenarios are not considered as for those scenarios, our approximation yields the same results as the exact SSIM metric.

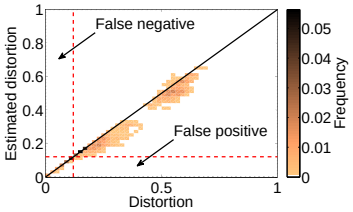
The results are visualized as scatter plots where the x-axis denotes the exact distortion values and the y-axis denotes the estimated distortion values. All plots also contain the identity line through the origin. Estimated distortion values which lie on that line perfectly match the exact distortion values. In addition, a vertical and horizontal dashed line at the corresponding 0.12 acceptance thresholds partition the figures into four different areas. The following classification of the four different areas is motivated by the terms introduced in statistical test theory:

True positive The estimated as well as the exact distortion values are both equal to or below the acceptance threshold of 0.12 and hence, both rate the video quality as good and no error occurs.

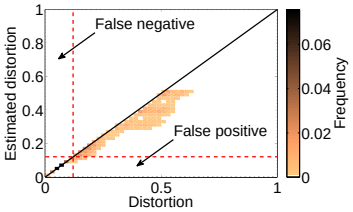
True negative Both distortion values are higher than the acceptance threshold of 0.12 and hence, both reject the video quality and again, no error occurs.

False negative The estimated distortion values are larger than the acceptance threshold while the exact distortion values are lower than or equal to the acceptance threshold. In this case, our proposed monitoring would reject the video quality while the exact metric would accept the quality.

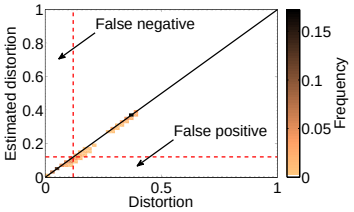
False positive The estimated distortion values are lower than the acceptance threshold while the exact distortion values are higher. In this case, our proposed monitoring accepts the quality while the exact metric would reject the video quality.



(a) Park Joy with *IBP* structure.



(b) Ducks Take Off with *IBP* structure.

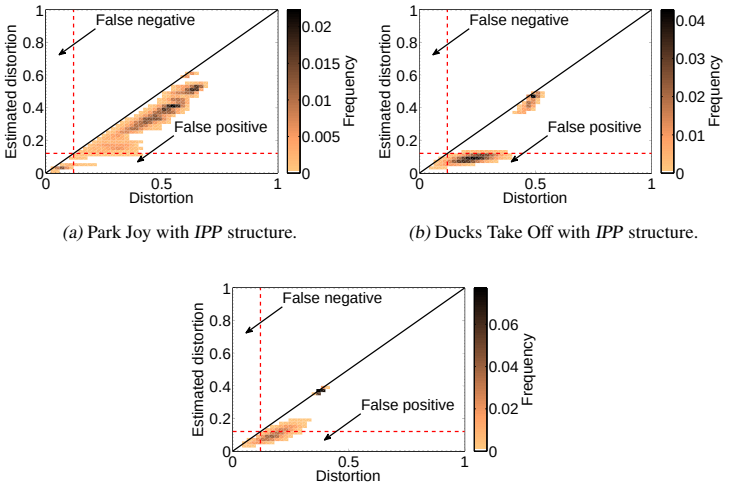


(c) Old Town Cross with *IBP* structure.

Figure 4.7: Scatter plots showing estimated against exact distortion values.

Considering the results for the three different videos with *IBP* structure shown in Figure 4.7, we see that our approach performs best for the *Old Town Cross* video which has the lowest spatial and temporal information (see Figure 4.7c). For the other two video sequences, our approximation still performs very well in the critical area around the video quality acceptance threshold of *0.12* as nearly no points lie in the *false negative* or *false positive* areas. Only for the higher distortion area, i.e., the *true negative* area, our estimated values deviate from the exact values. There however, our monitoring does not accept GOPs with a bad video quality or a distortion value higher than the acceptance threshold and hence, no error occurs.

A different observation can be seen for the results for the *IPP* structure shown in Figure 4.8. For this structure, our proposed approximation does not perform well and underestimates the distortion in the critical area around the acceptance



(a) Park Joy with *IPP* structure.

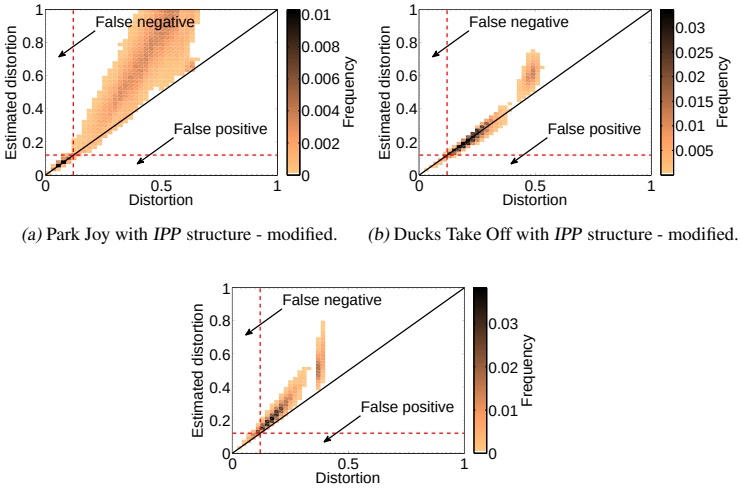
(b) Ducks Take Off with *IPP* structure.

(c) Old Town Cross with *IPP* structure.

Figure 4.8: Scatter plots showing estimated against exact distortion values.

threshold for all three videos and a lot of points lie in the *false positive* area.

For this structure, all subsequent frames are always dependent on their precedent frames and errors in earlier frames influence all subsequent frames. Our approximation however ignores the distortion values for frames which are dependent on earlier frames (see Section 4.3.2) as the distortion of dependent frames is included in the distortion value of their required frame. This is a good approximation for GOP structures with minor inter-frame dependencies like the *IBP* structure but not for the *IPP* structure. Hence, to improve our approach, we modify the calculation of the distortion per GOP d_{GOP} and always add the distortion of lost frames d_{Frame} instead of ignoring the distortion of frames which are dependent on another lost frame. This is a very simple modification to our initial metric. But for the specific use case, the results prove the viability of this approach. Figures 4.9a, 4.9b, and 4.9c show the results for the modified version of our approxi-



(a) Park Joy with *IPP* structure - modified. *(b)* Ducks Take Off with *IPP* structure - modified.

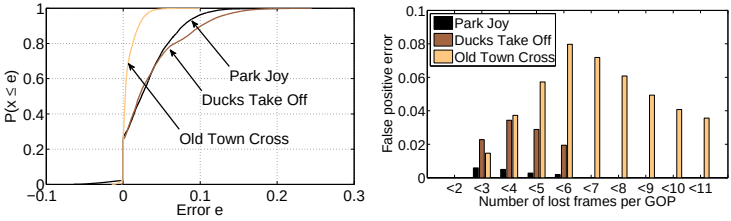
(c) Old Town Cross with *IPP* structure - modified.

Figure 4.9: Scatter plots showing estimated against exact distortion values.

mation. With the modified approach, we do not underestimate the distortion in the critical area anymore and nearly no points lie in the *false positive* area. Accordingly, a large fraction of the estimations lies on the identity line through the origin. For the higher distortion area, our modified approach now overestimates the distortion. There however, no points lie in the *false negative* area so that our monitoring does not reject GOPs with a good video quality or an exact distortion smaller than 0.12 and hence, no error occurs.

4.4.4 Quantitative Evaluation of Accuracy

The scatter plots in the former section give a basic understanding about how our unmodified and modified metrics perform for the different videos and GOP structures. However, for a quantitative statement, CDF plots are more suitable. Hence

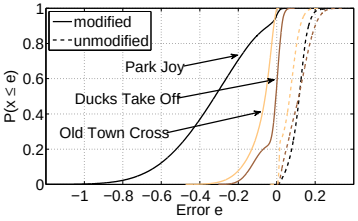


(a) CDF plot for both errors. *(b)* Frequency of false positive error.

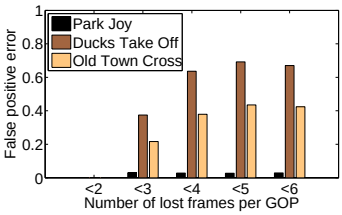
Figure 4.10: Error plots for IBP structure.

in Figure 4.10a and 4.11a, we plot the CDF for the error e between the estimated and the exact distortion values for both GOP structures. The error e is defined as the difference between the exact distortion and the estimated distortion. A negative error means that our proposed approximation overestimates the distortion and positive error means an underestimation of distortion. From the perspective of a network provider, a negative error is more serious because the monitoring underestimates the distortion and hence recognizes a bad video quality too late.

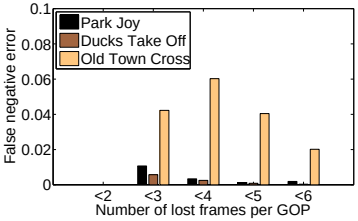
Regarding the *IBP* structure, we again see that our proposed monitoring solution performs best for the *Old Town Cross* sequence. For that video, all GOPs have an error $e < 0.05$. For the other two videos, about 80% of the GOPs have an error $e < 0.05$. However, as we have seen in Figure 4.7, the larger error occurs in a distortion range where accuracy is less important, i.e., the *true positive* area. More problematic are errors that correspond to points in the *false positive* or *false negative* areas. The percentage of *false negative* errors for all three video sequences is less than 1% and hence, we omit the plot for these errors. The percentage of *false positive* errors however is larger and the results are plotted in Figure 4.10b. The x-axis denotes the number of included frame loss scenarios and the y-axis shows the percentage of *false positive* errors. For the *Park joy* and *Ducks Take Off* video sequences, again only loss scenarios with less than five lost frames are considered. Regardless of the loss scenario, less than about 1% of the GOPs for the *Park joy* sequence are falsely accepted by our proposed monitoring



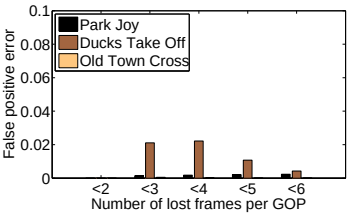
(a) CDF plot for both errors.



(b) Frequency of false positive error for unmodified metric.



(c) Frequency of false positive error for modified metric.



(d) Frequency of false positive error for modified metric.

Figure 4.11: Error plots for IPP structure.

solution while the exact metric rejects the video quality. Considering the *Ducks Take Off* video, between 2% and 4% of the GOPs are falsely accepted. Finally, the *Old Town Cross* video has the highest percentage of false positive errors of about 8% which occurs for frame loss scenarios where less than six frames are lost. For loss scenarios with at least six frames, the percentage decreases because the fraction of false positive errors per individual loss scenario decreases.

For the *IPP* structure, it can be seen that the unmodified approach always significantly underestimates the distortion and hence is not suitable for this structure. In contrast, the modified approach mostly overestimates the distortion and only a very small fraction of GOPs has a positive error e. The modified approach is hence more suitable for the *IPP* structure as the GOPs with a high negative er-

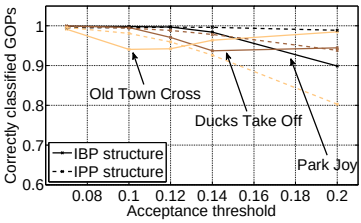


Figure 4.12: Sensitivity of proposed metric with respect to acceptance threshold.

ror e have a very high distortion and are rejected anyway, i.e., they lie in the *true negative* area. Even more obvious is the non suitability of the unmodified metric in case we consider the percentage of false positive errors in Figure 4.11b. Again, the x-axis denotes the considered loss scenarios and the y-axis shows the percentage of false positive errors. Only the *Park joy* video has an acceptable percentage of about 3%. Considering the other two videos, our monitoring falsely accepts between 20% and 60% of the GOPs while the exact metric rejects the quality. Hence, our proposed monitoring would detect a bad video quality at customers far too late. A totally different result can be seen for the modified metric in Figure 4.11d. Now, the percentage of false positive errors for the *Park joy* and *Old Town Cross* videos is nearly zero and for the *Ducks Take Off* video, the percentage is between 1% and 2%. The reduction of the false positive error however comes with an increase of the false negative error, which is shown in Figure 4.11c. Nevertheless, the increase is minimal and only the *Old Town Cross* video has a noticeable percentage between 4% and 6%. Overall, our proposed monitoring performs very well for the different videos and GOP structures and on average, more than about 95% of the GOPs are correctly classified.

4.4.5 Sensitivity with Respect to Acceptance Threshold

In the former evaluation of the accuracy of our proposed monitoring solution, we have used an acceptance threshold of 0.12 for the video quality. The video quality

of GOPs is only accepted if the corresponding distortion is less than this threshold. However, this threshold was chosen according to [41], where the authors have shown via subjective tests for web services that 90% of the users accept a fair service quality (MOS 3). If due to new findings in future work this threshold needs to be adapted, our monitoring solution should still be accurate. Hence in the following, we investigate how the percentage of correctly classified GOPs behaves for different acceptance thresholds. The corresponding results are shown in Figure 4.12. The x-axis shows different values for the acceptance threshold and the y-axis shows the percentage of correctly classified GOPs. The solid lines denote the results for the *IBP* structure and the dashed lines denote the results for the *IPP* structure. For the *IPP* structure, only the results with the modified approach are shown.

For all structures and videos, a threshold close to 0 leads to 100% correctly classified GOPs. In that case, both metrics always reject the video quality for a GOP if the distortion is larger than 0 which is not a reasonable approach as such a monitoring would be far to pessimistic. For the chosen threshold of 0.12, our monitoring classifies about 98% of the GOPs correctly. Only the *IBP* structure for the *Old Town Cross* sequence experiences a slightly lower classification rate of 93%. For an increasing threshold, the classification rate for nearly all videos and structures decreases. For the *IBP* structure, the classification rate does not drop below 90% which is still an acceptable result. For the *IPP* structure, the classification rate drops to 80% for the *Old Town Cross* video. Overall, our monitoring still achieves a high correct classification of GOPs even in the higher distortion range of about 0.2.

4.5 Performance Comparison

In the preceding section, we performed an exhaustive evaluation to assess the accuracy of our approach on frame level. Subsequently, we compare our proposed precomputation-based monitoring with another SSIM-based solution that implements a different idea to approximate the SSIM calculation. In addition

to this similar but different idea, we also consider a packet loss-based solution to show whether the effort of the approximated SSIM calculation is worth the increased accuracy in estimating the video quality for different types of video content. Therefore, we first present more details regarding the two considered opposing approaches. Second, we explain the developed framework and describe the different steps during the evaluation. Finally, we show the results of the comparison and explain, which scenarios are worth the increased effort due to the approximated SSIM calculation.

4.5.1 Monitoring Candidates

For the comparison, we selected another SSIM-based monitoring approach that differs in various details. The basic idea behind the No-Reference Video Quality Monitoring (NORM) approach proposed by Naccari et al. in [129] is to include received motion vectors, prediction residuals, and coding modes as well as concealed motion vectors to assess the distortion at macroblock level based on the *Mean Squared Error (*MSE) metric. Therefore, the received lossy video bitstream is decoded and a modified version of the H.264/AVC reference software version 12.3 [138] is used to extract the required bitstream information. As proof-of-concept, Naccari et al. further developed a reduced reference system that uses the assessed MSE values at macroblock level to calculate an estimate of the SSIM metric. Therefore, a feature extraction module in the H.264/AVC reference encoder computes mean value and standard deviation for each macroblock prior to the streaming process and distributes that information via an error free channel to monitoring nodes in the network. The monitoring nodes then use this information along with the information extracted from the decoded video stream to calculate an estimate of the SSIM metric. In comparison to our proposed objective monitoring approach, the precomputation step is less complex but the necessary decoding process in the network for each monitored video stream requires high computational expense. Hence, the reduced reference objective quality monitoring is the first candidate for the following comparison.

As second candidate, we assume an objective video quality metric that uses the measured RTP packet loss in the network to assess the video quality. As long as the measured RTP loss per GOP is below a predefined threshold $thres_{pl}$, the perceived video quality is rated as good and in case the RTP loss exceeds the threshold $thres_{pl}$, the video quality is rated as bad. In comparison to our monitoring, this approach requires proper thresholds for different kinds of content. In addition, the threshold is also dependent on the number of slices per frame as this directly influences the packet structure. Each slice is usually encapsulated in a single NALU and in case the NALU size exceeds the MTU of the network, the NALU is fragmented and encapsulated in several RTP packets. Another option is to encapsulate the NALU in a single RTP packet which is then fragmented on IP layer. Either way, the resilience against packet loss is different and hence, the number of slice and the encapsulation in RTP packets have to be considered for proper thresholds. Finding these thresholds can be seen as precomputation step which is less complex as it has to be done only once per content and encoding parameters like number of slices as well as frame and packet structure.

4.5.2 Evaluation Methods

The following evaluation again treats the full reference SSIM calculation as reference and compares our approach as well as the above described two other approaches against this reference. The comparison is twofold. First, we evaluate the correlation of the approximated SSIM values, calculated by our approach and the NORM approach, with the exact calculation of the SSIM metric. The packet loss-based classification does not yield approximated SSIM values and hence, it is not considered in this first evaluation. The second evaluation then comprises the classification of the video quality itself and here, we include all three approaches. For the mapping from SSIM to video quality, we assume the same threshold as introduced in Section 4.3.3. SSIM scores larger than 0.88 correspond to at least MOS 3 and hence good video quality. SSIM scores smaller than 0.88 result in bad video quality. Figure 4.13 shows the different steps during the evaluation as well

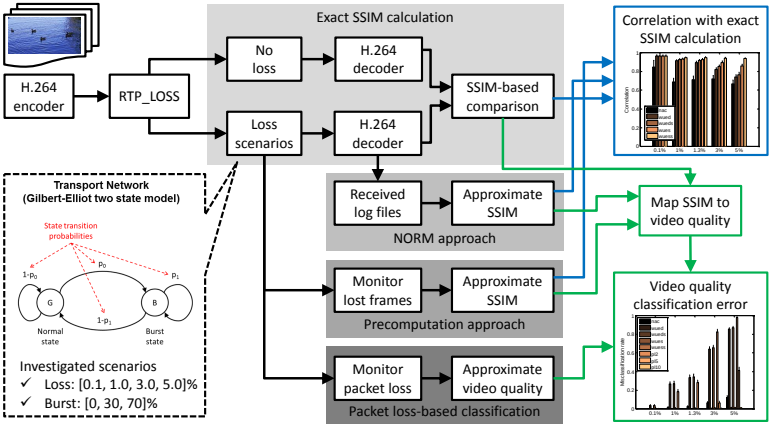


Figure 4.13: Flow chart showing the different steps during the evaluation.

as the used tools. First, we encode the considered videos with the H.264/AVC reference encoder using the same parameters as in [129]. Besides the high definition content, we also include the low resolution sequences *Foreman* and *Soccer* as they were also used by Naccari et al. in [129]. The encoder output is already RTP packetized, as specified in [118]. In the next step, we generate RTP packet loss by applying the Gilbert-Elliot two state model [139, 140]. The model has two parameters p_0 and p_1 with which the total packet loss and the burstiness can be configured. The total packet loss can be calculated as $p = \frac{p_0}{p_0 - p_1 + 1}$. We include total packet loss rates of $[0.1, 1.0, 3.0, 5.0]\%$ and burst rates of $[0, 30, 70]\%$. Higher burst rates are not considered since the decoding process is only possible with the H.264/AVC reference decoder in case at least one slice per frame is received correctly. Higher burst rates in combination with high packet loss rates however would result in entire frame loss and hence cannot be evaluated. In the last step, the lossy and lossless video streams are decoded and the respective exact and approximated SSIM values as well as the resulting video quality are calculated and compared against the full reference SSIM approach. To compute the approx-

imated SSIM scores based on NORM, the log files produced during the decoding process of the lossy video bitstream by the modified H.264/AVC reference decoder are required. Also necessary precomputed data sets like mean and standard deviation per macroblock for NORM or the precomputed distortion values for our approach are not shown in the figure. Finally, the approximated SSIM values are mapped to the video quality. In addition, the estimated video quality due to the RTP packet loss-based approach is computed by mapping the measured RTP loss to the video quality. Both the SSIM values and the assessed video quality are calculated on a per GOP basis so that all three approaches are comparable since per GOP calculation is supported by all three approaches.

4.5.3 Discussion for SD Content

Subsequently, we discuss the results for the considered standard definition video content, i.e., Foreman in QCIF and CIF resolution as well as *Soccer* in 4CIF resolution. First, we study the correlation between the exact full reference SSIM calculation and the two to be compared monitoring approaches, i.e., the reduced reference approach by Naccari et al. and our precomputation-based approach. Table 4.3 presents the results for all three SD videos and the two different GOP structures, as introduced in Section 4.1.3. The column labeled *NAC* contains the results for the reduced reference approach by Naccari et al. and the column labeled *WUE* contains the results for our proposed monitoring mechanism. In the remainder, we consider a more sophisticated version of our approach, which is an upgrade to the modified version presented in Section 4.4.3.

This more advanced version was necessary as the former one was a pessimistic approach that did only consider loss on frame level. However, this results in far too low approximated SSIM values and hence, we modified our approach to consider loss on slice level. Each slice per frame is treated equally and we divide the precomputed distortion per frame by the number of slices to get the distortion per lost slice. The monitoring agents in the network then map RTP packet loss to slice and frame loss and computed the approximated SSIM values as presented

Scenario		IPP		IBP	
PLR	Burst	NAC	WUE	NAC	WUE
0.1	0	0.88	0.91	0.78	0.93
	30	0.95	0.94	0.84	0.96
	70	1.00	1.00	0.96	0.99
1.0	0	0.83	0.82	0.77	0.85
	30	0.84	0.88	0.78	0.83
	70	0.88	0.90	0.81	0.89
3.0	0	0.79	0.81	0.81	0.84
	30	0.88	0.87	0.83	0.82
	70	0.90	0.85	0.85	0.84
5.0	0	0.73	0.80	0.77	0.83
	30	0.86	0.87	0.83	0.86
	70	0.88	0.79	0.82	0.72

(a) Foreman in QCIF resolution.

Scenario		IPP		IBP	
PLR	Burst	NAC	WUE	NAC	WUE
0.1	0	0.91	0.96	0.79	0.94
	30	0.97	0.98	0.82	0.95
	70	1.00	1.00	0.96	0.99
1.0	0	0.90	0.78	0.80	0.86
	30	0.89	0.85	0.81	0.84
	70	0.89	0.90	0.84	0.91
3.0	0	0.91	0.79	0.82	0.84
	30	0.90	0.86	0.84	0.84
	70	0.90	0.86	0.84	0.83
5.0	0	0.89	0.80	0.81	0.83
	30	0.87	0.86	0.83	0.86
	70	0.88	0.80	0.77	0.72

(b) Foreman in CIF resolution.

Scenario		IPP		IBP	
PLR	Burst	NAC	WUE	NAC	WUE
0.1	0	0.85	0.90	0.73	0.93
	30	0.88	0.94	0.69	0.95
	70	0.52	0.97	0.48	0.95
1.0	0	0.80	0.89	0.70	0.88
	30	0.85	0.91	0.72	0.92
	70	0.82	0.86	0.63	0.84
3.0	0	0.77	0.87	0.67	0.87
	30	0.81	0.90	0.64	0.89
	70	0.82	0.80	0.61	0.62
5.0	0	0.75	0.87	0.66	0.87
	30	0.79	0.90	0.75	0.91
	70	0.80	0.71	0.71	0.48

(c) Soccer in 4CIF resolution.

Table 4.3: Correlation coefficients for standard definition content.

in Section 4.3.2. The correlation of this advanced version with the exact full reference calculation of the SSIM metric was much better than the correlation of the non-modified version and hence, we consider only this advanced version.

For the comparison between the NAC and WUE approach, we conducted 30 repetitions for each combination of RTP packet loss and burst rate to calculate mean correlation coefficients and confidence intervals. However, Table 4.3 only comprises the mean values as the confidence intervals were sufficiently small and always below 2%. Hence, for clarity reasons, we omit showing the upper and lower bound of the confidence intervals in the table.

Concerning the IPP structure, both approaches perform similar and the correlation coefficients for all three videos are in the most cases between 0.8 and 0.9. This is especially true for the *Foreman* video sequences in QCIF and CIF resolution. Overall for this sequence, the correlation of both approaches is getting lower for higher packet loss and burst rates. The NAC approach achieves a correlation of at least 0.73 whereas our WUE approach achieves at least 0.78. For the *Soccer* sequence in 4CIF resolution however, the observation is different. Our WUE approach nearly always achieves a significantly higher correlation than the NAC approach, except for the two higher burst scenarios with 3% and 5% packet loss.

Concerning the IBP structure, the observation is similar than for the IPP structure except that the overall correlation is a bit lower with the IBP structure. For the Foreman sequence in QCIF and CIF resolution, there are only subtle differences to the IPP structure and both monitoring approaches again perform similar. For the *Soccer* sequence, again our monitoring achieves the significantly higher correlation except for the high burst scenario with 5% packet loss. For this scenario, our monitoring achieves only a correlation of 0.48.

Comparing both approaches by means of the correlation gives a first understanding about how the two competing approaches perform but eventually, the classification of the video quality and possible misclassifications are important. Hence, we discuss the mean error rate of both approaches in the remainder. The error rate denotes the number of misclassified GOPs per repetition and in Table 4.4, we present the mean error rates over 30 repetitions. Again, the upper and

Scenario		IPP				IBP			
PLR	Burst	NAC	WUE	PL5	PL10	NAC	WUE	PL5	PL10
0.1	0	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	30	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	70	0.00	0.00	0.01	0.01	0.01	0.00	0.01	0.00
1.0	0	0.08	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	30	0.06	0.00	0.01	0.01	0.01	0.01	0.01	0.01
	70	0.04	0.01	0.06	0.03	0.04	0.01	0.05	0.02
3.0	0	0.20	0.01	0.06	0.01	0.03	0.00	0.06	0.00
	30	0.18	0.01	0.14	0.03	0.05	0.01	0.14	0.02
	70	0.10	0.03	0.20	0.08	0.07	0.05	0.20	0.10
5.0	0	0.30	0.03	0.41	0.02	0.07	0.01	0.42	0.01
	30	0.25	0.04	0.38	0.06	0.09	0.02	0.37	0.05
	70	0.15	0.06	0.38	0.15	0.12	0.08	0.37	0.16

(a) Foreman in QCIF resolution.

Scenario		IPP				IBP			
PLR	Burst	NAC	WUE	PL5	PL10	NAC	WUE	PL5	PL10
0.1	0	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	30	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	70	0.01	0.00	0.01	0.00	0.01	0.00	0.01	0.00
1.0	0	0.12	0.00	0.00	0.00	0.04	0.00	0.00	0.00
	30	0.09	0.01	0.01	0.01	0.03	0.01	0.01	0.01
	70	0.05	0.04	0.07	0.04	0.06	0.02	0.06	0.02
3.0	0	0.31	0.01	0.06	0.01	0.16	0.00	0.06	0.00
	30	0.26	0.04	0.14	0.04	0.15	0.02	0.14	0.02
	70	0.14	0.07	0.20	0.09	0.12	0.08	0.20	0.11
5.0	0	0.46	0.04	0.40	0.04	0.26	0.00	0.42	0.01
	30	0.32	0.09	0.36	0.09	0.25	0.04	0.37	0.05
	70	0.22	0.11	0.37	0.17	0.21	0.13	0.37	0.18

(b) Foreman in CIF resolution.

Scenario		IPP				IBP			
PLR	Burst	NAC	WUE	PL5	PL10	NAC	WUE	PL5	PL10
0.1	0	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	30	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	70	0.02	0.00	0.01	0.00	0.01	0.02	0.02	0.02
1.0	0	0.20	0.00	0.00	0.00	0.09	0.00	0.00	0.00
	30	0.16	0.01	0.02	0.01	0.09	0.00	0.01	0.00
	70	0.15	0.06	0.09	0.06	0.13	0.06	0.09	0.06
3.0	0	0.52	0.02	0.07	0.02	0.33	0.00	0.06	0.00
	30	0.41	0.06	0.14	0.06	0.28	0.02	0.13	0.02
	70	0.31	0.14	0.21	0.15	0.24	0.19	0.26	0.20
5.0	0	0.69	0.06	0.39	0.06	0.50	0.03	0.41	0.03
	30	0.55	0.13	0.35	0.13	0.40	0.09	0.36	0.10
	70	0.40	0.24	0.35	0.23	0.30	0.34	0.39	0.33

(c) Soccer in 4CIF resolution.

Table 4.4: Error rates for standard definition content.

lower bound of the confidence intervals are not shown as they were sufficiently small. In addition to the NAC and WUE approach, we also consider the RTP packet loss-based approach introduced earlier. As threshold for the RTP packet loss metric, we consider 5% and 10%. The corresponding results in Table 4.4 are denoted with PL5 and PL10 respectively.

Concerning the Foreman sequence in QCIF and CIF resolution and with the two different GOP structures, our monitoring significantly outperforms the other three approaches and achieves an error rate of at most 0.11, which means that at most 11% of the GOPs have been wrongly classified. The 5% threshold for the packet loss-based approach is not suitable for the Foreman sequence in the considered configurations and produces an error rate of at most 0.42 or 42%. However, the 10% threshold is much better and produces an error rate of at most 0.18 or 18%. The error rate achieved by the NAC approach is rather high for the considered Foreman sequences although the correlation with the exact SSIM metric was quite good. Hence, a high correlation with the exact full reference SSIM metric is no guarantee that the monitoring yields a proper video quality classification.

Concerning the *Soccer* sequence, again our monitoring performs very well except for the 70% burst scenarios with 3% and 5% packet loss. For those scenarios, the error rate lies between 0.14 and 0.34. The other three approaches are even worse and produce an even higher error rate. Nevertheless, the packet loss-based approach with 10% packet loss now performs comparable to our approach and achieves nearly the same error rates across all scenarios. A possible explanation is that we have chosen a fixed number of slices per frame and hence, a fixed number of RTP packets per frame. Since we consider loss on RTP layer, each lost packet automatically corresponds to a lost slice. Hence, the mapping from RTP packet loss to video quality is similar, but neglects the different slice types. If however packet loss on IP layer is considered, the mapping from packet loss to RTP and hence slice loss depends on the MTU of traversed transport networks. Larger slices are fragmented over several IP packets and it is not obvious anymore, how many slices are lost per GOP. So, finding proper thresholds in such

scenarios is cumbersome and hence, care has to be taken when considering the RTP packet loss-based approach.

Overall, concerning only the correlation with the exact full reference metric, both the NAC and our WUE approach perform very well for the standard definition content. However, considering also the video quality classification, our proposed WUE approach performs best while the packet loss-based approach is comparable as long as only loss on RTP layer and fixed number of slices are considered.

4.5.4 Discussion for HD Content

Besides the three test sequences with non high definition resolution in the previous section, we conducted the comparison also for the three sequences with high definition 720p resolution introduced in Section 4.1.3. The resulting correlation coefficients for the NAC and our WUE approach are shown in Table 4.5. In addition, the error rates for the two approaches and the RTP packet loss-based approach are shown in Table 4.6. Again, we present only mean values and omit again upper and lower bounds of the confidence intervals for clarity reasons.

Regarding the correlation of the NAC and WUE approach for the high definition content, the same reasoning holds as for the *Soccer* sequence with 4CIF resolution. Again, our WUE monitoring outperforms the NAC approach and achieves the higher correlation with the full reference SSIM metric. The overall correlation for our WUE approach lies in the range between 0.75 and 0.95 for both structures and all three videos. Only for the high burst scenario with 5% RTP packet loss, the correlation of our approach drops to values around 0.6. However, the NAC approach performs only slightly better in those cases.

Considering spatial and temporal information, our monitoring achieves a higher correlation with increasing spatial and temporal information. The *Old Town Cross* sequence has the lowest temporal and spatial information and for this sequence, our monitoring produces the lowest overall correlation. *Ducks Take Off* and *Park Joy* have medium and high amounts of information respectively and for

Scenario		IPP		IBP	
PLR	Burst	NAC	WUE	NAC	WUE
0.1	0	0.94	0.97	0.84	0.96
	30	0.95	0.98	0.84	0.97
	70	0.99	0.99	0.95	0.98
1.0	0	0.84	0.96	0.69	0.95
	30	0.80	0.96	0.72	0.96
	70	0.86	0.91	0.70	0.89
3.0	0	0.81	0.97	0.72	0.94
	30	0.79	0.97	0.65	0.95
	70	0.77	0.78	0.64	0.73
5.0	0	0.80	0.97	0.66	0.94
	30	0.79	0.97	0.65	0.95
	70	0.72	0.68	0.55	0.61

(a) Ducks Take Off in 720p resolution.

Scenario		IPP		IBP	
PLR	Burst	NAC	WUE	NAC	WUE
0.1	0	0.90	0.79	0.82	0.84
	30	0.93	0.91	0.83	0.89
	70	0.98	0.96	0.89	0.88
1.0	0	0.65	0.72	0.77	0.77
	30	0.57	0.73	0.79	0.84
	70	0.65	0.82	0.67	0.74
3.0	0	0.67	0.74	0.75	0.75
	30	0.67	0.79	0.70	0.80
	70	0.68	0.74	0.61	0.67
5.0	0	0.68	0.76	0.74	0.74
	30	0.67	0.80	0.62	0.78
	70	0.67	0.66	0.55	0.54

(b) Old Town Cross in 720p resolution.

Scenario		IPP		IBP	
PLR	Burst	NAC	WUE	NAC	WUE
0.1	0	0.94	0.95	0.83	0.95
	30	0.95	0.98	0.79	0.97
	70	0.96	0.99	0.93	0.97
1.0	0	0.74	0.91	0.70	0.94
	30	0.73	0.93	0.71	0.95
	70	0.69	0.92	0.69	0.88
3.0	0	0.70	0.90	0.67	0.92
	30	0.65	0.92	0.67	0.94
	70	0.60	0.81	0.55	0.71
5.0	0	0.66	0.90	0.61	0.91
	30	0.62	0.91	0.62	0.94
	70	0.54	0.75	0.49	0.58

(c) Park Joy in 720p resolution.

Table 4.5: Correlation coefficients for high definition content.

these sequences, our monitoring achieves a higher correlation. For the NAC approach, the amount of spatial and temporal information has only minor influence and the performance is not significantly influenced by the different types of video content.

Concerning the error rates for the high definition content, we again see the same behavior as for the *Soccer* sequence in 4CIF resolution. Our monitoring achieves lower overall error rates than compared to the NAC approach while the RTP packet loss-based approach with 10% threshold performs comparable to our approach. However, the RTP packet loss-based approach has the same problem as stated earlier. The correctness of the video quality classification highly depends on the mapping from occurred loss on IP or RTP layer to the number of lost slices. As far as only loss on RTP layer is considered, the mapping from RTP packet loss to lost slices is always the same and usually, a lost RTP packet corresponds to a lost slice. Hence, according to the shown results, losing more than 10% of the slices per GOP results in bad video quality. If however loss on IP layer is considered, the mapping from IP packet loss to slice loss is not fixed anymore and depends on the MTU of the underlying network and possible fragmentation on IP or RTP layer. Finding suitable thresholds for loss on IP layer is hence cumbersome as it requires knowledge about the path properties between video streaming source and receiver. Our monitoring in contrast is agnostic to whether loss on IP or RTP layer is considered as the monitoring agents dynamically map IP or RTP packet loss to slice loss and hence adapt to the underlying network.

Overall, we have shown that our approach achieves the highest correlation with the exact SSIM metric and produces the lowest error rates with respect to the classification of the video quality. The approach proposed by Naccari et al. performs similar for the low resolution content but for higher resolutions, the correlation is significantly lower and the error rates are higher. The RTP packet loss-based approach produces similar error rates compared to our approach but the correctness is highly dependent on the used loss thresholds. These thresholds are influenced by encoding parameters like number of slices and network parameters that change the mapping from slice to IP packets. Hence, finding suitable

Scenario		IPP				IBP			
PLR	Burst	NAC	WUE	PL5	PL10	NAC	WUE	PL5	PL10
0.1	0	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	30	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	70	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.01
1.0	0	0.08	0.00	0.00	0.00	0.02	0.00	0.00	0.00
	30	0.05	0.01	0.01	0.01	0.02	0.00	0.00	0.00
	70	0.04	0.04	0.08	0.04	0.06	0.04	0.07	0.04
3.0	0	0.22	0.01	0.06	0.01	0.06	0.00	0.06	0.00
	30	0.14	0.05	0.13	0.05	0.07	0.03	0.12	0.03
	70	0.07	0.10	0.20	0.11	0.16	0.15	0.23	0.16
5.0	0	0.32	0.05	0.40	0.05	0.12	0.01	0.41	0.01
	30	0.18	0.10	0.35	0.10	0.13	0.06	0.36	0.07
	70	0.12	0.17	0.34	0.17	0.24	0.26	0.34	0.25

(a) Ducks Take Off in 720p resolution.

Scenario		IPP				IBP			
PLR	Burst	NAC	WUE	PL5	PL10	NAC	WUE	PL5	PL10
0.1	0	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	30	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00
	70	0.00	0.00	0.00	0.00	0.02	0.01	0.02	0.01
1.0	0	0.08	0.00	0.00	0.00	0.04	0.00	0.00	0.00
	30	0.05	0.00	0.01	0.00	0.03	0.01	0.01	0.01
	70	0.02	0.01	0.05	0.02	0.05	0.04	0.07	0.04
3.0	0	0.22	0.00	0.06	0.00	0.16	0.00	0.06	0.00
	30	0.16	0.02	0.13	0.03	0.12	0.03	0.14	0.03
	70	0.06	0.04	0.20	0.06	0.15	0.14	0.24	0.16
5.0	0	0.34	0.01	0.42	0.02	0.24	0.01	0.41	0.02
	30	0.23	0.04	0.40	0.06	0.20	0.05	0.37	0.07
	70	0.12	0.07	0.38	0.13	0.25	0.23	0.38	0.25

(b) Old Town Cross in 720p resolution.

Scenario		IPP				IBP			
PLR	Burst	NAC	WUE	PL5	PL10	NAC	WUE	PL5	PL10
0.1	0	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	30	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	70	0.01	0.01	0.00	0.01	0.01	0.01	0.02	0.01
1.0	0	0.12	0.00	0.00	0.00	0.08	0.00	0.00	0.00
	30	0.09	0.01	0.01	0.01	0.07	0.01	0.01	0.01
	70	0.09	0.08	0.08	0.07	0.10	0.06	0.09	0.06
3.0	0	0.35	0.03	0.06	0.03	0.21	0.02	0.07	0.02
	30	0.28	0.07	0.12	0.06	0.19	0.06	0.13	0.06
	70	0.21	0.18	0.17	0.16	0.21	0.21	0.25	0.21
5.0	0	0.50	0.13	0.34	0.13	0.31	0.07	0.37	0.07
	30	0.37	0.19	0.29	0.17	0.27	0.13	0.32	0.14
	70	0.28	0.30	0.26	0.22	0.28	0.33	0.34	0.30

(c) Park Joy in 720p resolution.

Table 4.6: Error rates for high definition content.

thresholds is cumbersome and not always possible in advance. Our monitoring however does not experience such problems since the monitoring agents dynamically map lost IP packets to lost slices. However, one observed drawback of our monitoring is the bad performance for scenarios with high burst and high packet loss. For those scenarios, there is a lot of damage per frame due to sequences of lost packets. A possible explanation is that the H.264/AVC reference decoder does not allow entire frame loss which negatively influences the precomputed distortion values. However, this decoder is strongly required since it produces the necessary log files for the NAC approach. Hence, the observed drawback is inherent to this evaluation and not a problem of our proposed approach since for example the Mencoder decoder used in Section 4.4 is able to decode H.264/AVC video although entire frames are lost.

4.6 Lessons Learned

The objective of this chapter was to evaluate our proposed network-based video monitoring, which infers the user-perceived quality from packet loss measured in the network. The innovative concept of our approach is to precompute distortion values for certain loss scenarios. Due to high complexity for computing all frame loss possibilities within a GOP, we introduced a less complex algorithm that computes the distortion of multiple frame losses within a GOP based on the distortion of single frame losses.

To show the viability of our proposed solution, we conducted an exhaustive frame-based evaluation of our approach with respect to high definition video content. The evaluation includes different video sequences corresponding to different amounts of spatial and temporal information as well as different GOP structures corresponding to different IP streaming service providers. We have shown that for the given video configurations, losing more than five frames per GOP results in constantly bad video quality whereas video sequences with low temporal and spatial information encoded with B-frames are more robust against frame loss and require at least eight lost frames per GOP. This knowledge was used to re-

duce the evaluation overhead by excluding scenarios with constantly bad video quality. The results of the evaluation indicate an accuracy of more than 96% correctly classified GOPs of the proposed approximative distortion computation as compared to the correct values. At the same time, the number of required computations is significantly reduced since only single frame losses within a GOP have to be computed.

Further, we compared our monitoring for different RTP loss scenarios with another approach that uses information from the decoded video bitstream to approximate the SSIM metric. We have learned that the frame-based version of our mechanism was far to pessimistic when considering loss on RTP layer and that a slice-based version is necessary to improve the correlation with the exact SSIM metric. According to our results, this slice-based version then achieves an overall correlation with the exact SSIM metric between about 0.75 and 0.95 and significantly outperforms the decoding-based approach, especially for high definition content. For standard definition content, the difference is smaller and both approaches perform similar. For the discussion of the estimated user-perceived quality, we also included an RTP packet loss-based approach that uses predefined loss thresholds per GOP to approximate the video quality. For low and medium burst scenarios, our approach correctly classifies between 90% and 100% of the GOPs. For high burst scenarios however, our approach classifies only 70% to 80% of the GOPs correctly. The weaker performance for high burst scenarios is caused by the chosen decoder, which does not decode sequences where entire frames are lost. This negatively influences our precomputed distortions and hence, the approximated SSIM values. Overall, our approach still achieves the lowest error rates whereas the loss-based approach performs similar for standard and high definition content. However, the considered configuration with fixed number of slices and loss on RTP layer biases the results towards the loss-based approach since a lost packet always corresponds to a lost slice. Overall, our precomputation-based monitoring approach provides an accurate and scalable estimation of the user-perceived video quality since the proposed design enables content and codec unaware monitoring agents in the network.

5 Conclusion

The future Internet will not consist of a single network that provides the underlying environment for heterogeneous services with differing requirements. More likely, a stable physical substrate serves as necessary basis to virtual networks that are tailored to the needs of application groups with overlapping requirements. These logical networks are highly flexible, since virtual resources can be added and removed on demand, depending on for example the current load situation. Further, *Application-Aware Networking (*AAN) enables dynamic adaptations of the network environment to satisfy the current needs of hosted applications. This approach is in contrast to traditional network configuration or provisioning methods. A prominent use case that benefits from these extended capabilities of the network is denoted with service component mobility. Services hosted on *Virtual Machines (*VMs) follow their consuming mobile endpoints, so that access latency as well as consumed network resources are reduced. Especially for applications like video streaming, which consume a large fraction of the available resources, is this an important means to relieve the resource constraints and eventually provide better service quality.

This monograph addresses challenges occurring in a future virtualized architecture that accommodates dynamic virtual networks facilitating AAN. In particular, these challenges are discussed considering a video streaming scenario with service component as well as endpoint mobility.

Service component mobility is usually achieved by migrating the hosting VM between different locations of the underlying data center. The migration process involves a downtime during which the service is not available. With traditional approaches, migration processes are performed concurrently and the available

bandwidth is shared among all transmissions. This however leads to a longer downtime since the available bandwidth highly influences the downtime. To reduce this downtime, we develop an optimized architecture that utilizes the delay-tolerant nature of planned migration processes. The architecture comprises a subscription model and a scheduling mechanism that respects the individual transmission profiles of customers. We show the viability of our approach by conducting a simulative comparison with the traditional approach that transmits migration requests concurrently. With our proposed architecture, the transmission time is significantly lower than with the traditional concurrent approach while the subscription model ensures the individual profiles of the customers. Overall, the presented approach enables network providers to offer innovative high-speed services to customers based on its flexible network infrastructure without giving away the control over its network.

Concerning the network requirements of service component mobility, both source and destination node have to belong to the same broadcast domain, so that the hypervisor at the destination node is able to inform intermediate Ethernet nodes about the new location of the migrated VM. As a result, a large number of Ethernet nodes are part of the same Ethernet network, which leads to scalability challenges regarding the address resolution process. To quantify the amount of broadcast messages for various data center connect mechanisms, we introduce an *Address Resolution Protocol (*ARP) traffic model. This model is applied to two different interconnect solutions that implement contrary approaches, i.e., *Overlay Transport Virtualization (*OTV) and *Virtual Private LAN Services (*VPLS). According to our results, both solutions result in similar signaling overhead for medium load scenarios while VPLS produces less signaling overhead for low and high load scenarios. However, this advantage of VPLS is only partly important since the actual difference is on a low level. Instead, other features like initial configuration effort or maintenance overhead seem more important than the small difference with respect to address resolution scalability. In general, the proposed ARP model is not limited to the considered scenario since it can be adapted to estimate the ARP traffic also for other scenarios.

Besides service component mobility, endpoint mobility is another important aspect in the discussed scenario and we focus on the *Locator/ID Separation Protocol (*LISP) and its mobility extension *LISP Mobile Node (*LISP-MN) as enabling technologies. In principle, LISP-MN supports mobile nodes but causes increased communication delay, packet overhead, and path stretch as compared to the basic LISP architecture. We address these shortcomings by proposing several extensions related to the control and data plane of LISP-MN. In addition, a NAT traversal mechanism is designed that restores connectivity for mobile nodes behind NAT gateways. Furthermore, we present our developed simulation framework that can be used to evaluate the LISP performance for various network configurations. In particular, we assess the handover performance of LISP-MN regarding a mobile endpoint that receives a video stream while roaming between different types of access domains. Among others, these results provide valuable insights in the to be expected LISP performance and support the decision for the best suitable handover mechanism in the considered scenario.

Service and endpoint mobility both allow an adaptation of the used paths between an offered service, i.e., video streaming and the consuming users in case the service quality drops due to network problems. To make evidence-based adaptations in case of quality drops, a scalable monitoring component is required that is able to monitor the service quality for video streaming applications with reliable accuracy. To address that, we propose a network-based video quality monitoring solution that infers the user-perceived quality from packet loss measured in the network. The innovative concept is to precompute distortion values for certain loss scenarios. This information is then used by monitoring nodes to estimate the video quality in the network. We conduct an exhaustive evaluation and the results on frame level indicate an accuracy of more than 96% correctly classified GOPs. For slice or RTP level, we compare our approach with contrary approaches to prove the viability of our idea. Our results show that our proposed mechanism achieves the highest percentage of correctly classified GOPs.

In the course of this monograph, we discussed challenges that arise in a future virtualized network architecture and presented developed solutions. In particular,

we considered a use case that involves service component as well as endpoint mobility for a video streaming scenario. The discussed challenges provide valuable insights for video streaming providers and the introduced solutions cover the different layers of virtualized architectures as well as the entire service chain with respect to video streaming in mobile environments. Overall, the proposed solutions constitute important building blocks towards a future virtualized environment hosting video streaming specific networks for mobile endpoints.

Acronyms

4WARD	Architecture and Design for the Future Internet
AAN	Application-Aware Networking
ALG	Application Layer Gateway
ARP	Address Resolution Protocol
ASM	Any Source Multicast
ASP	Application Service Provider
AVC	Advanced Video Coding
CN	Correspondent Node
COMCON	COntrol and Management of COexisting Networks
DCI	Data Center Interconnect
ED	Edge Device
EID	Endpoint Identifier
EoMPLS	Ethernet over MultiProtocol Label Switching
ETR	Egress Tunnel Router
E-VPN	Ethernet Virtual Private Network
FP7	Seventh Framework Program
GLI-Split	Global Locator, Local Locator, and Identifier Split
GMPLS	Generalized MultiProtocol Label Switching
GOP	Group of Pictures
HA	Home Agent
HIP	Host Identity Protocol
ID	identifier
IETF	Internet Engineering Task Force
IH	Inner Header

ILNP	Identifier Locator Network Protocol
IRTF	Internet Research Task Force
IS-IS	Intermediate System to Intermediate System
ITR	Ingress Tunnel Router
Ivip	Internet Vastly Improved Plumbing
LISP	Locator/ID Separation Protocol
LISP-MN	LISP Mobile Node
LISP+ALT	LISP Alternative Topology
LISP-IW	LISP Interworking
LLOC	Local Locator
LOC	Locator
LOC/ID Split	Locator/Identifier Split
LSP	Link State Packet
MB	Macroblock
MBGP	Multiprotocol Border Gateway Protocol
MIPv6	Mobility Support in IPv6
MN	Mobile Node
MOS	Mean Opinion Score
MPLS	Multiprotocol Label Switching
MR	Map Resolver
MS	Map Server
MSE	Mean Squared Error
MTU	Maximum Transfer Unit
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
NAS	Network Attached Storage
NEMO	Network Mobility
NTR	NAT Traversal Router
NV	Network Virtualization
NVGRE	Network Virtualization using Generic Route Encapsulation
OH	Outer Header

OTV	Overlay Transport Virtualization
PE	Provider Edge
PETR	Proxy-ETR
PIP	Physical Infrastructure Provider
PITR	Proxy-ITR
PMIPv6	Proxy-MIPv6
RLOC	Routing Locator
RRG	Routing Research Group
RTP	Real-time Transport Protocol
SDN	Software-Defined Networking
SLA	Service Level Agreement
SMR	Solicit Map-Request
SN	Stationary Node
SSIM	Structural SIMilarity
VCL	Video Coding Layer
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Monitor
VNO	Virtual Network Operator
VNP	Virtual Network Provider
VoD	Video-on-Demand
VPLS	Virtual Private LAN Services
VQA	Video Quality Assessment
VXLAN	Virtual eXtensible Local Area Network
WAN	Wide Area Network

Bibliography and References

— Bibliography of the Author —

— Journal Papers —

[1] M. Menth, M. Hartmann, P. Tran-Gia, , and D. Klein, "Future Internet Routing: Motivation and Design Issues," *it - Information Technology*, vol. 50(6), Dec. 2008.

[2] M. Menth, M. Hartmann, and D. Klein, "Global Locator, Local Locator, and Identifier Split (GLI-Split)," *Future Internet*, vol. 5, Mar. 2013.

[3] D. Klein, P. Tran-Gia, and M. Hartmann, "Big data," *Informatik-Spektrum*, pp. 1–5, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00287-013-0702-3

— Conference Papers —

[4] D. Klein, M. Menth, R. Pries, P. Tran-Gia, M. Scharf, and M. Söllner, "A Subscription Model for Time-Scheduled Data Transfers," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011)*, Dublin, Ireland, May 2011.

[5] D. Klein, R. Pries, M. Scharf, M. Söllner, and M. Menth, "Modeling and Evaluation of Address Resolution Scalability in VPLS," in *IEEE ICC 2012 - Next-Generation Networking Symposium*, Ottawa, Canada, Jun. 2012.

[6] M. Menth, D. Klein, and M. Hartmann, "Improvements to LISP Mobile Node," in *22nd International Teletraffic Congress (ITC)*, Sep. 2010, pp. 1 –8.

[7] D. Klein, M. Hartmann, and M. Menth, "NAT traversal for LISP mobile node," in *Proceedings of the Re-Architecting the Internet Workshop*, ser. ReARCH '10. Philadelphia, USA: ACM, 2010, pp. 8:1–8:6.

[8] ——, "NAT Traversal for LISP Mobile Node," IETF Internet-Draft, work in progress, Jul. 2010. [Online]. Available: http://tools.ietf.org/html/draft-klein-lisp-mn-nat-traversal

[9] D. Klein, M. Höfling, M. Hartmann, and M. Menth, "Integrating LISP and LISP-MN into INET," in *5th International Workshop on OMNeT++*, Desenzano, Italy, Mar. 2012.

[10] D. Klein and M. Jarschel, "An OpenFlow Extension for the OMNeT++ INET Framework," in *6th International Workshop on OMNeT++*, Cannes, France, Mar. 2013.

[11] T. Zinner, D. Klein, K. Tutschku, T. Zseby, P. Tran-Gia, and Y. Shavitt, "Performance of Concurrent Multipath Transmissions - Measurements and Model Validation," in *Proceedings of the 7th Conference on Next Generation Internet Networks (NGI)*, Kaiserslautern, Germany, Jun. 2011.

[12] T. Zinner, D. Klein, and T. Hoßfeld, "User-Centric Network-Application Interaction for Live HD Video Streaming," in *4th International Conference on Mobile Networks and Management (MONAMI 2012)*, Hamburg, Germany, Sep. 2012.

[13] D. Klein, T. Zinner, S. Lange, V. Singeorzan, and M. Schmid, "Video Quality Monitoring based on Precomputed Frame Distortions," in *IFIP/IEEE International Workshop on Quality of Experience Centric Management (QCMan)*, Ghent, Belgium, May 2013.

[14] D. Klein, T. Zinner, K. Borchert, S. Lange, V. Singeorzan, and M. Schmid, “Evaluation of Video Quality Monitoring based on Precomputed Frame Distortions,” in *19th EUNICE Workshop on Advances in Communication Networking*, Chemnitz, Germany, Aug. 2013.

— Software Demonstrations —

[15] M. Menth, M. Hartmann, and D. Klein, “Global Locator, Local Locator, and Identifier Split (GLI-Split),” 9th Wuerzburg Workshop on IP: ITG Workshop "Visions of Future Generation Networks" (EuroView2009),, Wuerzburg, Germany, Jul. 2009.

[16] D. Klein, M. Hartmann, M. Höfling, and M. Menth, “Improvements to LISP Mobile Node Including NAT Traversal,” 10th Wuerzburg Workshop on IP: ITG Workshop "Visions of Future Generation Networks" (EuroView2010),, Wuerzburg, Germany, Aug. 2010.

[17] T. Zinner, D. Klein, S. Meier, D. Wagner, M. Hoffmann, W. Kiess, V. Singorzan, and M. Schmid, “Dynamic Topology Adaptation enabled by Network Virtualization: A Use-Case for the Future Internet,” 12th Wuerzburg Workshop on IP: ITG Workshop "Visions of Future Generation Networks" (EuroView2012), Jul. 2012.

— General References —

[18] P. Tran-Gia, “G-Lab: A Future Generation Internet Research Platform,” 2008. [Online]. Available: http://www.german-lab.de/

[19] “COMCON - Control and Management of Coexisting Networks,” 2011. [Online]. Available: http://www.german-lab.de/phase-2/comcon/

[20] D. Schlosser, M. Hoffmann, T. Hoßfeld, M. Jarschel, A. Kirstaedter, W. Kellerer, and S. Köhler, “COMCON: Use Cases for Virtual Future Networks,” in *TridentCom 2010*, Berlin, May 2010.

[21] D. Schlosser, M. Jarschel, M. Duelli, T. Hoßfeld, K. Hoffmann, M. Hoffmann, H. J. Morper, D. Jurca, and A. Khan, "A Use Case Driven Approach to Network Virtualization," in *accepted at IEEE Kaleidoscope 2010, published via OPUS Wuerzburg under OpenAccess*, Wuerzburg, Dec. 2010.

[22] D. Erickson, G. Gibb, B. Heller, D. Underhill, J. Naous, G. Appenzeller, G. Parulkar, N. McKeown, M. Rosenblum, M. Lam, S. Kumar, V. Alaria, P. Monclus, F. Bonomi, J. Tourrilhes, P. Yalagandula, S. Banerjee, C. Clark, and R. McGeer, "A Demonstration of Virtual Machine Mobility in an OpenFlow network," in *ACM SIGCOMM*, Seattle, Washington, 2008.

[23] M. Soellner, P. Schefczik, P. Bertin, G. Wei, X. Zhang, T.-M.-T. Nguyen, J. Mäkelä, T. Rautio, O. Mämmelä, S. Pérez, A. Eriksson, A.-M. Biraghi, C. Foley, M. P. de Leon, C. Dannewitz, T. Biermann, and M. Marchisio, "Mobility in the Future Internet: the 4WARD Innovations (White Paper)," Jun. 2010.

[24] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, A. Wundsam, M. Kind, O. Maennel, and L. Mathy, "Network virtualization architecture: proposal and initial prototype," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, ser. VISA '09. New York, NY, USA: ACM, 2009, pp. 63–72. [Online]. Available: http://doi.acm.org/10.1145/1592648.1592659

[25] Cisco Systems, "Cisco Visual Networking Index: Forecast and Methodology, 2011-2016," June 2012. [Online]. Available: http://www.cisco.com/en/US/netsol/ns827/networking_solutions_solution_category.html

[26] M. Duelli, S. Meier, D. Wagner, T. Zinner, M. Schmid, M. Hoffmann, and W. Kiess, "Experimental Demonstration of Network Virtualization and Resource Flexibility in the COMCON Project," in *8th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, Thessaloniki, Jun. 2012.

[27] S. Meier, M. Barisch, A. Kirstädter, D. Schlosser, M. Duelli, M. Jarschel, T. Hoßfeld, K. Hoffmann, M. Hoffmann, W. Kellerer, A. Khan, D. Jurca, and K. Kozu, "Provisioning and Operation of Virtual Networks," *Electronic Communications of the EASST, Kommunikation in Verteilten Systemen 2011*, vol. 37, Mar. 2011.

[28] ICT Data and Statistics Division within the Telecommunication Development Bureau of ITU, "Measuring the Information Society 2012," 2012. [Online]. Available: http://www.itu.int/ITU-D/ict/publications/idi/index.html

[29] T. Narten, M. Karir, and I. Foo, "Address Resolution Problems in Large Data Center Networks," RFC 6820 (Informational), Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/rfc6820

[30] L. Dunbar, S. Hares, M. Sridharan, N. Venkataramaiah, and B. Schliesser, "Address Resolution for Large Data Center Problem Statement," IETF Internet-Draft, work in progress, Mar. 2011. [Online]. Available: http://tools.ietf.org/html/draft-dunbar-armd-problem-statement

[31] Y. Li, "Problem Statement on Address Resolution in Virtual Machine Migration," IETF Internet-Draft, work in progress, Mar. 2011. [Online]. Available: http://tools.ietf.org/html/draft-liyz-armd-vm-migration-ps

[32] IETF Working Group , "Address Resolution for Massive numbers of hosts in Data center (Active WG)," Oct. 2011. [Online]. Available: http://tools.ietf.org/wg/armd/

[33] P. Knight and C. Lewis, "Layer 2 and 3 Virtual Private Networks: Taxonomy, Technology, and Standardization Efforts," *IEEE Communications Magazine*, vol. 42, no. 6, pp. 124 – 131, Jun. 2004.

[34] A. Sajassi, R. Aggarwal, W. Henderickx, A. Isaac, J. Uttaro, N. Bitar, S. Boutros, K. Patel, S. Salam, J.Drake, and R. Shekhar, "BGP MPLS

Based Ethernet VPN," IETF Internet-Draft, work in progress, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-ietf-l2vpn-evpn

[35] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," RFC 4984 (Informational), Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://tools.ietf.org/html/rfc4984

[36] T. Li, "Recommendation for a Routing Architecture," RFC 6115 (Informational), Internet Engineering Task Force, Feb. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6115

[37] Internet Research Task Force (IRTF), "Routing Research Group (RRG)." [Online]. Available: http://trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup

[38] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," RFC 6830 (Experimental), Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/rfc6830

[39] D. Farinacci, D. Lewis, D. Meyer, and C. White, "LISP Mobile Node," IETF Internet-Draft, work in progress, Oct. 2012. [Online]. Available: http://tools.ietf.org/html/draft-meyer-lisp-mn

[40] Cisco Systems, "Locator/ID Separation Protocol (LISP) Virtual Machine Mobility Solution," White Paper, 2011. [Online]. Available: http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9402/white_paper_c11-693627.pdf

[41] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger, "From Packets to People: Quality of Experience as New Measurement Challenge," in *Data Traffic Monitoring and Analysis: From measurement, classification and anomaly detection to Quality of experience*, ser. Lecture Notes in Computer Science, M. M. Ernst Biersack, Christian Callegari, Ed.

Springer Berlin Heidelberg, 2012, vol. 7754, pp. 219–263. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36784-7_10

[42] M. Lasserre and V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling," RFC 4762 (Proposed Standard), Internet Engineering Task Force, Jan. 2007. [Online]. Available: http://tools.ietf.org/html/rfc4762

[43] H. Grover, D.Rao, and D. Farinacci, "Overlay Transport Virtualization," IETF Internet-Draft, work in progress, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-hasmit-otv

[44] Cisco Systems, "Cisco Overlay Transport Virtualization Technology Introduction and Deployment Considerations," White Paper, Jan. 2011. [Online]. Available: http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DCI/whitepaper/DCI_1.html

[45] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.

[46] F. Checconi, T. Cucinotta, and M. Stein, "Real-Time Issues in Live Migration of Virtual Machines," 4th Workshop on Virtualization in High Performance Computing (VHPC'09), Delft, The Netherlands, Sep. 2009.

[47] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 337–350. [Online]. Available: http://dl.acm.org/citation.cfm?id=1387589.1387613

[48] M. Jarschel and R. Pries, "An OpenFlow-Based Energy-Efficient Data Center Approach," in *ACM SIGCOMM*, Helsinki, Finland, Aug. 2012.

[49] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schiöberg, "Live wide-area migration of virtual machines including local persistent state," in *Proceedings of the 3rd international conference on Virtual execution environments*, ser. VEE '07. New York, NY, USA: ACM, 2007, pp. 169–179. [Online]. Available: http://doi.acm.org/10.1145/1254810.1254834

[50] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251203.1251223

[51] A. Stage and T. Setzer, "Network-aware migration control and scheduling of differentiated virtual machine workloads," in *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, ser. CLOUD '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 9–14. [Online]. Available: http://dx.doi.org/10.1109/CLOUD.2009.5071527

[52] S. Akoush, R. Sohan, A. Rice, A. Moore, and A. Hopper, "Predicting the Performance of Virtual Machine Migration," in *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, 2010, pp. 37–46.

[53] S. Cheshire, "IPv4 Address Conflict Detection," RFC 5227 (Proposed Standard), Internet Engineering Task Force, Jul. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5227

[54] A. Voyiatzis, "A Survey of Delay- and Disruption-Tolerant Networking Applications," *Journal of Internet Engineering*, vol. 5, no. 1, May 2012.

[55] J. Zheng and H. T. Mouftah, "Routing and Wavelength Assignment for Advance Reservation in Wavelength-Routed WDM Optical Networks," in

IEEE International Conference on Communications (ICC), vol. 5, Aug. 2002, pp. 2722 – 2726.

[56] L.-O. Burchard, H.-U. Heiss, and C. A. F. De Rose, "Performance Issues of Bandwidth Reservations for Grid Computing," in *Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. Washington, DC, USA: IEEE Computer Society, 2003, p. 82.

[57] U. Farooq, S. Majumdar, and E. Parsons, "Dynamic scheduling of lightpaths in lambda grids," in *IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS)*, Oct. 2005, pp. 1463 –1472 Vol. 2.

[58] E. He, X. Wang, and J. Leighton, "A Flexible Advance Reservation Model for Multi-Domain WDM Optical Networks," in *IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS)*, San Jose, CA, 2006, pp. 1 – 10.

[59] B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, "Data Management and Transfer in High-Performance Computational Grid Environments," *Parallel Computation Journal*, vol. 28, pp. 749–771, 2001.

[60] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," in *International Conference on Network and Parallel Computing (IFIP)*, vol. 21, no. 4, Jul. 2006, pp. 513–520.

[61] J. Zheng, B. Zhang, and H. Mouftah, "Toward automated provisioning of advance reservation service in next-generation optical internet," *IEEE Communications Magazine*, vol. 44, no. 12, pp. 68–74, 2006.

[62] T. Lehman, J. Sobieski, and B. Jabbari, "DRAGON: a framework for service provisioning in heterogeneous grid networks," *IEEE Communications Magazine*, vol. 44, no. 3, pp. 84 – 90, Mar. 2006.

[63] L.-O. Burchard, "Networks with Advance Reservations: Applications, Architecture, and Performance," *Journal of Network and Systems Management*, vol. 13, pp. 429–449, 2005, 10.1007/s10922-005-9004-7. [Online]. Available: http://dx.doi.org/10.1007/s10922-005-9004-7

[64] S. Bryant and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture," RFC 3985 (Informational), Internet Engineering Task Force, Mar. 2005, updated by RFC 5462. [Online]. Available: http://tools.ietf.org/html/rfc3985

[65] L. Martini, E. Rosen, N. El-Aawar, and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks," RFC 4448 (Proposed Standard), Internet Engineering Task Force, Apr. 2006, updated by RFC 5462. [Online]. Available: http://tools.ietf.org/html/rfc4448

[66] K. Kompella and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling," RFC 4761 (Proposed Standard), Internet Engineering Task Force, Jan. 2007, updated by RFC 5462. [Online]. Available: http://tools.ietf.org/html/rfc4761

[67] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks," IETF Internet-Draft, work in progress, May 2013. [Online]. Available: http://tools.ietf.org/html/draft-mahalingam-dutt-dcops-vxlan

[68] M. Sridharan, A. Greenberg, N. Venkataramiah, Y. Wang, K. Duda, I. Ganga, G. Lin, M. Pearson, P. Thaler, and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation," IETF Internet-Draft, work in progress, Feb. 2013. [Online]. Available: http://tools.ietf.org/html/draft-sridharan-virtualization-nvgre

[69] W. Imajuku, E. Oki, R. Papneja, S. Morishita, K. Ogaki, M. Miyazawa, K. Miyazaki, H. Nakazato, H. Sugiyama, J. Allen, S. Hasegawa, N. Sakuraba, I. Nishioka, S. Seno, Y. Nakahira, D. Ishii, S. Okamoto,

T. Unen, M. Blumhardt, H. Rakotoranto, and V. Pandian, "A multi-area MPLS/GMPLS interoperability trial over ROADM/OXC network," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 168–175, 2009.

[70] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, 1993.

[71] L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks," *ACM SIGCOMM Computer Communications Review*, vol. 20, no. 4, pp. 19–29, Aug. 1990. [Online]. Available: http://doi.acm.org/10.1145/99517.99525

[72] L. Kleinrock, *Queueing Systems*, 1st ed. New York: John Wiley and Sons, 1976, vol. 2: Computer Applications.

[73] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 60:1–60:10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1416222.1416290

[74] D. Oran, "OSI IS-IS Intra-domain Routing Protocol," RFC 1142 (Informational), Internet Engineering Task Force, Feb. 1990. [Online]. Available: http://tools.ietf.org/html/rfc1142

[75] Microsoft Knoweledge Base, "Description of Address Resolution Protocol (ARP) Caching Behavior in Windows Vista TCP/IP Implementations," Jan. 2010. [Online]. Available: http://support.microsoft.com/kb/949589/en-us

[76] Linux Man-Pages Project, "Description of ARP Kernel Module in Release 3.24." [Online]. Available: http://www.huge-man-linux.net/man7/arp.html

[77] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861 (Draft Standard), Internet Engineering Task Force, Sep. 2007, updated by RFC 5942. [Online]. Available: http://tools.ietf.org/html/rfc4861

[78] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The Nature of Data Center Traffic: Measurements & Analysis," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. New York, NY, USA: ACM, 2009, pp. 202–208.

[79] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," *ACM SIGCOMM Computer Communications Review*, vol. 39, no. 4, pp. 51–62, Aug. 2009. [Online]. Available: http://doi.acm.org/10.1145/1594977.1592576

[80] T. Benson, A. Akella, and D. A. Maltz, "Network Traffic Characteristics of Data Centers in the Wild," in *Proceedings of the 10th Annual Conference on Internet Measurement*. New York, NY, USA: ACM, 2010, pp. 267–280.

[81] Cisco Systems, "Introduction to Intermediate System-to-Intermediate System Protocol," White Paper, Apr. 2010. [Online]. Available: http://www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a3e6f.shtml

[82] ——, "Intergrated IS-IS Commands," Cisco IOS IP Command Reference Volume 2 of 3: Routing Protocols, 2006, Release 12.2. [Online]. Available: http://www.cisco.com/en/US/docs/ios/12_2/iproute/command/reference/fiprrp_r.html

[83] K. Elmeleegy and A. Cox, "EtherProxy: Scaling Ethernet By Suppressing Broadcast Traffic," in *IEEE Infocom*, Rio de Janeiro, Brazil, Jun. 2009, pp. 1584 – 1592.

[84] IETF Working Group , "Locator/ID Separation Protocol (lisp)," Apr. 2009. [Online]. Available: http://www.ietf.org/html.charters/lisp-charter.html

[85] R. Boyd and J. R. Purser, "LISP Host Mobility Demo," http://www.lisp4.net/news/2012/10/lisp-host-mobility-demo-vmworld-2012-barcelona/, Oct. 2012, [Online; accessed 8-July-2013].

[86] V. Fuller and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface," RFC 6833 (Experimental), Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/rfc6833

[87] L. Iannone and O. Bonaventure, "On the Cost of Caching Locator/ID Mappings," in *International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Dec. 2007.

[88] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)," RFC 6836 (Experimental), Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/rfc6836

[89] D. Lewis, D. Meyer, D. Farinacci, and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites," RFC 6832 (Experimental), Internet Engineering Task Force, Jan. 2013. [Online]. Available: http://tools.ietf.org/html/rfc6832

[90] Cisco Systems, "Cisco Locator/ID Separation Protocol and Overlay Transport Virtualization Data Center Infrastructure Solutions for Distributed Data Centers," White Paper, Mar. 2011. [Online]. Available: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/white_paper_c11-647157.html

[91] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," RFC 4423 (Informational), Internet Engineering Task Force, May 2006. [Online]. Available: http://tools.ietf.org/html/rfc4423

[92] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," RFC 5201 (Experimental), Internet Engineering Task Force, Apr. 2008, updated by RFC 6253. [Online]. Available: http://tools.ietf.org/html/rfc5201

[93] P. Nikander and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions," RFC 5205 (Experimental), Internet Engineering Task Force, Apr. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5205

[94] P. Nikander, T. Henderson, C. Vogt, and J. Arkko, "End-Host Mobility and Multihoming with the Host Identity Protocol," RFC 5206 (Experimental), Internet Engineering Task Force, Apr. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5206

[95] R. Atkinson and S. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description," RFC 6740 (Experimental), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6740

[96] R. Atkinson, S. Bhatti, and S. Rose, "DNS Resource Records for the Identifier-Locator Network Protocol (ILNP)," RFC 6742 (Experimental), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6742

[97] R. Atkinson and S. Bhatti, "ICMP Locator Update Message for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)," RFC 6743 (Experimental), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6743

[98] ——, "Optional Advanced Deployment Scenarios for the Identifier-Locator Network Protocol (ILNP)," RFC 6748 (Experimental), Internet Engineering Task Force, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6748

[99] R. Whittle, "Ivip (Internet Vastly Improved Plumbing) Architecture," IETF Internet-Draft, work in progress, Mar. 2010. [Online]. Available: http://tools.ietf.org/html/draft-whittle-ivip-arch

[100] ——, "DRTM - Distributed Real Time Mapping for Ivip and LISP," IETF Internet-Draft, work in progress, Mar. 2010. [Online]. Available: http://tools.ietf.org/html/draft-whittle-ivip-drtm

[101] ——, "Glossary of some Ivip and scalable routing terms," IETF Internet-Draft, work in progress, Mar. 2010. [Online]. Available: http://tools.ietf.org/html/draft-whittle-ivip-glossary

[102] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure, "Evaluating the Benefits of the Locator/Identifier Separation," in *ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Kyoto, Japan, Aug. 2007.

[103] A. Gladisch, R. Daher, and D. Tavangarian, "Survey on mobility and multihoming in future internet," *Wireless Personal Communications*, pp. 1–37, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11277-012-0898-6

[104] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275 (Proposed Standard), Internet Engineering Task Force, Jul. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6275

[105] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963 (Proposed Standard), Internet Engineering Task Force, Jan. 2005. [Online]. Available: http://tools.ietf.org/html/rfc3963

[106] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, "Proxy Mobile IPv6," RFC 5213 (Proposed Standard), Internet Engineering Task Force, Aug. 2008, updated by RFC 6543. [Online]. Available: http://tools.ietf.org/html/rfc5213

[107] R. Koodli, "Mobile IPv6 Fast Handovers," RFC 5568 (Proposed Standard), Internet Engineering Task Force, Jul. 2009. [Online]. Available: http://tools.ietf.org/html/rfc5568

[108] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380 (Proposed Standard), Internet Engineering Task Force, Oct. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5380

[109] Z. Zhu, R. Wakikawa, and L. Zhang, "A Survey of Mobility Support in the Internet," RFC 6301 (Informational), Internet Engineering Task Force, Jul. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6301

[110] B. T. University, "LISPmob: an open-source LISP implementation for Linux, Andriod and OpenRT." [Online]. Available: http://lispmob.org/

[111] R. Beverly, A. Berger, Y. Hyun, and kc claffy, "Understanding the Efficacy of Deployed Internet Source Address Validation Filtering," in *ACM Internet Measurements Conference (IMC)*, Nov. 2009.

[112] A. Varga, "INET Framework for the OMNeT++ Discrete Event Simulator," 2012. [Online]. Available: http://github.com/inet-framework/inet

[113] "The OpenLISP Project," Oct. 2011. [Online]. Available: http://www.openlisp.org/

[114] V. Ermagan, D. Farinacci, D. Lewis, J. Skriver, F. Mainor, and C. White, "NAT traversal for LISP," IETF Internet-Draft, work in progress, Mar. 2013. [Online]. Available: http://tools.ietf.org/html/draft-ermagan-lisp-nat-traversal

[115] ITU-T, "Recommendation H.264 : Advanced video coding for generic audiovisual services," Apr. 2013. [Online]. Available: http://www.itu.int/rec/T-REC-H.264-201304-I/en

[116] B. Juurlink, M. Alvarez-Mesa, C. Chi, A. Azevedo, C. Meenderinck, and A. Ramirez, "Understanding the Application: An Overview of the H.264 Standard," in *Scalable Parallel Programming Applied to H.264/AVC Decoding*, ser. SpringerBriefs in Computer Science. Springer New York, 2012, pp. 5–15. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-2230-3_2

[117] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (INTERNET STANDARD), Internet Engineering Task Force, Jul. 2003, updated by RFCs 5506, 5761, 6051, 6222. [Online]. Available: http://tools.ietf.org/html/rfc3550

[118] Y.-K. Wang, R. Even, T. Kristensen, and R. Jesup, "RTP Payload Format for H.264 Video," RFC 6184 (Proposed Standard), Internet Engineering Task Force, May 2011. [Online]. Available: http://tools.ietf.org/html/rfc6184

[119] ITU-T, "Recommendation P.910: Subjective video quality assessment-methods for multimedia applications," Apr. 2008. [Online]. Available: http://www.itu.int/rec/T-REC-P.910/en

[120] Xiph.org, "Derf's Test Media Collection," Mar. 2013. [Online]. Available: http://media.xiph.org/video/derf/

[121] B. Zatt, M. Porto, J. Scharcanski, and S. Bampi, "Gop structure adaptive to the video content for efficient H.264/AVC encoding," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sep. 2010, pp. 3053–3056.

[122] ITU-T, "Recommendation P.800: Methods for subjective determination of transmission quality," Aug. 1996. [Online]. Available: http://www.itu.int/rec/T-REC-P.800-199608-I/en

[123] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of Subjective and Objective Quality Assessment of Video," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1109/TIP.2010.2042111

[124] ITU-R, "Recommendation BT.500-13: Methodology for the subjective assessment of the quality of television pictures," Jan. 2012. [Online]. Available: http://www.itu.int/rec/R-REC-BT.500-13-201201-I/en

[125] ITU-T, "Recommendation P.10/G.100: Amendment 2: New definitions for inclusion in Recommendation ITU-T P.10/G.100," Jul. 2008. [Online]. Available: http://www.itu.int/rec/T-REC-P.10-200807-I!Amd2/en

[126] S. Winkler, "Video Quality and Beyond," in *Proc. European Signal Processing Conference*, 2007.

[127] A. Reibman, V. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Transactions on Multimedia*, vol. 6, no. 2, Apr. 2004.

[128] S. Tao, J. Apostopoloulos, and R. Guerin, "Real-Time Monitoring of Video Quality in IP Networks," *IEEE Transactions on Networking*, vol. 16, no. 6, Dec. 2008.

[129] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-Reference Video Quality Monitoring for H.264/AVC Coded Video," *IEEE Transactions on Multimedia*, vol. 11, no. 5, Aug. 2009.

[130] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment using structural distortion measurement," in *International Conference on Image Processing*, vol. 3, 2002, pp. 65–68.

[131] "x264 - H.264/AVC Encoder." [Online]. Available: http://www.videolan.org/developers/x264.html

[132] GPAC Multimedia Open Source Project, "MP4Box." [Online]. Available: http://gpac.wp.mines-telecom.fr/mp4box/

[133] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid - A Framework for Video Transmission and Quality Evaluation," in *Proc. of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, USA, Sep. 2003.

[134] "Tcpdump/Libcap." [Online]. Available: http://www.tcpdump.org/

[135] "Windump/Winpcap." [Online]. Available: http://www.winpcap.org/windump/

[136] The MPlayer Project, "MEncoder." [Online]. Available: http://mplayerhq.hu/design7/news.html

[137] Graphics and Media Lab, CMC department, Moscow State University, "MSU Video Quality Measurement Tool." [Online]. Available: http://graphics.cs.msu.su/en/node/941

[138] Joint Video Team (JVT), "H.264/AVC Reference Software Version JM12.3." [Online]. Available: http://iphome.hhi.de/suehring/tml/download

[139] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, Sep. 1960.

[140] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, Sep. 1963.

ISSN 1432-8801