Electronic Communications of the EASST
Volume 37 (2011)

Workshops der wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2011 (WowKiVS 2011)

Provisioning and Operation of Virtual Networks

Sebastian Meier, Marc Barisch, Andreas Kirstädter
Daniel Schlosser, Michael Duelli, Michael Jarschel, Tobias Hoßfeld
Klaus Hoffmann, Marco Hoffmann
Wolfgang Kellerer, Ashiq Khan, Dan Jurca, Kazuyuki Kozu

12 pages

Guest Editors: Horst Hellbrück, Norbert Luttenberger, Volker Turau
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Provisioning and Operation of Virtual Networks

Sebastian Meier[1], Marc Barisch[2], Andreas Kirstädter[3]
Daniel Schlosser[4], Michael Duelli[5], Michael Jarschel[6], Tobias Hoßfeld[7]
Klaus Hoffmann[8], Marco Hoffmann[9]
Wolfgang Kellerer[10], Ashiq Khan[11], Dan Jurca[12], Kazuyuki Kozu[13]

[1]sebastian.meier@ikr.uni-stuttgart.de, [2]marc.barisch@ikr.uni-stuttgart.de,
[3]andreas.kirstaedter@ikr.uni-stuttgart.de
Institute of Communication Networks and Computer Engineering, University of Stuttgart

[4]schlosser@informatik.uni-wuerzburg.de, [5]duelli@informatik.uni-wuerzburg.de,
[6]michael.jarschel@informatik.uni-wuerzburg.de, [7]hossfeld@informatik.uni-wuerzburg.de
Institute of Computer Science, University of Würzburg

[8]klaus.hoffmann@nsn.com, [9]marco.hoffmann@nsn.com
Nokia Siemens Networks GmbH & Co. KG

[10]kellerer@docomolab-euro.com, [11]khan@docomolab-euro.com,
[12]jurca@docomolab-euro.com, [13]kozu@docomolab-euro.com
DOCOMO Communications Laboratories Europe GmbH

Abstract: In today's Internet, requirements of services regarding the underlying transport network are very diverse. In the future, this diversity will increase and make it harder to accommodate all services in a single network. A possible approach to keep up with this diversity in future networks is the deployment of isolated, custom tailored networks on top of a single shared physical substrate. The COMCON (COntrol and Monitoring of COexisting Networks) project aims to define a reference architecture for setup, control, and monitoring of virtual networks on a provider- and operator-grade level. In this paper, we present the building blocks and interfaces of our architecture.

Keywords: Network Virtualization, GMPLS, Control Plane, Monitoring

1 Introduction

Today's Internet provides access to many services, e.g., email, web, and file transfers, but its structure is inflexible and it is hard to introduce new network services with individual quality of service (QoS) requirements. The future Internet will be faced with additional challenges like a rising number of mobile users connecting to the Internet using a wireless link, which currently cannot provide the same QoS as a fixed network. Moreover, the Internet will change from a mere everywhere network to a "real-time" everywhere network, which supports for example high-quality video transmission. Therein, transport of information underlies hard time constraints, regardless of the type of information or its data volume. However, the Internet architecture is

still bound to its best effort basis and is not able to satisfy these demands in its current form.

Network virtualization (NV) technology is the key component to keep up with this development by reducing the time and overhead required to introduce new services, change the reach of existing networks or support cloud computing. NV can be used to consolidate networks on a functional level and differentiate them on a service level. In such a future Internet, a multitude of virtual networks (VNet) will coexist and complement each other. These coexisting networks allow specialization but require isolation of functionalities to provide dependable and predictable networks.

The objective of the COMCON project is to design novel control and management mechanisms that support the coexistence of VNets in a future networking scenario and to illustrate their economic advantages. Thereby, COMCON addresses a couple of challenges that have not been sufficiently considered by existing approaches. *1) Network Operation:* Many approaches detail the setup of VNets but do not consider their operation, including reassigment of ressources, resizing of VNets, failure handling and efficient monitoring. *2) Arbitrary Network Technologies:* NV has to support arbitrary network technologies (WDM, SDH, Ethernet, IP, etc.) as physical substrate as well as each VNet should be allowed to run its own set of protocols that may differ from today's Internet Protocol (IP). *3) Technology Migration and Reuse:* Existing protocols and architectures for network management and control should be reused for VNet setup and VNet operation in order to simplify the introduction of NV. *4) Traffic Management:* Future VNets have to enable traffic management by a control plane that combines network infrastructure (of mobile and fixed core and access networks) and IT infrastructure (e.g. data centers) supporting QoS requirements. This allows that the end user always perceives the best quality of experience (QoE), while the operator uses the resources in the most economical way.

In this paper, we describe our reference architecture for NV that addresses the above introduced challenges. It includes the operation, control, and monitoring of VNets considering different functional roles and their information exchange. We present the building blocks and interfaces between them.

The remainder of this work is structured as follows. In Section 2, we give an overview of related work. Based on the role model introduced in Section 3, we specify our reference architecture for NV in Section 4 and illustrate the setup and operation of VNets. Section 5 presents one instantiation of parts of the reference architecture by means of GMPLS. Finally, Section 6 concludes the paper.

2 Related Work

The influence of NV for the traditional Internet Service Provider (ISP) role model has been introduced in [FGR07]. The authors propose to split up the ISP role into an infrastructure provider managing the physical resources and a service provider deploying enabler services such as routing, DNS as well as end-to-end services. The 4WARD project refined this role model in [BW09]. Furthermore, 4WARD introduced interfaces for role interaction with the focus on virtual network deployment and end user attachment. For instantiation of virtual networks with QoS guarantees, [CFT+05] proposes a virtualization architecture based on DiffServ/MPLS enabled transport networks. To translate between different QoS parameters across several roles, a multi-tier architec-

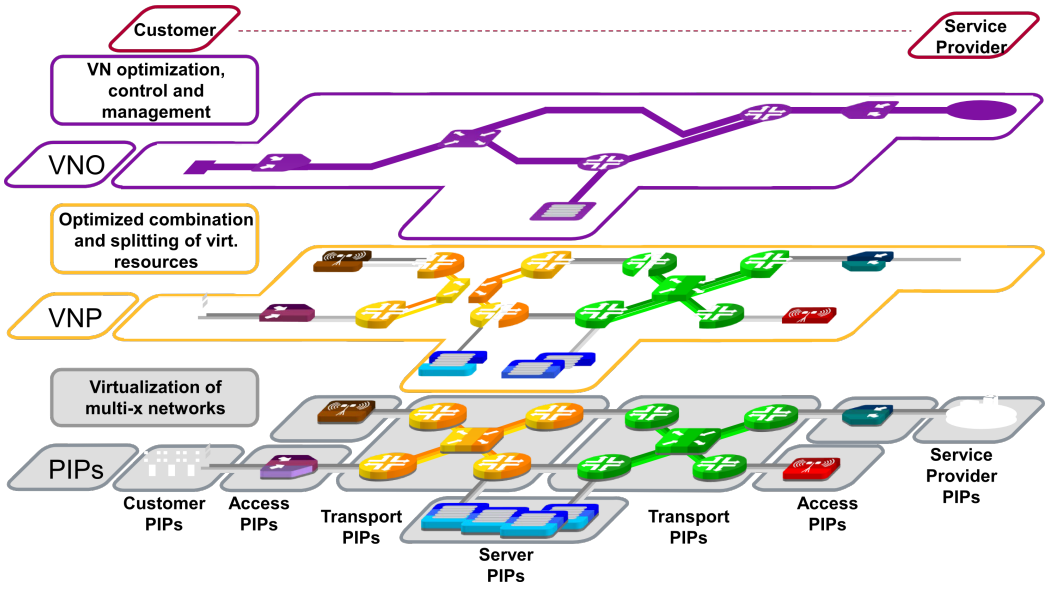


Figure 1: Overview of Roles for a Carrier-Grade Network Virtualization

ture for SLA management is proposed. Considering architectures to enforce QoS parameters, [BPW98] discusses the advantages and disadvantages of mobile software agents in contrast to a centralized management approach. Due to the large scale and heterogenity of future networks the authors of [F+08] propose a related approach, where network elements carry out local management operations in an autonomous way without relying on a central management component. An implementation of NV that focuses on edge networking is FlowVisor [S+10]. A FlowVisor acts as a transparent proxy for OpenFlow switches, replicates the OpenFlow interface, and supervises access and usage of it. Thereby, the underlying infrastructure can be shared by several VNets that are configured via OpenFlow. In contrast to FlowVisor which is focused on virtualization of campus size networks, [SIO08] addresses virtualization in optical backbone networks. The authors present the IP Optical TE server that calculates and triggers creation of traffic engineered paths in multi-layer networks. The DRAGON project presents an architecture for inter-domain virtual path provisioning [Y+06]. The implemented mechanisms for path computation and resource reservation are based on GMPLS.

3 Role Model

NV provides the opportunity for new business models. It is no longer mandatory that a single company owns the physical hardware and operates the network on top. Therefore, we have defined a role model that characterizes the roles according to their functional aspects. Our role model has been inspired by the role model defined in the 4WARD project [CJ09]. We distinguish between three functional roles – Physical Infrastructure Provider (PIP), Virtual Network Provider (VNP) and Virtual Network Operator (VNO) – as well as between the end customer (EC) and

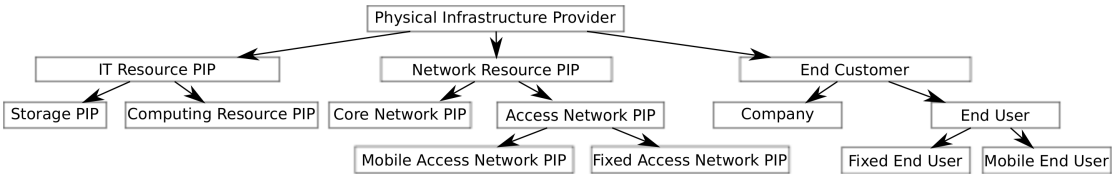


Figure 2: Classification of Physical Infrastructure Providers

the service provider (SP), as illustrated in Figure 1. The main difference between our role model and 4WARD is that we consider the EC as a PIP, as we explain in the following.

3.1 Physical Infrastructure Provider

The PIP owns and operates physical resources (e.g. networking equipment), which it splits into virtualized resources and offers them to its customers, the VNPs. The properties of the provided virtual resources are defined in a service level agreement (SLA). To verify whether the provided resources are compliant with the SLA, the PIP monitors its physical and virtual resources.

We consider any party as a PIP, if it owns physical resources and offers these resources to other parties. Figure 2 provides an overview on what we consider as PIPs.

Network Resource PIP (N-PIP): A N-PIP offers not only paths between different nodes to VNPs, but also virtualized network elements (VNEs) like virtual switches and virtual routers. Virtualization of these elements has several advantages. First, VNEs facilitate the control of routing within a VNet and install individual network protocols that replace existing protocols like IP. Second, a N-PIP can optimize its resource utilization by moving VNEs and virtual links inside the physical substrate. This technique might be used amongst others to improve energy efficiency within the physical network.

IT Resource PIP (I-PIP): An I-PIP is a typical data or computing center operator that offers virtual servers in terms of CPU, memory, and storage capacity to a VNP. The offered virtual servers permit installation of customized operating system images to provide special services within a VNet. Any provider of cloud services can be considered as an I-PIP.

End Customer: The EC can be either a company that is for example interconnecting its branches or an end user (EU). We consider the EC, in particular the EU, as PIP since it owns in some cases an end device, called customer premise equipment (CPE). The CPE can be operated by another party that virtualizes it and offers parts of the resources to VNPs providing different VNets. In addition, it is possible that software is installed on the CPE in order to support new network protocols.

3.2 Virtual Network Provider

A VNP gathers virtual resources from different PIPs and composes VNets upon these resources. The resulting VNets provide basic connectivity on a particular networking layer (e.g. L2 Ethernet connectivity) and interfaces for configuration. This role is required for several reasons. *1) Geography*: PIPs offer their resources only in certain geographic regions. Therefore, a VNP can for example construct global VNets with end-to-end QoS characteristics consisting of re-

sources provided by different PIPs. *2) Specialization*: PIPs will specialize their services, i.e. one PIP will offer only storage and computing resources. *3) Competition:* PIPs offer their resources at different cost. This allows the VNP to select the cheapest PIPs to compose its networks and allows changing PIPs upon cost changes. *4) Reliability:* A VNP can select resources from different PIPs to improve the availability of its VNet.

3.3 Virtual Network Operator

A VNO requests networks with special characteristics from the VNP, which allows for optimization of the network according to its purpose, e.g. content delivery. Afterwards, the VNO brings the obtained VNet with basic connectivity to life, which means among others the definition and installation of networking protocols, configuration of routing, and installation of virtual servers.

After bringing the VNet to life, the VNO is responsible for network operation. This includes the monitoring of the VNet and appropriate reactions to network service degradations. For example, if the customer base of the SP increases, the required bandwidth increases and the VNO must be able to request additional resources from the VNP. If the service degradation is caused by one of the participating PIPs that is hidden from the VNO, the SLA violation has to be reported to the VNP, which can trigger appropriate actions.

3.4 Service Provider

The SP designs services and offers them to ECs via dedicated VNets. For example, a SP can offer an IPTV service via a dedicated VNet that has special characteristics for fast and reliable transmission of TV streams. That means the SP characterizes its service and provides the VNO information about the service requirements, like bandwidth, tolerable delay, and jitter.

4 Reference Architecture

Based on the functional roles, we have defined a reference architecture for NV. Section 4.1 provides different views on the reference architecture, which we introduce in more detail in Section 4.2 and Section 4.3.

4.1 Overview

Each role performs particular tasks and relies on functionality provided by subjacent roles. The reference architecture refines this approach by introducing functional blocks within each role and abstract interfaces for interaction. These interfaces are implemented by particular instantiations of the reference architecture using different technologies. One instantiation is detailed in Section 5. The reference architecture specifies all required interfaces to cover the lifecycle of a VNet that consists of three different phases as depicted in Figure 3.

1) Network Setup: The SP triggers the setup of a new VNet. Based on this request, the VNO and the VNP instantiate the network with the resources provided by the PIPs.

2) Network Operation: Within this phase the VNet is monitored, controlled, and adapted to changing requirements if needed.

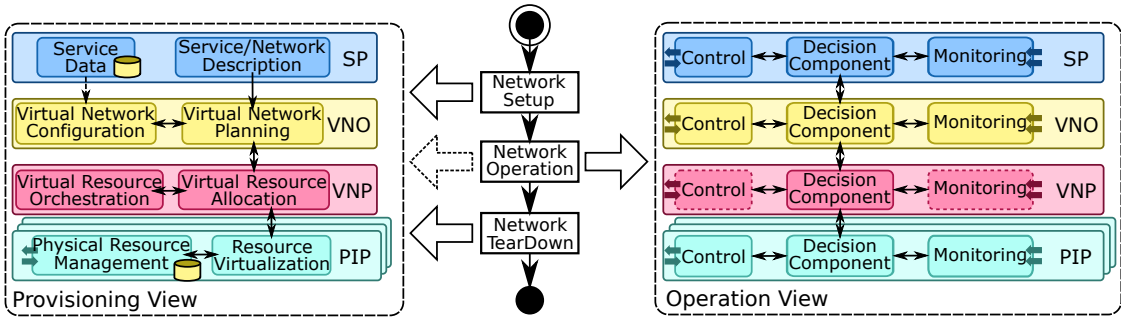


Figure 3: Lifecycle of a Virtual Network and Corresponding Interfaces

3) Network Teardown: The lifecycle of a VNet ends with the Network Teardown phase, during which the occupied resources are released.

Each phase of the lifecycle requires dedicated interfaces and corresponding service primitives between the different roles. Basically, we can group all work flows and interfaces into two categories, which is reflected by the definition of two views.

Provisioning View: It covers all interfaces that are needed to setup, modify, and tear-down a VNet. The interfaces needed in this phase are detailed in Section 4.2.

Operation View: During the operation phase of a VNet, the VNet needs to be monitored, controlled, and adapted to changing requirements and changing environmental conditions (e.g. link failures, link overload). This requires additional interfaces compared to the provisioning phase. More details on the operation phase are given in Section 4.3.

4.2 Provisioning View

The provisioning view contains all work flows and interfaces to setup, adjust, and teardown a virtual network. The process to setup a VNet is illustrated in Figure 4. In advance, the VNP and the PIPs have to establish trust relations (step 1). This is necessary to secure the exchange of resource, control, and monitoring messages. Based on the degree of trust, the PIP provides information about its resources to the VNP (step 2). In case of low trust, the PIP provides only information on the type of resources it provides. Whereas in case of high trust, e.g. the VNP and PIP are the same entity, also load and topology information might be exchanged.

The actual setup of a VNet is triggered by a SP with a *Request for VNet* (step 3). Depending on the type of service that should be deployed in the VNet and the knowledge of the SP we distinguish two different approaches for service and network planning: SP-driven VNet planning and VNO-driven VNet planning. In case of *VNO-driven planning*, the SP describes the service it intends to offer in general terms, i.e. the type of service (e.g. VoIP, IPTV), the number of users and their location, etc. The VNO is responsible for the service and network planning and the corresponding deployment (step 4). This includes the dimensioning and placement of network and computing resources as well as determining required enabling services (e.g. DNS). To carry out this task, the VNO relies on network templates that contain information about the service type and the required network characteristics (e.g. bandwidth, delay constraints). The

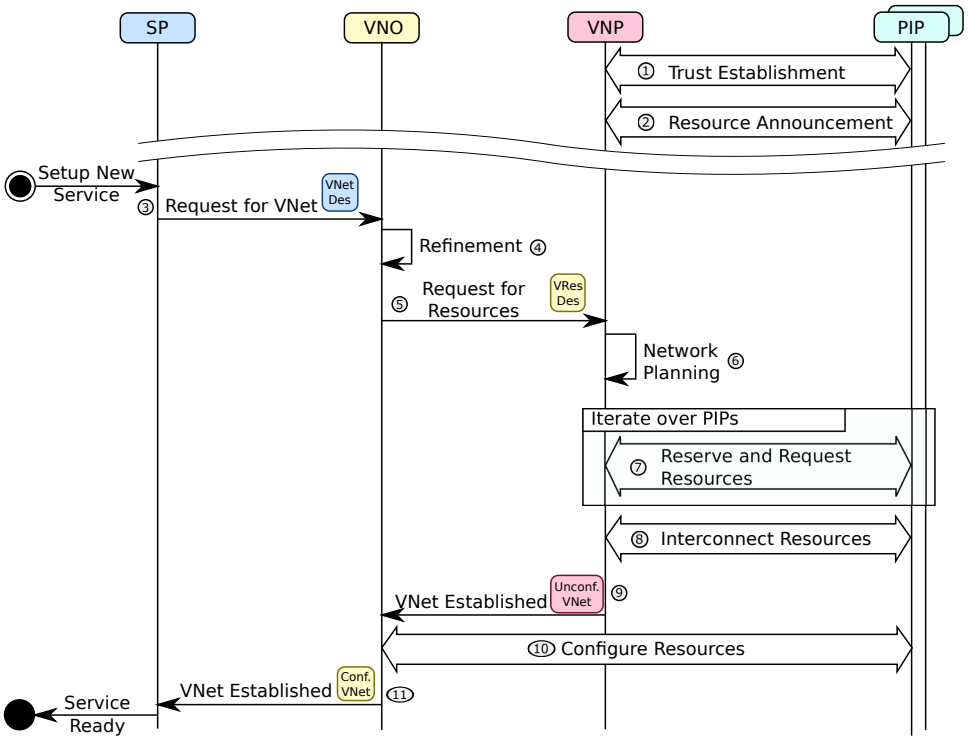


Figure 4: Setup of a Virtual Network

template allows VNet dimensioning according to the service requirements. In case of *SP-driven planning*, the SP provides a detailed virtual network description (VNetDes) towards the VNO. Therefore, the SP has direct influence on the VNet to provide its service. This approach is more suited for new services that require particular network configurations, which cannot be covered by templates.

The VNO creates a virtual resource description (VResDes) based on the information obtained from the SP as input for the VNP (step 5). The VResDes contains among others the description of the virtual network topology and the required virtual resources (e.g. storage). Depending on the requested resources and their geographic distribution, the VNP plans the VNet (step 6) and requests the resources from one or several PIPs (step 7). In the latter case, the VNP splits up the VResDes and requests and reserves resources from particular PIPs. For instance, storage resource requests are forwarded to storage PIPs, whereas network resource requests are forwarded to network PIPs. Eventually, the VNP receives the requested resources from all involved PIPs and interconnects the network resources (step 8), IT resources and EC with each other – and hands the VNet over to the VNO (step 9). Finally, the VNO configures the obtained network and IT resources with special network protocols and particular server images (step 10) and informs the SP about successful VNet setup (step 11).

4.3 Operation View

During the operation phase of a VNet, the VNet needs to be monitored, controlled, and adapted to changing requirements and changing environments. To that end, each functional role comprises measurement nodes within its components. These measurement nodes gather information about the component state and accumulate this knowledge into a monitoring database which provides the current network state to the decision component (DC) (see Figure 3). Furthermore, the DC is attached to actuators which are able to change settings and control the resources. Even if the measured data and the available controls differ for each functional role, we can generalize some aspects. We introduce these generalization by means of control and monitoring patterns in Section 4.3.1 and examplify them in Section 4.3.2.

4.3.1 Control and Monitoring Patterns

We have identified three different control and monitoring patterns that can be hierarchically combined: Horizontal Control Loops, Vertical Control Loops triggered by upper layers and Vertical Control Loops triggered by lower layers.

Horizontal Control Loops: In the operation phase, each functional role (PIP, VNP, VNO, SP) has a control component, a DC, and a monitoring component as shown in Figure 5(a). With these three components the role is able to manage its resources and fulfil the agreed SLAs. Based on obtained monitoring data (step 1), the DC can instruct the control component to trigger certain actions (step 2). The result of the actions is perceived by the monitoring component (step 3). For example, if the monitoring component measures high packet delays, the DC can decide to increase the bandwidth on one link, which is performed by the control component. This results in reduced packet delays confirmed by the monitoring component.

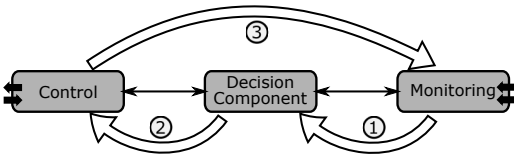
Vertical Control Loops: Vertical control loops show the interworking between two adjacent roles. In case one role is not able to solve the detected issue alone it has to cooperate with an adjacent role.

Vertical Control Loops triggered by upper layer: The monitoring of the upper layer detects a problem and informs its DC (see Figure 5(b)). The DC cannot solve the problem by means of horizontal control and informs the DC of the lower layer, which triggers appropriate control actions. The result of the control actions is perceived by the monitoring component of the lower layer as well as by the monitoring layer of the upper layer.

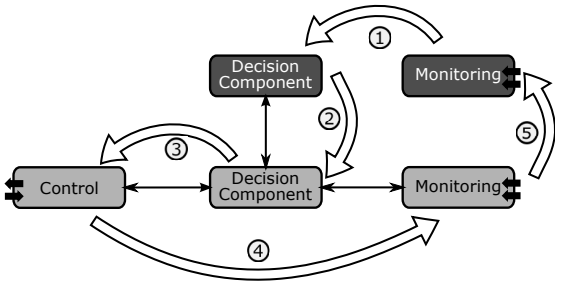
Vertical Control Loops triggered by lower layer: In contrast to Figure 5(b), the monitoring of the lower layer in Figure 5(c) detects a problem and informs its DC which escalates the problem to the DC of the upper layer. The control loop is not necessarily closed. For instance, a VNP might decide to replace a misbehaving PIP by another PIP.

4.3.2 Control and Monitoring Patterns in COMCON

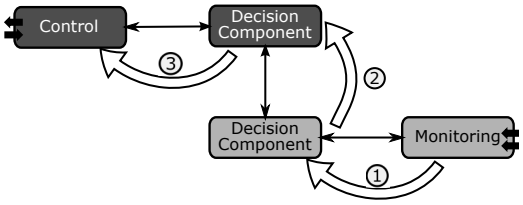
In the following, the above introduced control and monitoring patterns are exemplified for the COMCON reference architecture. The PIP monitors the health of the physical substrate and the current state of the virtualized resources. The health of the physical substrate is thereby defined by the physical state of the PIPs links and nodes. A failure of a physical entity may affect the offered virtualized components in different ways. The failure of an unprotected physical link,



(a) Horizontal Control Loop



(b) Vertical Control Loop Triggered by Upper Layer



(c) Vertical Control Loop Triggered by Lower Layer

Figure 5: Control Loops

this results in the failure of a virtual link mapped to this physical resource. In contrast, if the virtual link was set up with failure recovery options, it will not be affected. In both cases, the DC receives an alert and checks whether the problem is caused by a configuration or a hardware failure. In the first case, it might be possible to use the actuators to solve the problem. In the latter case, someone will have to replace the broken entity. Besides these physical problems, the PIP is interested in the general load on its components and the state of the virtual entities. To save energy, a PIP can optimize its resource utilization by moving VNEs between physical resources. Idle network elements can be suspended.

Within our role model the VNP is mainly involved in the network setup phase. But some of its activities are also performed during the operation of a network. Whenever a VNO or a PIP wants to change the contracted virtual resources the VNP has to act as a mediator as it builds a layer of indirection and has to renegotiate between the VNO and the PIP. For all other actions during the operation phase, the VNP is restricted by the fact that it neither has its own network nor is able to access a PIP's physical or a VNO's virtual network. Hence, the VNP has to rely on the information and actions provided by the PIPs. In case the VNP wants to establish a pool of resources, which it keeps reserved in order to react quicker to resource requests by a VNO, the "monitoring" of this pool is based on contracts with PIPs. Furthermore, a VNP can only migrate virtual resources between different PIPs if the PIPs offer an appropriate feature.

The VNO is responsible for operating the network within the SLA of the SP. Therefore, the VNO will use measurement points at its controlled virtual network elements. These will provide it with all the information, needed to operate and optimize its virtual network. The DC of the VNO decides on this information, e.g. packets dropped in virtual routers, packet delays, and bandwidth usage, how to adapt the network. It has to be noted that the VNO is in full control of the virtual network elements like virtual routers. For example, the VNO may install its own virtual router operating system and protocol stack. Hence, the VNO is not only able to change

the routing or shape bandwidth, but also to set up performance enhancing proxies and perform arbitrary traffic engineering within the network.

The monitoring performed by the SP depends on the scenario. In case of *VNO-driven monitoring*, the SP may hand over the service component to the VNO in order to let it run the complete service including monitoring. The only thing a SP wants to record might be direct user feedback in order to verify its own expectations regarding the users perception (QoE) of the service quality. In case of *SP-driven monitoring*, the SP runs the service on application layer itself and monitors the end-to-end perspective, i.e. the performance of the service and network QoS between the service location and the end user devices. Depending on the service offered, the SP might have different ways to optimize the service on application layer. For instance, if we consider a video provider, the usage of scalable video codecs would allow many options for adopting the service on the application layer.

Besides the horizontal control loops, there are also situations in which vertical control loops are required. It is the task of the DC in each layer to decide whether the problem is caused and might be solved within the current role or if the situation demands the cooperation of multiple layers. If we consider the case of a failure, the DC has to determine first whether the problem is caused in the layer itself or at an underlying layer. For instance, if a VNO detects that the delay between two nodes is too high, the DC has to decide whether the problem is caused by, e.g., bogus packet routing, or if the transmission between two virtual nodes in the substrate is slower than defined in the SLAs with the PIP. In the latter case, the DC of the affected layer contacts the DC of the layer underneath and complains about an SLA violation. In case of a multi-domain scenario, i.e. one VNO is running its network on the substrate of several PIPs, the VNO needs to request virtual network nodes at the peering points between PIPs. Only with virtual nodes at these points, the VNO is able to provide the VNP with sufficient information to find the PIP causing the problem.

5 Instantiation

For network operators that occupy one or several roles (e.g. PIP, VNP or VNO) it is essential to present a migration path from existing network architectures towards NV. We consider Generalized Multi-Protocol Label Switching (GMPLS, [Man04]) as a possible technology to instantiate the reference architecture for virtualization of core networks. Since the GMPLS framework is technology independent we are in the position to support heterogeneous substrate technologies with a common GMPLS control plane. The IETF specifies protocols and elements GMPLS relies on. The Open Shortest Path First protocol with Traffic Engineering (OSPF-TE) and GMPLS extensions [KR05] enables network topology discovery and distribution. OSPF-TE provides topology information to a Path Computation Element (PCE, [FVA06]), which performs constraint-based path computation for traffic engineered paths (TE-Paths). The Resource Reservation Protocol with Traffic Engineering and GMPLS extensions (RSVP-TE, [Ber03]) enables TE-Path maintenance consisting of setup, management, and teardown. An extension to RSVP-TE enables the setup of forwarding adjacencies (FAs). Those are TE-Paths which appear as a point-to-point connection and thus hide the underlying network topology supporting the path. These adjacencies are re-announced by OSPF-TE and appear as regular links in the network topology.

These links may be associated to an administrative group, which is a header field in OSPF-TE messages and might be used to transport ownership information for a link. Thereby only minor extensions to GMPLS are required for link virtualization. Node virtualization is not covered by GMPLS so far. However, we intend to extend GMPLS for node virtualization. OSPF-TE already supports announcement of link properties and occupation. By introducing new message types announcement of node properties and node occupation is possible as well. With topology information extended to contain node information, calculation for virtual node placements is feasible. As VNets consist of many nodes and links, more sophisticated virtual network computation elements (VNCE) will replace todays PCEs. Those VNCEs have to coordinate solution finding for virtual nodes and virtual links. To signal creation of virtual nodes, RSVP-TE might be extended.In general all mentioned control plane protocols have to be extended regarding the handling of virtual resources in the core network. So far, we do not consider GMPLS for the attachment of end users or the handling of mobility. However GMPLS can be used to redimension virtaual networks to cope with mobile users.

6 Conclusion And Outlook

In this paper, we presented the COMCON reference architecture for network virtualization. We presented the role model underlying this architecture as well as the functional blocks and interfaces for setup, operation and tear-down of virtual networks. In the future, we will address more elaborate topics, such as multi-domain scenarios, attachment of mobile end users, and hierarchical virtualization. So far we verified that the architecture we presented can be extended to support the aforementioned topics.

In multi-domain scenarios, interworking between PIPs is required. That means PIPs have to exchange peering information for virtual links ranging over several PIPs. So far we have identified two approaches to establish multi-domain links. Either a VNP can coordinate the establishment, or PIPs have to exchange signaling information directly. Especially the latter approach is challenging in heterogeneous scenarios, where different PIPs might use different control plane technologies for network control. Both approaches need further evaluation.

Considering the attachment of end users several issues need further discussion. As we consider users as PIPs, especially in mobile scenarios, effective solutions for organizing PIP peering information have to be found. For multi-homed CPE mechanisms for automatically selecting the most suitable network PIP require further research. In mobile scenarios, hand-over mechanisms at the PIP level as well as the VNO level need to be considered in more detail. Furthermore we need to investigate how a user gains access to a VNet, in case the network is currently not hosted by the PIP he is connected to (roaming scenario).

Acknowledgements: This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01BK0918, GLab). The authors alone are responsible for the content of the paper.

Bibliography

[Ber03] L. Berger. Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions. RFC 3473 (Proposed Standard), Jan. 2003.

[BPW98] A. Bieszczad, B. Pagurek, T. White. Mobile Agents For Network Management. *Communications Surveys Tutorials, IEEE* 1(1):2 –9, 1998.

[BW09] R. Bless, C. Werle. Network Virtualization From A Signaling Perspective. *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pp. 1–6, June 2009.

[CFT+05] Y. Cheng, R. Farha, A. Tizghadam, M. S. Kim, M. Hashemi, A. Leon-Garcia, J.-K. Hong. Virtual Network Approach To Scalable IP Service Deployment And Efficient Resource Management. *Communications Magazine, IEEE* 43(10):76 – 84, 2005.

[CJ09] J. Carapinha, J. Jiménez. Network Virtualization: A View From The Bottom. In *VISA '09: Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*. Pp. 73–80. ACM, New York, NY, USA, 2009.

[F+08] C. Foley et al. A Framework For In-Network Management In Heterogeneous Future Communication Networks. In *MACE '08: Proceedings of the 3rd IEEE international workshop on Modelling Autonomic Communications Environments*. Pp. 14–25. Springer-Verlag, Berlin, Heidelberg, 2008.

[FGR07] N. Feamster, L. Gao, J. Rexford. How To Lease The Internet In Your Spare Time. *SIGCOMM Comput. Commun. Rev.* 37(1):61–64, 2007.

[FVA06] A. Farrel, J.-P. Vasseur, J. Ash. A Path Computation Element (PCE)-Based Architecture. RFC 4655 (Informational), Aug. 2006.

[KR05] K. Kompella, Y. Rekhter. OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS). RFC 4203 (Proposed Standard), Oct. 2005.

[Man04] E. Mannie. Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945 (Proposed Standard), Oct. 2004.

[S+10] R. Sherwood et al. Carving Research Slices Out Of Your Production Networks With OpenFlow. *SIGCOMM Comput. Commun. Rev.* 40(1):129–130, 2010.

[SIO08] K. Shiomoto, I. Inoue, E. Oki. Network Virtualization In High-speed Huge-bandwidth Optical Circuit Switching Network. *INFOCOM Workshops 2008, IEEE*, April 2008.

[Y+06] X. Yang et al. Policy-based Resource Management And Service Provisioning In GMPLS Networks. In *In First IEEE Workshop on Adaptive Policy-based Management in Network Management and Control*. 2006.