

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: September 6, 2018

J. Arkko  
Ericsson  
J. Tantsura  
Nuagenetworks  
J. Halpern  
B. Varga  
Ericsson  
March 5, 2018

Considerations on Network Virtualization and Slicing  
draft-arkko-arch-virtualization-01

Abstract

This document makes some observations on the effects of virtualization on Internet architecture, as well as provides some guidelines for further work at the IETF relating to virtualization.

This document also provides a summary of IETF technologies that relate to network virtualization. An understanding of what current technologies there exist and what they can or cannot do is the first step in developing plans for possible extensions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Definitions . . . . .	3
3. General Observations . . . . .	4
4. Virtualization in 5G Networks . . . . .	6
5. Overview of IETF Virtualization Technologies . . . . .	6
5.1. Selection of Virtual Instances . . . . .	7
5.2. Traffic Separation in VPNs . . . . .	7
5.3. Traffic Engineering and QoS . . . . .	9
5.4. Service Chaining . . . . .	10
5.5. Management Frameworks and Data Models . . . . .	10
6. Architectural Observations . . . . .	12
7. Further Work . . . . .	14
8. Acknowledgements . . . . .	17
9. Informative References . . . . .	17
Authors' Addresses . . . . .	19

## 1. Introduction

Network virtualization is network management pertaining to treating different traffic categories in separate virtual networks, with independent lifecycle management and resource, technology, and topology choices.

This document makes some observations on the effects of virtualization on Internet architecture, as well as provides some guidelines for further work at the IETF relating to virtualization.

This document also provides a summary of IETF technologies that relate to network virtualization. An understanding of what current technologies there exist and what they can or cannot do is the first step in developing plans for possible extensions.

In particular, many IETF discussions earlier in the summer of 2017 started from a top-down view of new virtualization technologies, but were often unable to explain the necessary delta to the wealth of existing IETF technology in this space. This document takes a different, bottom-up approach to the topic and attempts to document existing technology, and then identify areas of needed development.

In particular, whether one calls a particular piece of technology "virtualization", "slicing", "separation", or "network selection" does not matter at the level of a system. Any modern system will use several underlying technology components that may use different terms but provide some separation or management. So, for instance, in a given system you may use VLAN tags in an ethernet segment, MPLS or VPNs across the domain, NAIs to select the right AAA instance, and run all this top of virtualized operating system and software-based switches. As new needs are being recognised in the developing virtualization technology, what should drive the work is the need for specific capabilities rather than the need to distinguish a particular term from another term.

## 2. Definitions

Network function virtualization is defined in Wikipedia as follows:

"Network function virtualization or NFV is a network architecture concept that uses the technologies of IT virtualization to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create communication services.

NFV relies upon, but differs from, traditional server-virtualization techniques, such as those used in enterprise IT. A virtualized network function, or VNF, may consist of one or more virtual machines running different software and processes, on top of standard high-volume servers, switches and storage devices, or even cloud computing infrastructure, instead of having custom hardware appliances for each network function."

We should not confuse NFV and network virtualization, the former, as the name suggests is about functions virtualization, and not the network.

The idea of network virtualization is almost as old as the networking technology itself. Network virtualization is hierarchical and multilayer in its nature, from layer 1 up to services on top. When talking about virtualization we usually define overlay to underlay relationship between different layers, bottom up. A VPN (Virtual Private Network) [RFC4026] is the most common form of network virtualization. The general benefits and desirability of VPNs have been described many times and in many places ([RFC4110] and [RFC4664]).

The only immutable infrastructure is the "physical" medium, that could be dedicated or "sliced" to provide services (VPNs) in a multi-tenant environment.

The term slicing has been used to describe a virtualization concept in planned 5G networks. The 3GPP architecture specification [TS-3GPP.23.501] defines network slices as having potentially different "supported features and network functions optimisations", and spanning functions from core network to radio access networks.

[I-D.king-teas-applicability-actn-slicing] defined slicing as "an approach to network operations that builds on the concept of network abstraction to provide programmability, flexibility, and modularity. It may use techniques such as Software Defined Networking (SDN) and Network Function Virtualization (NFV) to create multiple logical (virtual) networks, each tailored for a set of services that are sharing the same set of requirements, on top of a common network.

And, [I-D.geng-coms-problem-statement] defines slicing as a management mechanism that an service provider can use to allocate dedicated network resources from shared network infrastructures to a tenant.

### 3. General Observations

#### Software vs. Protocols

Many of the necessary tools for using virtualization are software, e.g., tools that enable running processes or entire machines in a virtual environment decoupled from physical machines and isolated from each other, virtual switches that connect systems together, management tools to set up virtual environments, and so on. From a communications perspective these tools operate largely in the same fashion as their real-world counterparts do, except that there may not be wires or other physical communication channels, and that connections can be made in the desired fashion.

In general, there is no reason for protocols to change just because a function or a connection exists on a virtual platform. However, sometimes there are useful underlying technologies that facilitate connection to virtualized systems, or optimised or additional tools that are needed in the the virtualized environment.

For instance, many underlying technologies enable virtualization at hardware or physical networking level. For instance, Ethernet networks have Virtual LAN (VLAN) tags and mobile networks have a choice of Access Point Names (APNs). These techniques allow users and traffic to be put on specific networks, which in turn may comprise of virtual components.

Other examples of protocols providing helpful techniques include virtual private networking mechanisms or management mechanisms and data models that can assist in setting up and administering virtualized systems.

There may also be situations where scaling demands changes in protocols. An ability to replicate many instances may push the limits of protocol mechanisms that were designed primarily or originally for physical networks.

#### Selection vs. Creation and Orchestration

Two primary tasks in virtualization should be differentiated: selection of a particular virtual instance, and the tasks related to how that virtual instance was created and continues to be managed.

Selection involves choosing a particular virtual instance, or an endpoint to a virtual network. In its simplest form, a customer could be hardwired by configuration to a particular virtual instance. In more complex cases, the connecting devices may have some settings that affect the choice. In the general case, both the connecting devices and the network they are connecting to it have a say in the choice.

The selection choice may even be dynamic in some cases. For instance, traffic pattern analysis may affect the selection.

Typically, however, connecting devices do not have a say in what the virtual instance does. This is directed by the network operator and its customers. An instance is specified, created, and needs to be continuously managed and orchestrated. The creation can be manual and occur rarely, or be more dynamic, e.g., an instance can actually be instantiated automatically, and only when the first connecting device connects to it.

#### Protocols vs. Representations of Virtual Networks

Some of virtualization technology benefits from protocol support either in the data or control plane. But there are also management constructs, such as data models representing virtual services or networks and data models useful in the construction of such services.

There are also conceptual definitions that may be needed when constructing either protocols or data models or when discussing service agreements between providers and consumers.

#### 4. Virtualization in 5G Networks

Goals for the support of virtualization in 5G relate to both the use of virtualized network functions to build the 5G network, and to enabling the separation of different user or traffic classes into separate network constructs called slices.

Slices enable a separation of concerns, allow the creation of dedicated services for special traffic types, allow faster evolution of the network mechanisms by easing gradual migration to new functionality, and enable faster time to market for new new functionality.

In 5G, slice selection happens as a combination of settings in the User Equipment (UE) and the network. Settings in the UE include, for instance, the Access Point Name (APN), Dedicated Core Network Indicator (DCN-ID) [TS-3GPP.23.401], and, with 5G, a slice indicator (Network Slice Selection Assistance Information or NSSAI) [TS-3GPP.23.501]. This information is combined with the information configured in the network for a given subscriber and the policies of the networks involved. Ultimately, a slice is selected.

A 5G access network carries a user's connection attempt to the 5G core network and the Access Management Function (AMF) network function. This function collects information provided by the UE and the subscriber database from home network, and consults the Network Slice Selection Function (NSSF) to make a decision of the slice selected for the user. When the selection has been made, this may also mean that the connection is moved to a different AMF; enabling separate networks to have entirely different network-level service.

The creation and orchestration of slices does not happen at this signalling plane, but rather the slices are separately specified, created, and managed, typically with the help of an orchestrator function.

The exact mechanisms for doing this continue to evolve, but in any case involve multiple layers of technology, ranging from underlying virtualization software to network component configuration mechanisms and models (often in YANG) to higher abstraction level descriptions (often in TOSCA), to orchestrator software.

#### 5. Overview of IETF Virtualization Technologies

General networking protocols are largely agnostic to virtualization. TCP/IP does not care whether it runs on a physical wire or on a computer-created connection between virtual devices.

As a result, virtualization generally does not affect TCP/IP itself or applications running on top. There are some exceptions, though, such as when the need to virtualize has caused previously held assumptions to break, and the Internet community has had to provide new solutions. For instance, early versions of the HTTP protocol assumed a single host served a single website. The advent of virtual hosting and pressure to not use large numbers of IPv4 addresses lead to HTTP 1.1 adopting virtual hosting, where the identified web host is indicated inside the HTTP protocol rather than inferred from the reception of a request at particular IP address [VirtualHosting] [RFC2616].

But where virtualization affects the Internet architecture and implementations is at lower layers, the physical and MAC layers, the systems that deal with the delivery of IP packets to the right destination, management frameworks controlling these systems, and data models designed to help the creation, monitoring, or management of virtualized services.

What follows is an overview of existing technologies and technologies currently under development that support virtualization in its various forms.

#### 5.1. Selection of Virtual Instances

Some L2 technology allows the identification of traffic belonging to a particular virtual network or connection. For instance, Ethernet VLAN tags.

There are some IETF technologies that also allow similar identification of connections setup with the help of IETF protocols. For instance, Network Access Identifiers may identify a particular customer or virtual service within AAA, EAP or IKEv2 VPN connections.

#### 5.2. Traffic Separation in VPNs

Technologies that assist separation and engineering of networks include both end-point and provider-based VPNs. End-point VPN technologies include, for instance, IPsec-based VPNs [RFC4301].

For providing virtualized services, however, provider-based solutions are often the most relevant ones. L1VPN facilitates virtualization of the underlying L0 "physical" medium. L2[IEEE802.1Q] facilitates virtualization of the underlying Ethernet network Tunneling over IP (MPLS, GRE, VxLAN, IPinIP, L2TP, etc) facilitates virtualization of the underlying IP network - MPLS LSP's - either traffic engineered or not belong here L2VPN facilitates virtualization of a L2 network L3VPN facilitates virtualization of a L3 network.

The IETF has defined a multiplicity of technologies that can be used for provider-based VPNs. The technologies choices available can be described along two axes, control mechanisms and dataplane encapsulation mechanisms. The two are not completely orthogonal.

In the data plane, for provider based VPNs, the first important observation is that the most obvious encapsulation is NOT used. While IPSec could be used for provider-based VPNs, it does not appear to be used in practice, and is not the focus for any of the available control mechanisms. Often, when end2end encryption is required it is used as an overlay over MPLS based L3VPN

The common encapsulation for provider-based VPNs is to use MPLS. This is particularly common for VPNs within one operator, and is sometimes supported across operators.

Keyed GRE can be used, particularly for cross-operator cases. However, it seems to be rare in practice.

The usage of MPLS for provider-based VPNs generally follows a pattern of using two (or more) MPLS labels, top (transport) label to represent the remote end point/egress provider-edge device, and bottom (service) label to signal the different VPNs on the remote end point. Using TE might result in a deeper label stack.

L2 VPNs could be signaled thru LDP[RFC4762] or MP-BGP[RFC4761], L3 VPN is signaled thru MP-BGP[RFC4364]

The LDP usage to control VPN establishment falls within the PALS working group, and is used to establish pseudo-wires to carry Ethernet (or lower layer) traffic. The Ethernet cases tend to be called VPLS (Virtual Private LAN Service) for multi-point connectivity and VPWS (Virtual Private Wire Service) for point-to-point connectivity. These mechanism do augment the data plane capabilities with control words that support additional features. In operation, LDP is used to signal the communicating end-points that are interested in communicating with each other in support of specific VPNs. Information about the MAC addresses used behind the provider edges is exchanged using classic Ethernet flooding technology. It has been proposed to use BGP to bootstrap the exchange of information as to who the communicating endpoints are.

BGP can be used to establish Layer 2 or Layer 3 VPNs. Originally, the BGP based MPLS VPN technology was developed to support layer 3 VPNs. the BGP exchanges uses several different features in MP-BGP (specifically route distinguishers and route targets) to control the distribution of information about VPN end-points. The BGP information carries the VPN IP address prefixes, and the MPLS labels



to be used to represent the VPN. This technology combination is generally known as L3VPN.

This usage of BGP for VPNs has been extended to support Layer 2 VPNs. This is known as EVPN. The BGP exchanges are used to carry the MAC address reachability behind each provider edge router, providing an Ethernet multipoint service without a need to flood unknown-destination Ethernet packets.

In theory, the BGP mechanisms can also be used to support other tunnels such as keyed GRE. That is not widely practiced.

There are also hybrid variations, such as adding an ARP / ND proxy service so that an L3VPN can be used with an L2 Access, when the only desired service is IP.

### 5.3. Traffic Engineering and QoS

Traffic Engineering (TE) is the term used to refer to techniques that enable operators to control how specific traffic flows are treated within their networks.

The TEAS working group works on enhancements to traffic-engineering capabilities for MPLS and GMPLS networks:

TE is applied to packet networks via MPLS TE tunnels and LSPs. The MPLS-TE control plane was generalized to additionally support non-packet technologies via GMPLS. RSVP-TE is the signaling protocol used for both MPLS-TE and GMPLS.

The TEAS WG is responsible for:

- \* Traffic-engineering architectures for generic applicability across packet and non-packet networks.
- \* Definition of protocol-independent metrics and parameters.
- \* Functional specification of extensions for routing (OSPF, ISIS), for path computation (PCE), and RSVP-TE to provide general enablers of traffic-engineering systems.
- \* Definition of control plane mechanisms and extensions to allow the setup and maintenance of TE paths and TE tunnels that span multiple domains and/or switching technologies.

A good example of work that is currently considered in the TEAS WG is the set of models that detail earlier IETF-developed topology models with both traffic engineering information and connection to what

services are running on top of the network  
[I-D.bryskin-teas-use-cases-sf-aware-topo-model]  
[I-D.bryskin-teas-sf-aware-topo-model]. These models enable reasoning about the state of the network with respect to those services, and to set up services with optimal network connectivity.

Traffic engineering is a common requirement for many routing systems, and also discussed, e.g., in the context of LISP.

#### 5.4. Service Chaining

The SFC working group has defined the concept of Service Chaining:

Today, common deployment models have service functions inserted on the data-forwarding path between communicating peers. Going forward, however, there is a need to move to a different model, where service functions, whether physical or virtualized, are not required to reside on the direct data path and traffic is instead steered through required service functions, wherever they are deployed.

For a given service, the abstracted view of the required service functions and the order in which they are to be applied is called a Service Function Chain (SFC). An SFC is instantiated through selection of specific service function instances on specific network nodes to form a service graph: this is called a Service Function Path (SFP). The service functions may be applied at any layer within the network protocol stack (network layer, transport layer, application layer, etc.).

#### 5.5. Management Frameworks and Data Models

There have been two working groups at the IETF, focusing on data models describing VPNs. The IETF and the industry in general is currently specifying a set of YANG models for network element and protocol configuration [RFC6020].

YANG is a powerful and versatile data modeling language that was designed from the requirements of network operators for an easy to use and robust mechanism for provisioning devices and services across networks. It was originally designed at the Internet Engineering Task Force (IETF) and has been so successful that it has been adopted as the standard for modeling design in many other standards bodies such as the Metro Ethernet Forum, OpenDaylight, OpenConfig, and others. The number of YANG modules being implemented for interfaces, devices, and service is growing rapidly.

(It should be noted that there are also other description formats, e.g., Topology and Orchestration Specification for Cloud Applications (TOSCA) [TOSCA-1.0] [TOSCA-Profile-1.1], common in many higher abstract level network service descriptions. The ONAP open source project plans to employ it for abstract mobile network slicing models, for instance.)

A service model is an abstract model, at a higher level than network element or protocol configuration. A service model for VPN service describes a VPN in a manner that a customer of the VPN service would see it.

It needs to be clearly understood that such a service model is not a configuration model. That is, it does not provide details for configuring network elements or protocols: that work is expected to be carried out in other protocol-specific working groups. Instead, service models contain the characteristics of the service as discussed between the operators and their customers. A separate process is responsible for mapping this customer service model onto the protocols and network elements depending on how the network operator chooses to realise the service.

The L2SM WG specifies a service model for L2-based VPNs:

The Layer Two Virtual Private Network Service Model (L2SM) working group is a short-lived WG. It is tasked to create a YANG data model that describes a L2VPN service (a L2VPN customer service model). The model can be used for communication between customers and network operators, and to provide input to automated control and configuration applications.

It is recognized that it would be beneficial to have a common base model that addresses multiple popular L2VPN service types. The working group derives a single data model that includes support for the following:

- \* point-to-point Virtual Private Wire Services (VPWS),
- \* multipoint Virtual Private LAN services (VPLS) that use LDP-signaled Pseudowires,
- \* multipoint Virtual Private LAN services (VPLS) that use a Border Gateway Protocol (BGP) control plane as described in [RFC4761] and [RFC6624],
- \* Ethernet VPNs specified in [RFC7432].

Other L2VPN service types may be included if there is consensus in the working group.

Similarly, the L3SM WG specified a service model for L3-based VPNs.

The Layer Three Virtual Private Network Service Model (L3SM) working group is a short-lived WG tasked to create a YANG data model that describes a L3VPN service (a L3VPN service model) that can be used for communication between customers and network operators, and to provide input to automated control and configuration applications.

It needs to be clearly understood that this L3VPN service model is not an L3VPN configuration model. That is, it does not provide details for configuring network elements or protocols. Instead it contains the characteristics of the service.

## 6. Architectural Observations

This section makes some observations about architectural trends and issues.

### Role of Software

An obvious trend is that bigger and bigger parts of the functionality in a network is driven by software, e.g., orchestration or management tools that figure out how to control relatively simple network element functionality. The software components are where the intelligence is, and a smaller fraction of the intelligence resides in network elements, nor is the intelligence encoded in the behaviour rules of the protocols that the network elements use to communicate with each other.

### Centralization of Functions

An interesting architectural trend is that virtualization and data /software driven networking technologies are driving network architectures where functionality moves towards central entities such as various controllers, path computation servers, and orchestration systems.

A natural consequence of this is the simplification (and perhaps commoditization) of network elements, while the "intelligent" or higher value functions migrate to the center.

The benefits are largely in the manageability, control, and speed of change. There are, however, potential pitfalls to be aware of as well. First off, networks need to continue to be operate even

under partial connectivity situations and breakage, and it is key that designs can handle those situations as well.

And it is important that network users and peers continue to be able to operate and connect in the distributed, voluntary manner that we have today. Today's virtualization technology is primarily used to manage single administrative domains and to offer specific service to others. One could imagine centralised models being taken too far as well, limiting the ability of other network owners to manage their own networks.

#### Tailored vs. general-purpose networking

The interest in building tailored solutions, tailored Quality-of-Service offerings vs. building general-purpose "low touch" networks seems to fluctuate over time.

It is important to find the right balance here. From an economics perspective, it may not be feasible to provide specialised service -- at least if it requires human effort -- for large fraction of use cases. Even if those are very useful in critical applications.

#### Need for descriptions

As networks deal more and more with virtual services, there arises a need to have generally understood, portable descriptions of these service. Hence the creation of YANG data models representing abstract VPN services, for instance.

We can also identify some potential architectural principles, such as:

#### Data model layering

Given the heterogeneity of networking technologies and the differing users that data models are being designed for, it seems difficult to provide a single-level model. It seems preferable to construct a layered set of models, for instance abstract, user-facing models that specify services that can then be mapped to concrete configuration model for networks. And these can in turn be mapped to individual network element configuration models.

Getting this layered design right is crucial for our ability to evolve a useful set of data models.

#### Ability to evolve modelling tools and mapping systems

The networks and their models are complex, and mapping from high abstraction level specifications to concrete network configurations is a hard problem.

It is important that each of the components can evolve on its own. It should be possible to plug in a new language that represents network models better. Or replace a software component that performs mapping between layers to one that works better.

While this should normally be possible, there's room to avoid too tight binding between the different aspects of a system. For instance, abstraction layers within software can shield the software from being too closely tied with a particular representation language.

Similarly, it would be an advantage to develop algorithms and mapping approaches separately from the software that actually does that, so that another piece of software could easily follow the same guidelines and provide an alternate implementation. Perhaps there's an opportunity for specification work to focus more on processing rules than protocol behaviours, for instance.

#### General over specific

In the quick pace of important developments, it is tempting to focus on specific concepts and service offerings such as 5G slicing.

But a preferable approach seems to provide general-purpose tools that can be used by 5G and other networks, and whose longevity exceeds that of a version of a specific offering. The quick development pace is likely driving the evolution of concepts in any case, and building IETF tools that provide the ability to deal with different technologies is most useful.

#### 7. Further Work

There may be needs for further work in this area at the IETF. Before discussing the specific needs, it may be useful to classify the types of useful work that might come to question. And perhaps also outline some types of work that is not appropriate for the IETF.

The IETF works primarily on protocols, but in many cases also with data models that help manage systems, as well as operational guidance documents. But the IETF does not work on software, such as abstractions that only need to exist inside computers or ones that do not have an effect on protocols either on real or simulated "wires".

The IETF also does not generally work on system-level design. IETF is best at designing components, not putting those components together to achieve a particular purpose or build a specific application.

As a result, IETF's work on new systems employing virtualization techniques (such as 5G slicing concept) is more at the component improvement level than at the level of the concept. There needs to be a mapping between a vision of a system and how it utilizes various software, hardware, and protocol tools to achieve the particular virtualization capabilities it needs to. Developing a new concept does not necessarily mean that entirely new solutions are needed throughout the stack. Indeed, systems and concepts are usually built on top of solid, well defined components such as the ones produced by the IETF.

That mapping work is necessarily something that those who want to achieve some new functionality need to do; it is difficult for others to take a position on what the new functionality is. But at the same time, IETF working groups and participants typically have a perspective on how their technology should develop and be extended. Those two viewpoints must meet.

The kinds of potential new work in this space falls generally in the following classes:

#### Virtualization selectors

Sometimes protocols need mechanisms that make it possible to use them as multiple instances. E.g., VLAN tags were added to Ethernet frames, NAIs were added to PPP and EAP, and so on. These cases are rare today, because most protocols and mechanisms have some kind of selector that can be used to run multiple instances or connect to multiple different networks.

#### Traffic engineering

A big reason for building specific networks for specific purposes is to provide an engineered service level on delay and other factors to the given customer. There are a number of different tools in the IETF to help manage and engineer networks, but it is also an area that continues to develop and will likely see new functionality.

#### Virtual service data models

Data models -- such as those described by L2SM or L3SM working groups can represent a "service" offered by a network, a setup built for a specific customer or purpose.

Some specific areas where work is likely needed include:

- o The ability to manage heterogenous technologies, e.g., across SDN and traditionally built networks, or manage both general-purpose and very technology-specific parameters such as those associated with 5G radio.
- o The ability to specify "statistical" rather than hard performance parameters. In some networks -- notably with wireless technology -- recent advances have made very high peak rates possible, but with increased bursty-ness of traffic and with potential bottlenecks on the aggregation parts of the networks. The ability to specify statistical performance in data models and in VPN configuration would be important, over different timescales and probabilities.
- o Mapping from high abstraction level specifications to concrete network configurations.

There is a lot of work on data models and templates at various levels and in different representations. There are also many systems built to manage these models and orchestrate network configuration. But the mapping of the abstract models to concrete network configurations remains a hard problem, and it certainly will need more work.

There are even some questions about how to go about this. Is it enough that we specify models, and leave the mapping to "magic" of the software? Are the connections something that different vendors compete in producing good products in? Or are the mapping algorithms something that needs to be specified together, and their ability to work with different types of network equipment verified in some manner?

- o Cross-domain: A big problem is that we have little tools for cross-domain management of virtualized networks and resources.

Finally, there is a question of where all this work should reside. There's an argument that IETF-based virtualization technologies deserve proper management tools, including data models.

And there's another argument that with the extensive use of virtualization technology, solutions that can manage many different networks should be general, and as such, potential IETF work



material. Yet, the IETF is not and should not be in the space of replacing various tools and open source toolkits that have been created for managing virtualization. It seems though that work on commonly usable data models at several layers of abstraction would be good work at the IETF.

Nevertheless, the IETF should understand where the broader community is and what tools they use for what purpose, and try to help by building on those components. Virtualization and slicing are sometimes represented as issues needing a single solution. In reality, they are an interworking of a number of different tools.

## 8. Acknowledgements

The authors would like to thank Gonzalo Camarillo, Gabriel Montenegro, Alex Galis, Adrian Farrell, Liang Geng, Yi Zhao, Hannu Flinck, Yi Zhao, Barry Leiba, Georg Mayer, Benoit Claise, Daniele Ceccarelli, Warren Kumari, Ted Hardie, and many others for interesting discussions in this problem space.

## 9. Informative References

- [CC2015] claffy, kc. and D. Clark, "Adding Enhanced Services to the Internet: Lessons from History", September 2015 ([https://www.caida.org/publications/papers/2015/adding\\_enhanced\\_services\\_internet/adding\\_enhanced\\_services\\_internet.pdf](https://www.caida.org/publications/papers/2015/adding_enhanced_services_internet/adding_enhanced_services_internet.pdf)).
- [I-D.bryskin-teas-sf-aware-topo-model]  
Bryskin, I. and X. Liu, "SF Aware TE Topology YANG Model", draft-bryskin-teas-sf-aware-topo-model-01 (work in progress), March 2018.
- [I-D.bryskin-teas-use-cases-sf-aware-topo-model]  
Bryskin, I., Liu, X., Guichard, J., Lee, Y., Contreras, L., and D. Ceccarelli, "Use Cases for SF Aware Topology Models", draft-bryskin-teas-use-cases-sf-aware-topo-model-02 (work in progress), March 2018.
- [I-D.geng-coms-problem-statement]  
67, 4., Slawomir, S., Qiang, L., Matsushima, S., Galis, A., and L. Contreras, "Problem Statement of Supervised Heterogeneous Network Slicing", draft-geng-coms-problem-statement-00 (work in progress), September 2017.
- [I-D.ietf-sfc-nsh]

Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-28 (work in progress), November 2017.

- [I-D.king-teas-applicability-actn-slicing]  
King, D. and Y. Lee, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing", draft-king-teas-applicability-actn-slicing-01 (work in progress), July 2017.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <<https://www.rfc-editor.org/info/rfc2616>>.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <<https://www.rfc-editor.org/info/rfc4026>>.
- [RFC4110] Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, DOI 10.17487/RFC4110, July 2005, <<https://www.rfc-editor.org/info/rfc4110>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.
- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <<https://www.rfc-editor.org/info/rfc6624>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC8049] Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8049, DOI 10.17487/RFC8049, February 2017, <<https://www.rfc-editor.org/info/rfc8049>>.
- [TOSCA-1.0]  
OASIS, "Topology and Orchestration Specification for Cloud Applications Version 1.0", OASIS OASIS Standard, <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html>, November 2013.
- [TOSCA-Profile-1.1]  
OASIS, "TOSCA Simple Profile in YAML Version 1.1", OASIS OASIS Standard, <http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.1/TOSCA-Simple-Profile-YAML-v1.1.html>, January 2018.
- [TS-3GPP.23.401]  
3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access; (Release 15)", 3GPP Technical Specification 23.401, December 2017.
- [TS-3GPP.23.501]  
3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture and procedures for 5G System; (Release 15)", 3GPP Technical Specification 23.501, December 2017.
- [VirtualHosting]  
Wikipedia, "Virtual Hosting", Wikipedia article [https://en.wikipedia.org/wiki/Virtual\\_hosting](https://en.wikipedia.org/wiki/Virtual_hosting), August 2017.

Authors' Addresses

Jari Arkko  
Ericsson  
Kauniainen 02700  
Finland

Email: jari.arkko@piuha.net

Jeff Tantsura  
Nuagenetworks

Email: jefftant.ietf@gmail.com

Joel Halpern  
Ericsson

Email: joel.halpern@ericsson.com

Balazs Varga  
Ericsson  
Budapest 1097  
Hungary

Email: balazs.a.varga@ericsson.com