Describing Commercial Virtual Networks: Requirement Analysis and TMF SID-based Solution

David P. Wagner*, Sebastian Meier*, Klaus Hoffmann†, Franz Rambach‡
*Institute of Communication Networks and Computer Engineering, University of Stuttgart, Germany
Email: {dwagner | smeier}@ikr.uni-stuttgart.de
†Nokia Siemens Networks Management International GmbH, Germany, Email: klaus.hoffmann@nsn.com
‡Logica, Munich, Germany, Email:franz.rambach@logica.com

***Abstract*—Today, Network Virtualization (NV) plays an important role in the networking research and gains increasing interest in the business environment. In addition to technical enhancements, NV also allows more differentiated business models diversifying the traditional Internet Service Provider (ISP) role. A crucial prerequisite for such a new business world is a description methodology that allows the parties to negotiate about Virtual Networks (VNs). Such a common description model has to cover links and nodes, their key properties, capabilities and control interfaces. In this paper we detail the requirements for Virtual Network Descriptions (VNDs) exchanged in a business environment guided by a representative scenario. We then match existing description methodologies against these requirements and find that SID (formerly "Shared Information/Data Model") from Telemanagement Forum (TMF) is the most promising candidate. For SID we propose usage directives and extensions to make the information model fulfill all requirements of a commercial VN environment.**

I. Motivation

In recent years, for service providers it became more and more attractive to rent infrastructure rather than owning it themselves. This concerns cloud-based services based virtual computation and storage resources, but also in the networking area where it becomes increasingly attractive to share networking infrastructure, e.g. of mobile networks. This developing market trades Virtual Networks (VNs) consisting of nodes and links with defined and guaranteed properties [1] that can be controlled like real infrastructures. In such market, companies negotiate about VN, hand over control on VNs, adapt VNs etc. For these interactions an efficient description methodology for VNs is needed. Such Virtual Network Description (VND) must characterize the topology of the VN with its links and nodes and their functional properties as well as the means to dynamically configure or control the VN.

This paper identifies requirements for VNDs in business communications. Existing description methodologies for virtual and physical infrastructures are evaluated. We found the information model of Telemanagement Forum (TMF), SID (originating from "Shared Information/Data Model") [2] being well prepared for the targeted purposes and present extensions to overcome identified limitations.

II. Virtual Networks in a Business World

The core idea of the concept of VNs is to tailor a network to a service's needs by placing functionality at the best possible place in this network, and having the topology and the configuration of its nodes constantly adapted to internal and external changes.

A. Different views on VNs in a vertical market

We identified a task-oriented role model in [1] which consists of four roles. The *Application Service Provider (ASP)* regards his service on a rather high abstraction level, e.g. which kind of service and potential customers. He requests a *Virtual Network Operator (VNO)* to design, dimension and operate the VN for that service. Figure 1 shows the views of the different actors for a simple example. Here the VNO chooses a redundant topology to achieve reliability, Open Shortest Path First (OSPF) routers and Virtual Machines (VMs) for his software. He also requests access to control interfaces (called *Chaperone Interfaces*, see also II-B) for the nodes, i.e. routers and VMs. The VNO negotiates rental of the detailed VN design from a *Virtual Network Provider (VNP)*. A VNP aggregates and brokers virtual resources. So for a request from a VNO, the VNP defines a suitable partitioning of the VN and negotiates the supply of the respective virtual resources from different *Physical Infrastructure Providers (PIPs)*, see the fine dashed boxes in Figure 1. Furthermore, the VNP in some cases splits one virtual resource of the VN to several resources provided by several PIPs which in total fulfill the requirements. An example is the link between the upper playout server and the upper router in Figure 1: the VNP had to introduce a transit point and defined the delay limits accordingly. A PIP contributing to a VN only knows about the partition assigned to by the VNP, but for that he even knows about the mapping to the physical infrastructure.

B. Key Observations

In this section we highlight properties of the VNDs that would be exchanged between business partners in such business Virtual Network Environment (VNE).

1) Different Levels of Abstraction: The different views of the parties in negotiations on one VN have to be translated into each other. Therefore, a VND needs to support generalizing and removing or adding details without distortions.

2) Few Basic Types of Exchanged Objects: The VNDs exchanged during negotiations of the VN of Figure 1 consist of only four kinds of elements: generic *VMs* providing a platform to deploy the VNO's software, *routers* operating the OSPF protocol on selected interfaces, *links* with specified minimal bandwidth and maximum delay and domain *transit points*.

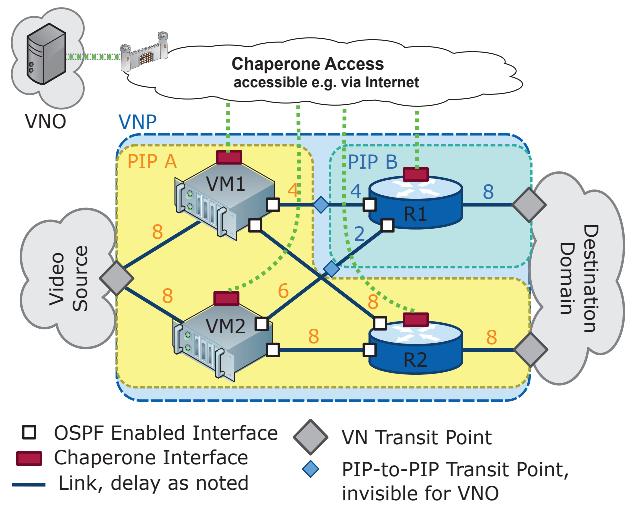

Fig. 1. VNO's view on the VN, embedding VNP's and PIPs' views

3) Connections to external networks: First, we expect many commercial VNs using external resources, e.g. the video source in the scenario in Section II-A. Second, the destinations of the (commercial) service for which this VN is operated, will often be not part of the VN itself.

4) Run-time Topology Adaptations: The VNO of commercial VNs may want to adapt the VN's topology during operation in order to react to internal and in particular external changes, e.g. changing demand.

5) Control Interfaces: Apart from topology changes, run-time adaptations may also affect the internal configuration of nodes only, e.g. the installation or configuration of software running on a VM. This control access should be independent from the VN itself. Therefore, each VN comes with an auxiliary management connectivity, which we call *Chaperone Access*. It is provided by the PIPs and aggregated by the VNP. PIPs and VNP secure the use of and access to the Chaperone Access by authorization, authentication and encryption mechanisms.

6) Limited Trust: Since any business partner can be a reseller and a potential competitor, players in this market of VNs are not willing to disclose more information than necessary, e.g. in negotiations.

III. Objective and Requirements of VN Descriptions

The work split between the different business roles introduces the need to converse on VNs, prominently to negotiate new VNs, but also during operation of a VN. Before creation of a VN there are negotiation processes between VNO and VNP and between VNP and PIPs which define *what* will be provided *when* for which *price*. These communications must be automatable in order to facilitate services using VNs on demand. Generally, the VNDs exchanged in such negotiations describe two different matters: On one hand VN instances, describing an existing VN, where all connections to the outside are set up and have to be included in the VND. On the other hand VN requests and offers, both describing the concept of a VN, a VN that does not exist (yet). VNDs of the second category leave room to choose (and optimize), while VNDs of the first identify specific VNs so all these choices have been made already. Besides from these differences, the two types are similar, so they are collectively examined in the following analysis of requirements for a suitable description methodology of VNs.

a) Universality & Generalization: We explicitly aim at supporting any company taking on any set of responsibilities, e.g. acting as a VNO for one service while contributing resources as a PIP for another. In order to facilitate this freedom, we look for a description methodology that allows representing very abstract networks as well as very technically detailed network models.

b) Extensibility and Fitness for the Future: The development of networking infrastructures, protocols and functions is ongoing, e.g. recently OpenFlow-controlled switches [3] gained much interest. We are looking for a description methodology that considers development of network infrastructures in its design. The use of inheritance can often provide a sound foundation for supporting future developments.

c) Different Node Types: The nodes in a VN can be assigned to different categories: First, there are nodes whose function is exhaustively defined by their type, e.g. an Ethernet switch. Second, there are nodes which are defined by the fixed functionality they provide. This functionality can be described either by the protocols or by the software running. Third, there are nodes whose function is freely defined by the VNO via installing own software on a generic platform. Then the platform itself has to be described in the VND with all information relevant for execution and performance, e.g. instruction set, amount of memory, number of cores etc.

d) Different Link Types: If a PIP only provides one part of a VN, the inter domain transit connection to neighboring PIPs has to be compatible. This requires an agreement on the physical line or port used as well as on the logical and physical signal, e.g. wavelength and encoding for an optical peering.

e) Capacity Properties: For some links and nodes, capacities have to be expressed, such as guaranteed bandwidth.

f) Notion of Network Interfaces and Binding Functionality to Interfaces: If nodes have several interfaces, often different functions shall be provided on these interfaces, just think of a network firewall. If that functionality is configured by the VNO, he needs to know which interface will face which connection and neighboring node. So the VND must support expressing binding functionality to interfaces, e.g. the OSPF protocol on node R1 in Figure 1.

g) Chaperone Interfaces & Chaperone Access: The interfaces for Chaperone Access must be accessible for the VNO so they must be part of the VND. In general, the VNO not necessarily requires Chaperone Access to all potentially reconfigurable elements and such access is only provided where requested. Therefore, a VND has to provide means to express the request for a Chaperone Interface as well as the access information for Chaperone Interfaces.

h) Discrete Representation of All Adaptable Elements, especially Functionalities: To support arbitrary adaptation requests all independently adaptable entities in a VND have to be represented by discrete elements and all entities need unique identifiers. This also concerns logical objects such as the OSPF instance on an OSPF router: to support a request to bind to another (newly added) interface during the VN's lifetime, that logical OSPF instance needs to be represented in the VND.

IV. Candidate Description Methodologies

A. Network Description Language (NDL) and Network Markup Language (NML)

The NDL [4] is an ontology for computer networks based on the Resource Description Framework (RDF). It bases on the ITU-T G.805 [5] and focuses on functional modeling of networks. Alas, NDL neither covers functionalities at all, generic nodes, i.e. VMs, nor does it support management interfaces. The NDL specification team also initiated the standardization by the NML-WG of the Open Grid Forum [6] which can't be considered an independent language yet.

B. Virtual Infrastructure Description Language (VXDL)

VXDL [7] defines identifiers, supports capacities of the components, many crucial node properties, and technological definitions for links. Nevertheless, although targeting dynamic topologies, VXDL does not know the concept of interfaces. Another critical deficiency is the lack of modeling functionality executed for network connections, i.e. network and control protocols.

C. DMTF Common Information Model (CIM)

The Distributed Management Task Force (DMTF) is an IT industry organization that focuses on standardization for efficient management of IT systems. Due to the focus on real management of resources, abstract or logical descriptions are limited in the DMTF's CIM [8]. CIM focuses on nodes, their services and their control. CIM provides solutions for representing resource virtualization, e.g. for hosting VMs on physical servers, but such models are beneficial for PIP-internal resource management only. For describing networking elements some classes exist, e.g. class *MPLSLSP*. Nevertheless theses classes lack a clear hierarchical structure which hinders abstracting: *MPLSLSP* inherits from *EnabledLogicalElement* only, not from *NetworkPipe*. In general, CIM does neither provide a general representation for connections nor for their termination. Another hindering inconsistency is that CIM models *ProtocolEndpoints* explicitly, but not network interfaces. Overall, CIM could be extended to fulfill the requirements but the big deficiencies in representing networking would require major changes.

D. TMF SID

The TMF is a communications and media industry association that focuses on standardization for service provisioning. The TMF specified the Information Framework SID [2] to cover all the information required to implement service provider processes. The TMF has a close relationship to the ITU: TMF SID is derived from ITU-T M.3100 [9] and ITU Recommendation M.3190 [10] is a specification by reference to SID. The current specification of SID [11] includes information models for Physical Resources and Logical Resources. These models are focused on multi-layer networks but also provide a broad foundation for functional nodes. SID makes extensive use of inheritance and of the role concept to define the actual role of an entity, e.g. being owned by an external party. A node in the network is described by the *LogicalDevice* class which can be associated with *DeviceInterfaces*. These *DeviceInterfaces* can host *ConnectionTerminationPoints* and may be defined to run specific *Protocols*. With respect to protocols, SID already includes networking protocols as well as control protocols, even authentication credentials. Nevertheless, in order to use SID for VNDs several definitions are necessary: First, some new subclasses and properties are needed to model VN-specifics and details not covered in SID yet, e.g. Chaperone Interfaces. Second, usage directives are needed, e.g. on how to express a request for control access.

E. Evaluation

The capability summary presented in Table I shows that no candidate fulfills all requirements. Nevertheless, we deem CIM and SID being able to meet all requirements with reasonable extensions or modifications (indicated by a $\sim$). Since CIM is less clearly structured hindering abstractions and SID already provides most of the needed structures without requiring fundamental changes, we chose SID as a basis for VNDs.

Candidate	Requirement matched							
	A	B	C	D	E	F	G	H
NDL&NML	∼	✓	✓	✓	✓	✗	∼	✗
VXDL	∼	✗	✓	✓	✓	✗	✗	✗
DMTF CIM	∼	✓	✓	∼	✓	∼	∼	✗
TMF SID	✓	✓	✓	✓	✓	✓	∼	✓

TABLE I. Summary of Requirement Fulfillment

V. Modeling Approach in SID

A. Modeled Objects

Our analysis showed that all components of a VN are instances of classes derived from three base types: Transit Points to other networks, Nodes and Links between elements of these groups.

1) Links in VNs: Links in VNs are virtually direct connections between two end points which can be easily modeled by a *Connection* object in SID with its *ConnectionTerminationPoints* associated by the *TPsInPipe* association class. To terminate a *Connection*, both transit points and nodes need to provide associaions with *ConnectionTerminationPoints*, too.

2) Inter Domain Transit Points: Transit points define the physical port used as well as the logical connection, e.g. the format of the PDUs and wavelength. SID provides the *ResourcePort* object providing this bivalent meaning. We need to support expressing characteristic properties of the link between the *DeviceInterface* and the Transit Point, most importantly the delay from the Transit Point to the next object, but we can't assume any PIP willing to reveal more information on that internal point of termination. Therefore, we represent the point of handover to the neighbor domain by a *ResourcePort* associated with the foreign ownership virtually sitting at the physical domain edge. As shown in Figure 2, the foreign ownership is expressed using the role concept, while physical and logical attributes are expressed by the associating a *PhysicalPort* and

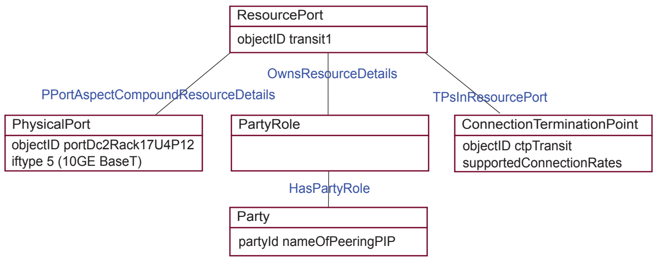

Fig. 2. Inter Domain Transit Point Model

a *ConnectionTerminationPoint*. Thus, the concept of an Inter Domain Transit Point can be expressed with existing SID classes.

3) Nodes: Nodes are represented by instances of *LogicalDevice* which are associated with *DeviceInterfaces* and *ConnectionTerminationPoints* connecting the node with the corresponding links in the topology.

The functionality provided by a node may be defined by a standardized protocol, by a standardized type of device or by the VNO providing software to be executed. In the first case, *LogicalDevice* and the respective interfaces are associated with an instance of the protocol. In the second case, a subclass of *LogicalDevice* is used unambiguously defining the functionality it provides. This approach is perfectly applicable to an OpenFlow Switch as presented in Section V-B. The case of VNO-defined functionality is not covered by SID directly. We propose to describe the provided platform, e.g. the VM, by a *LogicalDevice* hosting an *OperatingSystem* instance. An *OperatingSystem* is standardized having properties such as the total physical memory and the number of active processes. Nevertheless, other important parameters such as the non-volatile memory, the hardware architecture or the number of cores are not included in the base class, so either the base class needs to be extended or a subclass can be used. If control access shall be possible via the Chaperone Network the role concept can be used: A *DeviceInterface* is associated with the newly defined *ChaperoneInterface* class, which is derived from *DeviceInterfaceRole*.

B. Example: OpenFlow Switch with Chaperone Interface

Figure 3 shows the representation of an OpenFlow Switch with the control interface attached to an IP-based Chaperone Access (lower mid) and one exemplary operational *DeviceInterface* with a *ConnectionTerminationPoint* (upper left). SID objects are drawn black, SID associations are drawn blue and newly derived classes are drawn green. Virtualization-specific extensions, classes as well as attributes, are drawn red. Three subclasses have been created to represent the OpenFlow Protocol, the OpenFlow control method and the OpenFlow Switch. The well structured model and the consequent use of inheritance and roles in SID allow to easily model new concepts such as OpenFlow.

VI. Conclusion

Virtual Networks tailored to a specific service can provide a foundation for a prosperous market beneficial to many different players, among them today's Internet Service Providers (ISPs).

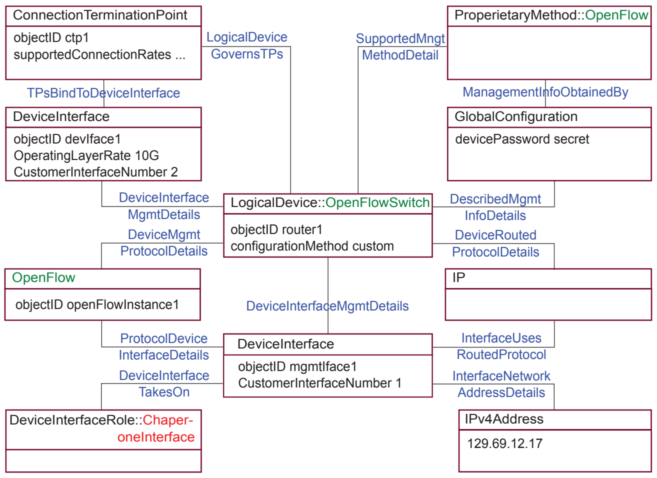
ConnectionTerminationPoint
objectID ctp1
supportedConnectionRates ...
LogicalDevice GovernsTPs
SupportedMngt MethodDetail
ProperietaryMethod::OpenFlow
TPsBindToDeviceInterface
ManagementInfoObtainedBy
DeviceInterface
objectID devIface1
OperatingLayerRate 10G
CustomerInterfaceNumber 2
GlobalConfiguration
devicePassword secret
DeviceInterface MgmtDetails
LogicalDevice::OpenFlowSwitch
objectID router1
configurationMethod custom
DescribedMgmt InfoDetails
DeviceMgmt ProtocolDetails
DeviceRouted ProtocolDetails
OpenFlow
objectID openFlowInstance1
IP
DeviceInterfaceMgmtDetails
ProtocolDevice InterfaceDetails
DeviceInterface
objectID mgmtIface1
CustomerInterfaceNumber 1
InterfaceUses RoutedProtocol
DeviceInterface TakesOn
InterfaceNetwork AddressDetails
DeviceInterfaceRole::ChaperoneInterface
IPv4Address
129.69.12.17

Fig. 3. Simplified OpenFlow Switch with Control Interface connected to the Chaperone Network

A crucial prerequisite for businesses based on offering, trading, operating or using tailored VNs is a suitable description methodology. These descriptions must allow describing VNs on different levels of detail, but have to support all details relevant for any party. We provided an analysis on the requirements for Virtual Network Descriptions in a business environment and matched existing description methodologies against these requirements. For SID from TMF as a well suitable candidate, we proposed usage directives and some extensions to make the information model fulfill all requirements. Specifically, we took control interfaces into account and showed that also innovative concepts such as OpenFlow can be properly modeled with little effort. Since SID is designed to be extended by inheritance and uses the role concept, the application of SID to VNs and the respective extensions are straightforward.

References

[1] S. Meier *et al.*, "Provisioning and Operation of Virtual Networks," *Electronic Communications of the EASST, Kommunikation in Verteilten Systemen*, vol. 37, Mar. 2011.

[2] "TM Forum Information Framework (SID)." http://www.tmforum.org/InformationFramework/1684/Home.html.

[3] "OpenFlow Switch Specification Version 1.1.0 Implemented," Feb. 2011. http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf.

[4] van der Ham *et al.*, "Using RDF to describe networks," *Future Gener. Comput. Syst.*, vol. 22, pp. 862–867, Oct. 2006.

[5] ITU-T, "Recommendation G.805 - Generic functional architecture of transport networks," Mar. 2000.

[6] Network Mark-up Language Working Group, *Project Home*. https://forge.ogf.org/sf/projects/nml-wg.

[7] G. P. Koslovski *et al.*, "VXDL: Virtual Resources and Interconnection Networks Description Language," in *Networks for Grid Applications* (Vicat-Blanc Primet and others, ed.), vol. 2 of *LNICST*, pp. 138–154, Springer Berlin Heidelberg, 2009.

[8] DMTF, "Common Information Model." Version 2.31.0, http://dmtf.org/standards/cim.

[9] ITU-T, "Recommendation M.3100 - Generic Network Information Model," Apr. 2005.

[10] ITU-T, "Recommendation M.3190 - Shared information and data model (SID)," July 2008.

[11] "GB922 Information Framework (SID) Suite Release 9.5." http://www.tmforum.org/DocumentCenter/10365/home.html#TRCDocuments/artf2301.