Network Virtualization: Technologies, Perspectives, and Frontiers

Anjing Wang, *Member, IEEE*, Mohan Iyer, Rudra Dutta, George N. Rouskas, *Fellow, IEEE*, and Ilia Baldine, *Member, IEEE*

(Invited Tutorial)

***Abstract*—Network virtualization refers to a broad set of technologies. Commercial solutions have been offered by the industry for years, while more recently the academic community has emphasized virtualization as an enabler for network architecture research, deployment, and experimentation. We review the entire spectrum of relevant approaches with the goal of identifying the underlying commonalities. We offer a unifying definition of the term "network virtualization" and examine existing approaches to bring out this unifying perspective. We also discuss a set of challenges and research directions that we expect to come to the forefront as network virtualization technologies proliferate.**

***Index Terms*—Network architecture, network virtualization.**

I. Introduction

NETWORK virtualization has become a popular topic in recent years, and it is often mentioned in technical magazines, network device providers' white papers, textbooks and research papers. Survey papers [1], [2] provide an overview of network virtualization, and online fora (e.g., http://networkvirtualization.com) have become promoters and advocates of network virtualization. Nevertheless, the former typically reflect the point of view of the research community, addressing only some aspects of the field, while the latter focus mostly on the industry perspective and do not cover the entire spectrum of network virtualization either. With its rapid adoption in different contexts to refer to a range of concepts and approaches, the term "network virtualization" has become overloaded, often to the point of confusion.

With this paper, we attempt to review the entire spectrum of network virtualization technologies, perspectives, and practices with a focus on identifying common features. Specifically, we aim to give clear answers to fundamental questions such as "*What is network virtualization?*" and "*What is virtualized?*" Our objective is twofold: that this work serve as an essential tutorial for readers interested in learning about specific network virtualization approaches; and to present a comprehensive picture from a new perspective that unifies seemingly diverse points of view.

The rest of this paper is organized as follows. In Section II, we review mature network virtualization-related technologies that are in commercial use. In Section III, we discuss network virtualization projects and prototypes mainly driven by the research community. We discuss several interpretations of network virtualization in Section IV, and we converge to a broad definition that captures its role as abstraction of resources. We discuss emerging technologies, research directions and challenges in network virtualization in Section V, and we conclude the paper in Section VI.

Manuscript received May 21, 2012; revised August 13, 2012; accepted August 14, 2012. Date of publication August 17, 2012; date of current version January 09, 2013. This work was supported in part by the National Science Foundation under Grants CNS-1111088 and CNS-1111256.

A. Wang is with Ericsson, San Jose, CA 95134 USA.

M. Iyer is with Oracle Corporation, Santa Clara CA 95054, USA.

R. Dutta and G. N. Rouskas are with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 USA (e-mail: rouskas@ncsu.edu).

I. Baldine is with the Renaissance Computing Institute, University of North Carolina at Chapel Hill, Chapel Hill, NC 27517 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2012.2213796

II. The Industry Perspective

In this section, we examine fundamental network virtualization-related technologies that are already in commercial use. We discuss technologies related to devices, links, and networks, and explain how they influence each other and evolve together.

A. Network Device Virtualization

We start by reviewing virtualization technologies of the fundamental building blocks of a network, namely, network interface cards (NICs, at the network edge) and routers (in the network core).

1) NIC Virtualization:

a) Software-Enabled NIC Virtualization: VMware, Microsoft, Citrix Systems (providing Xen [3]), and Oracle are among the main commercial providers of operating system (OS) virtualization solutions. An important task of such platforms is the sharing of NIC hardware among instances of the virtual OS. Fig. 1 illustrates a general architecture of NIC virtualization, although the specific implementation may vary across vendors. The cornerstone of NIC virtualization is *virtual NIC (vNIC)*, a software emulation of a physical NIC which may be assigned its own, dedicated IP and MAC addresses. In the figure, a vNIC client can be any client who wishes to use a vNIC. The most common vNIC clients include a virtual OS (which may be called a virtual machine (VM) [4] in VMware, or a domain [3] in Xen) or an OS-level virtualization instance, such as a Solaris Zone [5].

0733-8724/$31.00 © 2012 IEEE

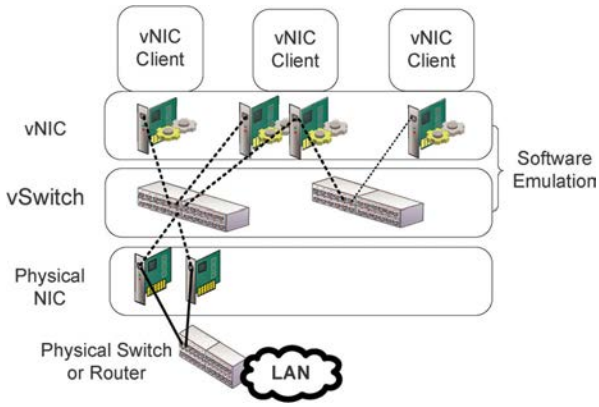

Fig. 1. General NIC Virtualization Architecture.

A virtual switch (vSwitch) in Fig. 1 is a software emulation of a physical switch which, however, may not support all features of a physical switch [6]. A vSwitch performs functions such as traffic switching, multiplexing, and scheduling and bridges vNICs with physical NIC(s) if needed. The links between vNIC and vSwitch are software-emulated links (not to be confused with the "virtual link" concept we describe in Section II.B). The bandwidth of these emulated links is only limited by the processing capabilities of the host itself. While it is possible to set an upper limit on the speed of each emulated link so as to maintain the overall balance of traffic, most NIC virtualizaton implementations do not support guaranteed bandwidth. A notable exception is the Sun Crossbow project [7]. Crossbow provides a guaranteed bandwidth feature that allows vNICs to reserve a hardware traffic lane, and also offers a more elegant framework to manage resources. All these abstractions reside either in the hypervisor (a program that manages OS resources to allow virtualization) or the host OS.

In Fig. 1, if the vNIC clients are servers (e.g., web servers, DNS servers or firewalls), NIC virtualization actually provides a virtual network composed of virtual servers, virtual NICs, virtual switches, and virtual links. This capability is often called "network-in-a-box." In this context, the virtual network is actually a software-emulated network which generates traffic that is injected to the real world through a non-virtual/non-emulated physical NIC.

With vSwitch, the communications between vNIC clients within one hypervisor are not visible to the outside world anymore, and need additional effort to be monitored or regulated. Network administrators are also burdened as different server virtualization technologies provide different flavors of vSwitch. Thus, it is also desired to have all traffic directed to the physical switch even if it is destined back to the same physical server. Two approaches could be used to address the problem, and both are under the standardization process. Virtual Ethernet Port Aggregator (VEPA) is also known as 802.1Qbg [8]. It defines standard mode and multi-channel mode, which uses standard 802.1ad Q-in-Q. In the other approach, VN-Tag, known as 802.1Qbh [9], introduces a new VN-Tag field and uses it to identify virtual interface. With the additional tag, a bridge port is virtually extended, and could be easily cascaded as well.

To eliminate the burden of configuring vSwitch on every server, Cisco provides software-based Nexus 1000v. It has

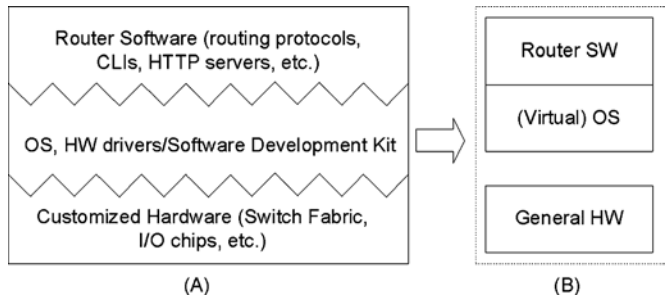

Fig. 2. Routers in Virtual OS.

two major components: Virtual Ethernet Module (VEM) and Virtual Supervisor Module (VSM). VEM is integrated with the hypervisor, and perform switching, QoS, port security, etc. VSM is the management module, and is able to manage up to 64 VEM to be one logical switch. Readers are referred to [10] for more information.

b) Hardware-Aided NIC Virtualization: Regular NIC only provides one peripheral component interconnect express (PCIe) channel, which usually becomes the I/O bottleneck in a VM centric data server. Single root I/O virtualization (SR-IOV) [11] is a hardware enhancement that could create multiple instances of PCI functions by its Virtual Functions (VFs). Instead of connecting vNIC clients to a vSwitch, every VM could be directly mapped to a VF for direct access of NIC resource. This module generally provides better throughput, scalability and lower CPU utilization [12].

2) Router Virtualization: In this section, we use "router" as a general term to refer to a network device that performs routing or switching operations. The router virtualization technologies described in the following three subsections represent distinct and fundamentally different functionalities.

a) Routers in Virtual OS: Fig. 2(A) shows how software and hardware are tightly coupled together in a typical router. The OS is often customized to the specially designed hardware for better performance; examples include Cisco's Internetwork Operating System (IOS), VxWorks, Linux, and BSD. Router software, including routing protocols, runs within such a customized OS.

There have been numerous efforts to separate router software from hardware, as in Fig. 2(B), especially in some Linux distributions, such as Alpine Linux, Mikrotik RouterOS, Untangle and Vyatta. Most of these distributions can run on a standard X86 hardware architecture, and can be installed in a virtual OS, including VMware or Xen.

Routers in virtual OS may be called "virtual routers"; in this context, the term "virtual" refers to the fact that these routers share hardware resources, such as CPU, memory, hard drive, NIC, etc., with other instances of any virtual OS in the same physical machine. In the virtualization scenario of Fig. 1, these virtual routers are vNIC clients and reinforce the concept of "network-in-a-box" described in Section II.A.1a.

b) Router Control/Data Plane Virtualization: A core component of routers is the routing table that maps incoming packets to output ports. Some routers may have a single routing table for all packets, with the routing table being maintained by a single process. Some routers may have multiple routing tables, each

table serving a different routing context. The various routing tables may be maintained by a single process or by multiple processes (e.g., one process for each routing table). This technology is referred to as "virtual routing and forwarding" (VRF). In the control plane, Routing Information Base (RIB) is virtualized to multiple routing tables; while in the data plane, Forwarding Information Base (FIB) is virtualized as multiple forwarding tables. We could view it as a form of control/data plane virtualization but it is not considered a virtualized device.

In addition to separate routing tables, more advanced routers may also support (logically) separate routing protocols, configurations, etc. A "virtual routing instance" in such routers consists of a specific logical combination of a configuration, routing protocol and routing table. Various names have been given to such a virtual routing instance, such as "virtual router", "logical router" or "routing context".

c) Hardware-Partitioned Router: Routers that support hardware partitioning may host multiple routing instances in a single device; these are called "protected system domains" [13] by Juniper Networks, or "logical routers" [14] by Cisco Systems. Hardware-partitioned routers are mainly deployed in Points of Presence (PoP) of network carriers to save space and power and reduce management cost. Hardware is typically partitioned per line card, so that this technology can be viewed as a router per line card.

B. Link Virtualization

Link virtualization technologies create "virtual links", a term which, depending on the context, may carry various meanings. In this section, we review these different interpretations and unveil the various layers of link virtualization.

1) Physical Channel Multiplexing: When discussing link virtualization, the first issue that needs to be clarified is what exactly constitutes a "link". If the link is a physical medium, then link virtualization might be identical to multiplexing. A physical medium could be wired (e.g., fiber, copper cable) or wireless (e.g., wireless spectrum). Technologies such as time division multiplexing (TDM), frequency division multiplexing (FDM) and code division multiple access (CDMA) are widely used to multiplex distinct communication channels over a single physical medium. Although multiplexing is generally not considered as a link virtualization technology, we point out that on a fundamental level it performs a function very similar to virtualization: the physical medium is split into distinct channels, and the sender and receiver are under the illusion that they own the physical medium. Therefore, virtualization may be viewed as a generalization of multiplexing.

2) Bandwidth Virtualization: In this context, "link virtualization" refers to technologies that combine the bandwidth of individual channels together to form virtual links.

a) Circuit: In a traditional telephone call, a circuit is established for a call by concatenating a series of channels (time slots on a physical link) along the links of the path between two parties. The conversation is carried on these time slots, and the parties have the illusion that they use a dedicated line. In this case, this concatenation of time slots forms a virtual link.

b) Reverse Multiplexing: Advances in optical technologies make it possible to combine bandwidth so as to create flexible services (i.e., pools of bandwidth) independent of the capacity of the underlying physical links and devices. Such reverse multiplexing technologies may operate at the granularity of sub-wavelength or full wavelength rates.

For instance, SONET virtual concatenation binds several STS-n frames together to create channels of flexible bandwidth that cannot be supported by the traditional SONET bandwidth hierarchy; the combined STS-n frames may be contiguous or non-contiguous frames on the same SONET link, or may even belong to different SONET links. Similarly, four (respectively, ten) 10 G channels (wavelengths) may be combined to create one 40 G (respectively, 100 G) channel. Channels created via such reverse multiplexing technology are referred to as virtual links. Similar optical bandwidth virtualization techniques may be used to provision Layer 1 virtual private networks (L1 VPNs), which we describe in Section II.C.2.

Infinera provides bandwidth virtualization solutions [15] to enable a programmable optical network. In a traditional optical network, services are tightly coupled to particular physical devices. Bandwidth virtualization enables a decoupling in that services are not bound to particular devices but to virtualized bandwidth.

3) Data Path Virtualization: Data path virtualization refers to technologies that do not manipulate the channel itself, but rather the data (packets) carried on this channel. In this case, a virtual link corresponds to a data path with certain properties. Such a virtual link does not depend (directly) on the physical properties (e.g., bandwidth) of the links, rather it is provisioned by nodes. Specifically, nodes use various technologies to direct data along these virtual links (data paths); two popular technologies are discussed next.

a) Labels: Labels (might be also called tags, IDs, etc.) occupy certain fields in the packet header and serve as identification and sharing mechanisms. Nodes are aware of labels so that they may point traffic to the right direction; this is an important point in that the virtual link (data path) is enabled by nodes.

802.1q virtual LAN (VLAN) tags enable different VLANs to share a single physical medium while being logically separated. VLAN tags are more about sharing and can be used to distinguish data from different VLANs. At the same time, they are also employed to help form data paths for the broadcasting domain. We will discuss VLANs in greater detail in Section II.C.3.

The labels in asynchronous transfer mode (ATM), frame relay (FR), and multiprotocol label switching (MPLS) technologies are also used to specify the path that data packets take. These data paths are called virtual circuits. In Section II.B.2a, circuit denotes the fact that time slots are concatenated end-to-end. The term "virtual" here refers to links are concatenated, but not physical time slots.

b) Tunnels and Encapsulation: Tunnels (often using encapsulation techniques) provide virtual (logical) links to connect network devices that are not physically adjacent. For example, tunnels may be used to create the illusion for some protocols running on a network device that this device has a direct connection to another device even when no physical link between the two devices exists. Some popular technologies

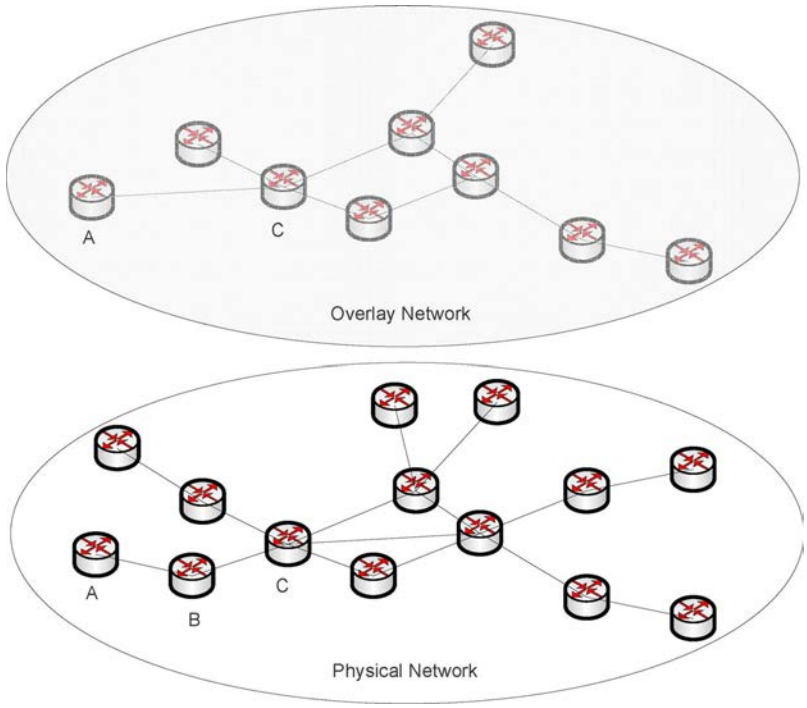

Fig. 3. Overlay Network.

TABLE I
LINK VIRTUALIZATION TECHNOLOGIES

Technology	Enabled by	Examples
Physical Channel Multiplexing	link physical properties	FDM, TDM
Bandwidth Virtualization	link physical properties, nodes	PSTN circuit, Reverse Multiplexing
Data Path Virtualization	nodes	VLANs, MPLS LSPs, GRE tunnels

include generic routing encapsulation (GRE) tunnels, Internet Protocol security (IPsec) tunnels, GPRS Tunnelling Protocol (GTP) tunnels, and MPLS label switched path (LSP) tunnels. Essentially, tunnels are overlay links and form the fundamental building blocks of overlay networks, which we discuss in Section II.C.1.

Table I summarizes the link virtualization technologies discussed in this section.

C. Virtual Networks

So far, we have encountered two types of virtual networks: "network-in-a-box" in Section II.A.1a and "bandwidth-virtualized network" in Section II.B.2b. In this section, we review several other network virtualization technologies, and we explain how they relate and compare to each other. In keeping with the theme of the overall section, we only consider commercial technologies offered by the industry; we will review virtual networks that are the subject of academic research in the next section.

1) Overlay Networks: An overlay network is one built upon an existing network, mainly using tunneling and encapsulation technologies. A major attraction of overlay networks is the ability to implement new network services economically by making use of existing network infrastructure. Consider, for instance, the physical network in Fig. 3, in which nodes A, B, and C are connected by two links, AB and BC. Suppose now that a new network service, such as a new routing protocol, between nodes A and C needs to be implemented. By using tunnels to connect nodes A and C only these two nodes need to be modified to implement the service. Node B on the other hand, needs no modification: it continues to forward data between A and C, but it is not aware of the new service. In the overlay network (at the top of Fig. 3), the tunnel gives nodes A and C the impression that they are connected to each other by link AC and have no knowledge of the existence of node B.

Many overlay networks have been constructed to create new services by augmenting the capabilities of existing infrastructure. Examples include Internet access networks (e.g., a digital subscriber line (DSL) or cable network overlaid upon older public switched telephone network (PSTN) or cable TV infrastructure), and the MBone [16] and 6Bone [17] that add multicast and IPv6 capabilities to the Internet, respectively. In overlay networks, it is the network topology that is virtualized, and all new-service-aware nodes form a virtual network. Overlay technology is also the key to implementing virtual private networks and virtual sharing networks, as described in the following two subsections, respectively.

2) Virtual Private Networks: A virtual private network (VPN), shown in Fig. 4, is an assembly of private networks that connect to each other but are isolated from public networks such as the Internet. For instance, organizations deploy VPNs to connect their offices in geographically distant locations, while individuals who work from their home typically use a VPN to access their company's internal network.

Layer 2 (L2) and Layer 3 (L3) VPN technologies are mature and widely deployed. In a L2 VPN, the VPN provider's network

Fig. 4. Virtual Private Network.

TABLE II
VPN SUMMARY

VPN Type	Provider's Network Virtualized As	Mainly Designed For
Layer 3	Router	IP/MPLS+BGP core
Layer 2	Switch	IP/MPLS+BGP core
Layer 1	Layer 1 Connection	TDM & optical network

is virtualized as a layer 2 switch, and customer sites are responsible for building their own routing infrastructure. On the other hand, in a L3 VPN, the provider's network is virtualized as a layer 3 router. Since 2005, Layer 1 (L1) VPN technology has been undergoing a rapid process of standardization. The fundamental difference between L2/L3 VPN and L1 VPN technology is the networks on which they operate. L2/L3 VPNs are typically built on an IP/MPLS+BGP core, while L1 VPN is mainly designed to run on TDM networks, such as SONET/SDH, or wavelength division multiplexing (WDM) optical networks. In L1 VPN, customers (i.e., the edge routers of private networks) will be able to request Layer 1 data paths across the provider's network, and these data paths are used to interconnect customer sites. However, an L1 VPN is provisioned by higher layer protocols, i.e., the requested data path is determined and established by protocols such as generalized MPLS (GMPLS). L1 VPN technology is still under development and rarely deployed in the field; the reader is referred to relevant RFCs [18]–[23] for more detailed information.

Table II summarizes these three types of VPN. In a VPN, the network topology is virtualized to provide the illusion that each customer's private networks are directly connected; the VPN also provides isolation among different customers.

3) Virtual Sharing Network: We use the term "virtual sharing network" (VSN) to denote technologies that support the sharing of physical resources among multiple network instances, while providing clear delineation between these instances. In the industry, such a network instance could be simply called a "virtual network" [24], further contributing to the overloading of the latter term. For clarity and to differentiate this concept from overlay networks and VPNs that we discussed above, we decided to coin the term VSN.

A virtual LAN (VLAN), discussed in Section II.B.3a can be thought of as a special case of a VSN. VLANs share the same physical LAN infrastructure while at the same time they are segmented within the boundary of broadcast domains. These two features, sharing and segmentation, are the two key properties of a VSN, a generalization of the VLAN concept to a broader network. Fig. 5 shows a typical deployment of VSN technology within a large- or medium-size corporate network. The guest network, employee network and administrator network may share the same access points (wireless hot spots, LAN

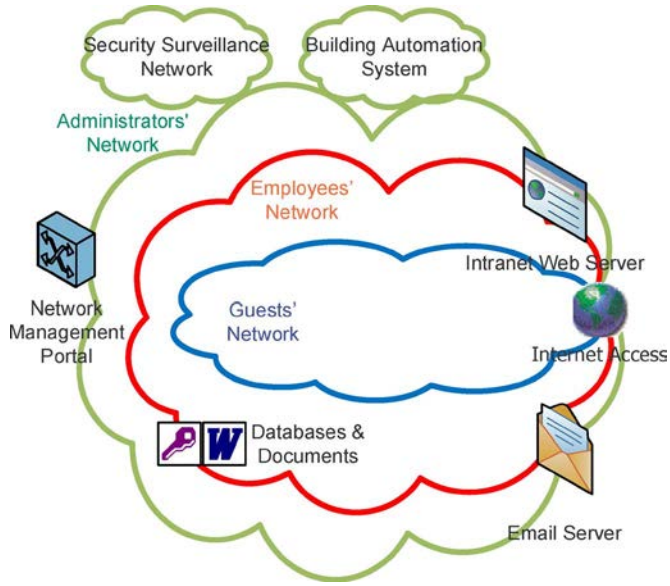

Fig. 5. Virtual Sharing Network.

Fig. 6. VPN-extended VSN.

access), physical switches and routers, and servers. However, these networks are also properly segmented as virtual networks with different access permissions. For instance, all virtual networks are able to access the Internet, but only the employee and administrator networks may access email servers, Intranet Web servers, and certain document and database servers; on the other hand, only the administrator network may provide the ability to configure physical devices or to access certain parts of the infrastructure, including the security surveillance network and building automation system. In this example, instead of building a separate physical network for guests, employees and administrators, provisioning a virtual network for each user group is a better solution in terms of cost, efficiency, maintenance effort, etc. The key requirement is to ensure that all virtual networks are able to share the same physical infrastructure while being properly segmented.

A VSN may be further extended by combining it with a VPN, as discussed above, to extend the user group-specific virtual networks of the VSN across multiple physical location connected by the VPN. In this case, the term *virtual* has two meanings: from the point of view of the VSN, each virtual (user group-specific) network shares the same physical resources with other VSN virtual networks; while from the point of view of the VPN, the VSN itself forms a virtual network since several peer VSN networks at geographically disperse locations are interconnected transparently. Fig. 6 is an illustration of a VPN-extended VSN.

Various technologies can be used to build VSNs. Fig. 7 shows the most common building blocks of a VSN, including access

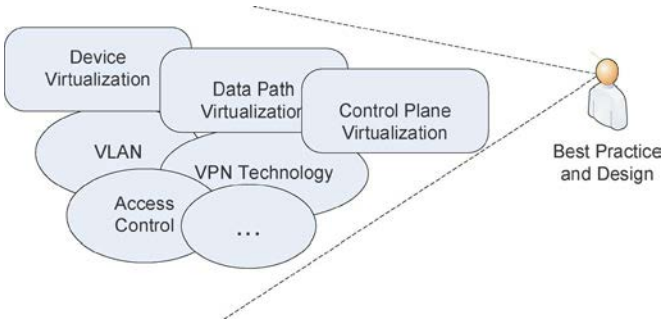

Fig. 7. Technology Building Blocks of VSN.

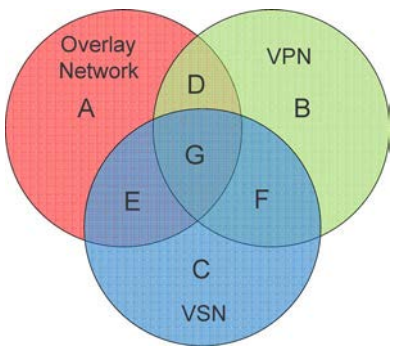

Fig. 8. Relations of Overlay Network, VPN and VSN (Examples in Table III).

control, VLAN, VPN, data-path virtualization, control-plane virtualization, etc. Depending on the design, a different set of technologies might be employed. In fact, designing a VSN is both science and art, and often requires substantial experience with both technologies and best practices.

4) Technology Comparison: A network can be regarded as "virtual" from different angles. An overlay network is considered virtual as it is separated from the underlay network; a VPN emphasis is virtual in the sense that it is separate from the public network; VSN focuses on the fact that multiple virtual networks share the same physical infrastructure. A type of virtual networks could become another type of virtual networks by simply changing the perspective. For example, considering the fact that overlay networks share the same underlay network, they are VSN to each other. As another example, part of a VPN is usually an overlay network upon an underlying provider network. Fig. 8 shows the relationship between the overlay network, VPN, and VSN concepts described in this section. Table III provides real-life technology examples that fit within the different areas (A to G) in Fig. 8. As Fig. 8 and Table III show, the three concepts overlap with each other in real deployments. We also note that a VPN over a provider's network (e.g., as in Fig. 4) is usually an overlay network, but not always. For instance, legacy peer-to-peer VPN technology [25], [26] does not use overlay networks.

Finally, Table IV compares the three technologies discussed in this section in terms of their emphasis and the specific entities that form the "virtual network" in each case.

TABLE III
EXAMPLES OF OVERLAY NETWORK, VPN AND VSN

Areas in Figure 8	Technology
A	Dial-up, DSL, Cable Internet
B	Peer-to-peer VPN [25], [26]
C	VLAN
D	Most modern VPNs, such as MPLS/BGP L3VPN
E	Overlay used in VSN
F	Peer-to-peer VPN used in VSN
G	VPN-Extended VSN (Figure 6)

TABLE IV
COMPARISON OF VIRTUAL NETWORK TECHNOLOGIES

Technology	Emphasis	Virtual Network
Overlay	New Service	Service-aware nodes
VPN	Connectivity	Attached private networks
VSN	Resource Sharing	Part of physical infrastructure

III. THE ACADEMIC COMMUNITY PERSPECTIVE

In this section, we focus on projects that are mainly driven by the academic community and explore the various aspects of network virtualization addressed by these projects. We categorize selected projects with an emphasis on identifying the entity or entities that are virtualized.

A. Testbeds Provisioned by Network Virtualization

The testbeds discussed in this section use network virtualization to facilitate the sharing of the same physical infrastructure among multiple experimenters (users of testbeds) while also ensuring that users of the infrastructure are isolated from each other. Our discussion focuses on what resources are virtualized and made available to the experimenters.

1) PlanetLab: PlanetLab is a research testbed, jointly established in 2002 by Princeton University, Intel, UC Berkeley, and the University of Washington. As Fig. 9(A) shows, PlanetLab is composed of PlanetNodes distributed around the world[1]. PlanetNodes are dedicated servers running PlanetOS (a customized Linux OS), and they are able to spawn VM slices upon request. PlanetLab does not deploy dedicated links, hence communication between PlanetNodes occurs through the Internet; in other words, PlanetLab forms an overlay network over the Internet. As a result, the performance of a particular experiment running on PlanetLab directly depends on the prevailing traffic conditions in the Internet at the time of the experiment.

Fig. 9(B) shows that with the help of the PlanetLab management system, users are able to select a subset of PlanetNodes and request a VM slice from each. As PlanetNodes are geographically distributed around the world, testbeds with a range of desired properties in terms of size, topology, and geographical coverage may be requested and constructed. Users can utilize the PlanetLab APIs to develop customized applications and distribute them to selected PlanetNodes so as to conduct experiments on a global scale. Therefore, PlanetLab provides a user-defined overlay network for experimentation with customized applications and services.

The PlanetLab design philosophy is described in [27], while the details of PlanetOS and the security aspects of PlanetLab are

[1]As of May 2012, there were 1 115 PlanetNodes available at 542 sites across the world, mainly in the United States, Europe, China and Japan.

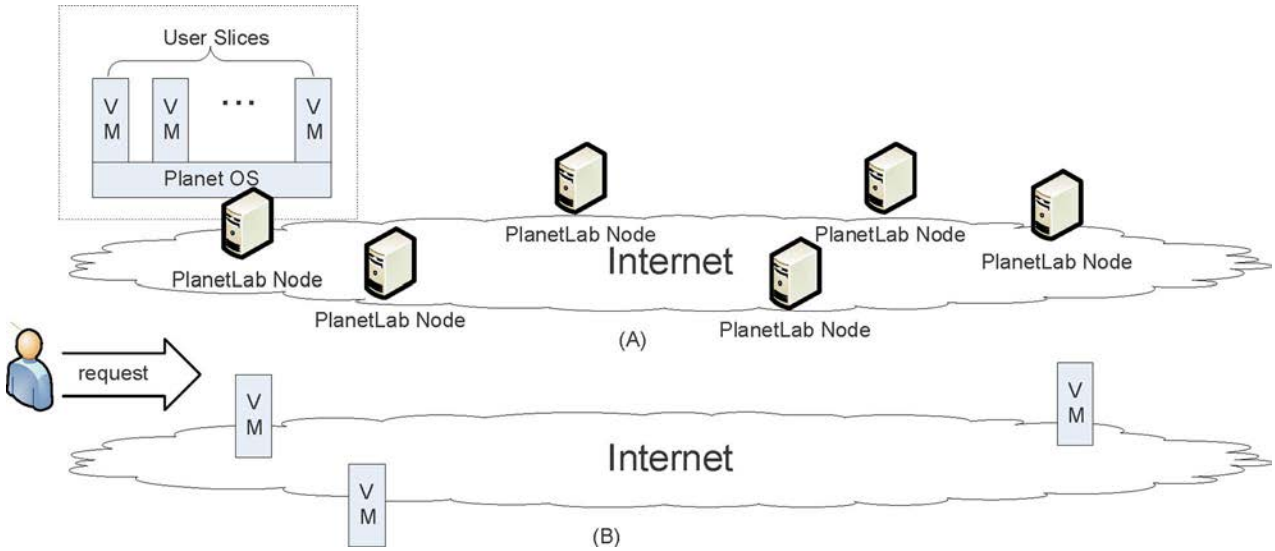

Fig. 9. PlanetLab.

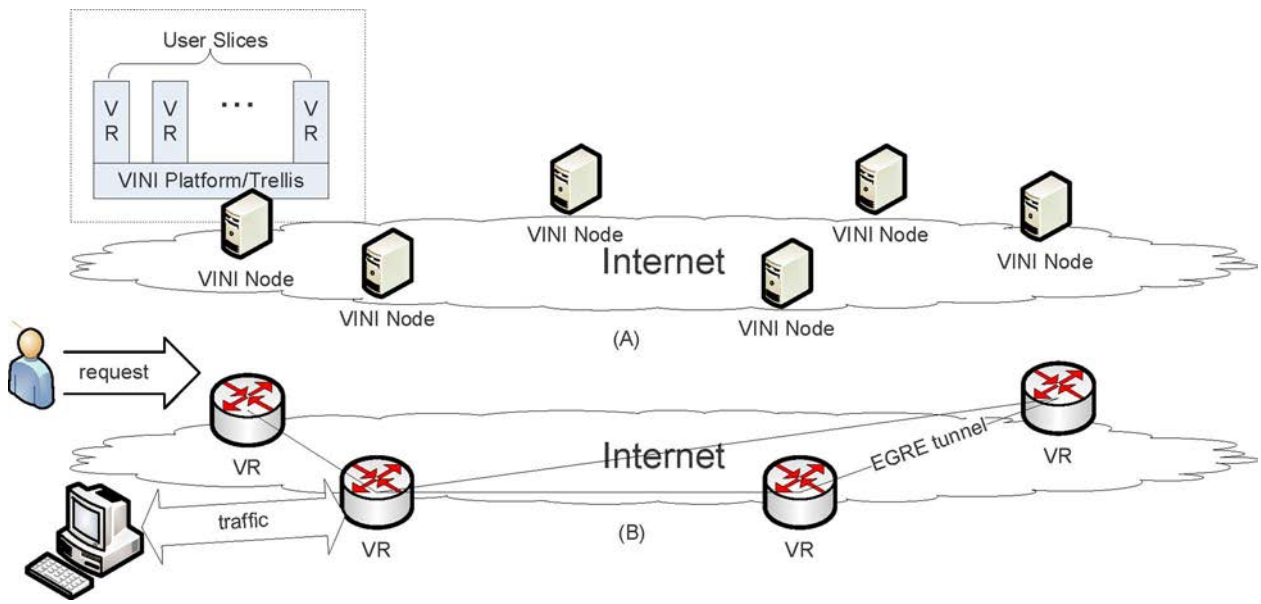

Fig. 10. VINI.

in [28] and [29], respectively; a discussion of the Linux virtual server used to provide virtual machine instances on PlanetNodes can be found in [30].

2) VINI & Trellis: VINI (which stands for "virtual network infrastructure") extends PlanetLab to provide a specific, enhanced, network protocol and service testbed. As Fig. 10 shows, VINI is also an overlay on the Internet, but it is different than PlanetLab in three important aspects. First, every slice (also called a virtual node) is enhanced as a router, not just as a general purpose slice; this is similar to the concept of routers in virtual OS we discussed in Section II.A.2a. Second, VINI provides tools to establish tunneling connections between virtual nodes. Finally, traffic can be directed between VINI and the outside world to simulate realistic conditions.

VINI was initially implemented upon PlanetLab as PL-VINI [31]. Later, it was upgraded to its own physical testbed[2] with a new software platform called Trellis [32]. Trellis uses GRE tunnels to connect virtual nodes on different physical machines. The tunnels are used by virtual nodes through an Ethernet-like virtual interface; hence these tunnels are called Ethernet generic routing encapsulation (EGRE) tunnels. Although there might be a dedicated uplink for the physical VINI nodes, bandwidth is not guaranteed as these tunnels use an IP path to set up virtual links. The tunnels are operated in the host OS while the virtual interface is operated in the virtual machine/container; the tunnels and virtual interfaces are connected using bridging (similar to *vSwitch* in Section II.A.1a). Trellis uses the Linux "bridge" module to connect one EGRE tunnel to multiple virtual interfaces and an optimization called "shortbridge" to connect one EGRE tunnel with one virtual interface. For more information on shortbridge, please refer to [32].

3) Emulab: Emulab [33], [34] is a research network testbed developed at the University of Utah. Since the original setup

[2]As of March 2011, there were 26 VINI nodes at 14 sites.

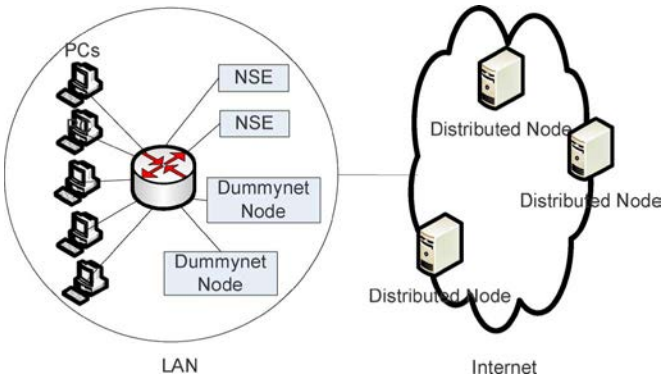

Fig. 11. Emulab.

in the late 1990s, Emulab has been an important platform for networking research and teaching.

As its name indicates, Emulab mainly emulates network properties based on user requests over a physical network. Emulab deployments (Fig. 11) consist of distributed nodes overlaid on the Internet, similar to PlanetLab and VINI; these nodes connect to one or more physical sites containing emulation hardware and/or software. Each physical site contains a set of Dummynet nodes [35], [36], ns-Emulates (NSE) [37] and PCs connected together through a LAN. Dummynet is an emulation tool used to test networking protocols while NSE generates software-simulated traffic; both interact with the PC-cluster emulated network and overlaid distributed nodes.

When users request a network link with specific delays, a Dummynet node is inserted within the LAN link assigned to emulate the requested link. A Dummynet node is simply a link emulator sitting between two PCs. User requests for specific types of traffic patterns are handled by the NSEs which direct the generated traffic to the network or host as requested by the user.

In recent years, Emulab and PlanetLab have been part of the global environment for network innovation (GENI) initiative [38], and significant development efforts have been undertaken to interconnect the two major testbeds.

4) VIOLIN: The virtual internetworking on overlay infrastructure (VIOLIN) [39], [40] project was an early effort to build an overlay network testbed that predates VINI. It was implemented on PlanetLab and is essentially similar to the PL-VINI implementation described above. VIOLIN software is implemented within a user-mode Linux (UML) container, which runs in a PlanetLab slice. A UML container is a virtual host, a virtual switch, or a virtual router. VIOLIN extends UML to enable UDP tunnels between the UML containers. As VIOLIN is independent of PlanetLab, it is easy to implant on other testbeds, such as Emulab.

5) G-Lab and OneLab: German Lab (G-Lab) is an initiative on the Future Internet sponsored by the German Federal Ministry of Education and Research. It uses PlanetLab-compliant technology and has deployed a testbed in Germany. Based on G-Lab, the topology management tool (ToMaTo) [41] provides a graphical user interface that allows users to design and use a virtual network by simply dragging and organizing network device and link icons. PlanetLab Europe, the European part of PlanetLab (separately administrated and federated with the rest of the PlanetLab), is operated under the OneLab project, funded by the European Commission. The main goals of OneLab include extending PlanetLab to wireless environments and federating with more testbeds, such as G-Lab.

B. Network Management

1) X-Bone: The X-Bone [42], [43] is a management tool that eases the effort of automatic configuration and management of overlay networks built upon an IP network, and includes enhanced security and monitoring features. It can be viewed as a tool to deploy overlay networks across the Internet, while also providing mechanisms for isolation and resource allocation among multiple overlay networks.

2) UCLPv2: User controlled lightpaths version 2 (UCLPv2) [44] is a set of software tools, in the form of Web services, which provide user-friendly, unified interfaces to various network elements (including optical switches, routers, etc.). The Web services simplify the process of combining or partitioning network resources such as bandwidth. In this context, the term "lightpaths" refers not only to wavelengths, but also to SONET circuits and MPLS LSPs.

UCLPv2 itself does not virtualize any network resource, but merely provides an abstraction of network elements. A network provisioned by UCLPv2 is called an "articulated private net" (APN). An APN is referred to as a next generation VPN [44], and bears similarities to the concept of a VPN that we discussed in Section II.C.2.

3) OpenFlow: OpenFlow [45] was proposed and prototyped by researchers mainly at Stanford University to enable flexible control of a switch's data plane. It unbundles the data plane and control planes and allows the controller to manage the data plane remotely through secure channels. OpenFlow identifies a common set of essential switching functions, and version 1.3 has been standardized by the open networking foundation in 2012. The separation of the controller makes it possible for researchers to experiment with new protocols by simply programming the controller remotely, which can be a simple PC. Since OpenFlow emphasizes open interfaces between switches and controllers, it is regarded as a key technology for the recently proposed software defined networking paradigm.

Although OpenFlow itself does not directly virtualize a network, it has been widely adopted as a means to enable network virtualization. For example, Flow Visor [46], [47] is an abstraction layer based on OpenFlow that enables the sharing of the same hardware data plane among multiple logical networks. OpenFlow Wireless (a.k.a, OpenRoads) [48], [49] is a framework based on OpenFlow and Flow Visor to manage WiFi and WiMAX wireless nodes. Other OpenFlow related network virtualization projects can be found at [50], [51].

C. Architectural Aspects

The virtual Internet architecture [52] is an overlay network architecture built upon a base net (e.g., the Internet). While still in a preliminary stage, this Virtual Internet (VI) is analogous to virtual memory in a computer. Just as virtual memory covers all the details of physical memory and appears to be owned by an

OS process, VI should also cover the details of the physical network and provide VI users with an abstraction of the network. VI is designed to provide maximum flexibility by supporting recursion (i.e., the ability to stack VIs to increase fault tolerance and path diversity) and revisitation (i.e., the capability for a base network node to emulate multiple VI nodes, thus extending the size of the network to be emulated). This architecture describes the host, router, link and topologies as network resources, and it examines the impact of using a virtualized environment over network resources.

IV. A Unifying Perspective

It should be evident by now that the term "network virtualization" is overloaded, it carries multiple connotations, and its meaning depends strongly on the particular context. In this section, we attempt to cut through this clutter and provide a unifying perspective of network virtualization.

A. Virtualization as Resource Abstraction

In computer science, "virtualization" is a general term used to refer to the abstraction of *fundamental* computing resources that gives users the illusion of sole ownership. For instance, virtual memory is an abstraction that provides processes with the impression that they control large amounts of physical memory. In OS platform virtualization, this abstraction mainly involves computer hardware components, including CPU, memory, hard drive, network card, etc., which are viewed as the fundamental resources.

We may extend this notion of abstraction to network virtualization by noting that the fundamental components of a network are nodes and links, hence these are the resources to be virtualized. Indeed, we discussed node and link virtualization in Sections II.A and II.B, respectively. In addition, users must have a mechanism to address/name nodes (e.g., IP address, MAC address), links (e.g., MPLS label, ATM circuit ID) and networks (VLAN ID, VPN ID) in order to make use of the network. Consequently, the set of addresses becomes an essential resource on its own: when the number of nodes, links and networks is larger than the address space, they compete for the limited address resources. Therefore, we consider addresses as a fundamental resource in the context of network virtualization. Addresses are widely used to aid node and link virtualization, but are not virtualized resources *per se*.

Recall also that in some instances "network virtualization" refers to providing a virtual topology that defines how nodes are connected to each other. One might argue that "topology" is a new type of fundamental resource, as links and nodes may be added or deleted to form a virtual topology. However, we note that a virtual topology cannot be constructed without some form of node or link virtualization. Hence, we view "topology" as a *derived*, not a fundamental resource; nevertheless, this is a subtle distinction that does not affect the fact that this resource may also be abstracted.

B. Existing Definitions

Let us now discuss existing definitions of "network virtualization" and point out how they fall short in encompassing the entire spectrum of virtualization we discussed in the previous two sections.

Consider first this definition, available in [53]:

> The term network virtualization refers to the creation of logical isolated network partitions overlaid on top of a common enterprize physical network infrastructure.

This statement defines what network virtualization creates, but does not explain what is virtualized, and why it is different from overlay networks discussed in Section II.C.1. Another statement can be found in [54]:

> The term "network virtualization" describes the ability to refer to network resources logically rather than having to refer to specific physical network devices, configurations, or collections of related machines. There are different levels of network virtualization, ranging from single-machine, network-device virtualization that enables multiple virtual machines to share a single physical-network resource, to enterprise-level concepts such as virtual private networks and enterprise-core and edge-routing techniques for creating subnetworks and segmenting existing networks.

This definition clearly points out that virtualization involves the abstraction of resources, and enumerates various such "levels" of abstraction. However, the concept of "level" in this definition is somewhat ambiguous and the provided list of levels does not comprise all aspects of virtualization we have encountered. Consider the following definition [55]:

> Network virtualization is an approach whereby several network instances can co-exist on a common physical network infrastructure. The type of network virtualization needed is not to be confused with current technologies such as Virtual Private Networks (VPNs), which merely provide traffic isolation: full administrative control as well as potentially full customization of the virtual networks (VNets) is also required to realize the vision of using network virtualization as the basis for a Future Internet.

This definition indicates that physical network infrastructure is virtualized as network instances, and it clearly differentiates network instances from VPNs. It does not provide a definition that unifies the various types of virtual networks we have discussed. Finally, consider this definition [56]:

> Network virtualization is the technology that enables the creation of logically isolated network partitions over shared physical network infrastructures so that multiple heterogeneous virtual networks can simultaneously coexist over the shared infrastructures. Also, network virtualization allows the aggregation of multiple resources and makes the aggregated resources appear as a single resource.

This definition emphasizes that the purpose of network virtualization is to create multiple heterogeneous virtual networks. It points out that network virtualization aggregates resources, but does not explain what resources are aggregated. Also, it does

TABLE V
SUMMARY OF NODE AND LINK VIRTUALIZATION

Category	Sub-category		Virtualized Resources
Node	NIC virtualization	Techniques in Xen and VMware	NIC is virtualized as vNIC with full interfaces to OS
		Techniques in Crossbow	vNIC with guaranteed bandwidth
		SR-IOV	HW is partitioned; no resource is virtualized
	host/endpoint		Same as OS virtualization
	Router virtualization	Routers in virtual OS	Same as OS virtualization
		Router control plane virtualization	Routing table, routing protocols and configurations
		Hardware-partitioned router	Power and chassis are shared; no resource is virtualized
Link	Bandwidth Virtualization		Bandwidth
	Tags and Labels		Data/flow identified by tags
	Tunnels		Encapsulated data/flow

TABLE VI
VIRTUAL NETWORK SOLUTIONS BY THE INDUSTRY

Type	Virtualized Resources
Network-in-a-box	The network is emulated by software; NICs are virtualized
Bandwidth-virtualized network	Bandwidth
Overlay	Links are virtualized by tunnels
VPN	Links are virtualized by tunnels
VSN	Links and nodes

not account for the slicing of resources imposed by many virtualization technologies.

C. A Unifying Definition

We offer this definition of network virtualization which we believe is more precise and encompasses a diverse spectrum of use:

> Network virtualization is any form of partitioning or combining a set of network resources, and presenting (abstracting) it to users such that each user, through its set of the partitioned or combined resources has a unique, separate view of the network. Resources can be fundamental (nodes, links) or derived (topologies), and can be virtualized recursively. Node and link virtualization involve resource partition/combination/abstraction; and topology virtualization involves new address (another fundamental resource we have identified) spaces.

Tables V–VII illustrate how the various network virtualization techniques we have discussed in this paper are covered by this definition by pointing out the set of resources that are virtualized in each case. Table V summarizes node and link virtualization, while Tables VI and VII considers select industry solutions and academic efforts, respectively, from the point of view of the virtualized resources.

D. Virtualization versus Simulation and Emulation

Occasionally, "network virtualization" is used as a term synonymous to network simulation or emulation [57]. We do not encourage this usage, considering that (1) network simulation and network emulation are better terms to use, depending on the intended meaning (discussed shortly), and (2) using the term "network virtualization" in a context unrelated to resource abstraction would be highly confusing.

TABLE VII
SUMMARY OF SELECTED ACADEMIC PROJECTS

Project	Virtualized Resources
PlanetLab, VINI Emulab	Links are virtualized by tunnels; Nodes are virtualized by OS virtualization.
X-Bone	Same as overlay network in Table VI
UCLPv2	Bandwidth
Virtual Internet Architecture	Hosts, routers, and links

Network simulation replicates network behavior by software and/or hardware without the actual network being present. OPNET [58] and NS2/NS3 [59] are two widely used network simulation tools. Such tools use discrete event simulation on network topologies that do not involve any real (physical) network. As we discussed in Section III.A.3, in Emulab, NSE relays the simulated world to the real world, injecting simulated traffic into the real network and processing the traffic from the real network. In this case, the simulated network comprises part of the virtual network provided by Emulab. Note also that the "network-in-a-box" scenario we described in Section II.A.1a is a fundamentally different concept as it exists as part of a real network.

Network emulation techniques replicate some properties of a network device or traffic over a physical network. Such tools may use software and/or hardware modules deployed on a physical network being tested so as to improve the performance and scalability of the testing process. Properties of links (e.g., bandwidth, latency, etc.), and/or traffic patterns (e.g., packet lost, packet duplications, jitter, etc.) are commonly emulated by such tools. Dummynet, Netem [60], and NistNET [61] are examples of popular network emulation tools. Because an emulated network is not a real physical network, it has been argued that it is a virtual network; as we mentioned above, we do not agree with this definition of virtualization as it does not involve resource abstraction.

V. FRONTIERS

We now discuss a number of emerging technologies and research directions in network virtualization that appear to us to be both important and challenging. Our purpose is not to offer an all-inclusive list of topics, but to generate interest in an area of practical importance.

A. Network Leasing

With more mature network virtualization technologies, infrastructure owners are able to lease part of their networks to others by means of virtualization. Although VPNs (discussed in Section II.C.2) have long served this purpose, a new approach in [62] has the potential for a broader impact by decomposing the role of service providers.

Traditional Internet service providers, such as AT&T and Time Warner, own and operate a physical infrastructure and provide services, such as Internet access, corporate VPN,

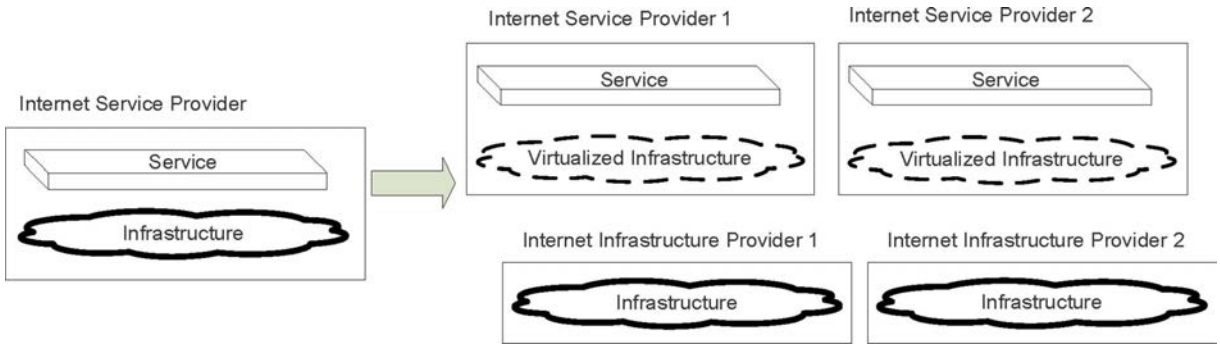

Fig. 12. Decomposition of an Internet Service Provider.

etc. Network virtualization allows the owners of a physical infrastructure to act as infrastructure providers by provisioning multiple virtualized infrastructures. Of course, an infrastructure provider may also be a service provider at the same time and can provide services upon its own infrastructure. However, it may also be able to lease unused resources in the form of virtualized infrastructure to other service providers. A service provider can lease resources from a single or multiple infrastructure providers. Fig. 12 illustrates this decomposition process.

B. 4WARD VNet Model

The 4WARD research project [63] is part of the European Union (EU) 7th Framework Programme. One of the 4WARD work packages (WP3-VNet) focuses on network virtualization and its impact on the Future Internet. Its objective is to provide virtualization of network resources, virtual networks and virtualization management. Network resources to be virtualized include servers and links. 4WARD focuses on a generalized approach to allow virtualization of different resources that form a unified framework, and supports both wireline and wireless resources.

In the 4WARD WP3 technical report [55], the above network leasing concept is further extended, and three roles are proposed. The infrastructure provider (InP) owns and maintains the physical network; it also provides means to virtualize its physical resources. The virtual network provider (VNP) uses virtual resources provided by one or more InPs, provisions virtual networks, and makes them available to virtual network operators (VNOs). In turn, VNOs operate virtual networks and provide services.

This three-role model is further expanded in [64] which introduces end-customers and application service providers and enables more flexible provisioning of virtual networks and services. Three use cases (beta slice, service broker in access networks, and service component mobility in mobile networks) are also discussed in detail.

In [65], the impact and challenges of network virtualization, especially for InPs, are discussed. Manageability, scalability and reliability are three aspects that should be properly solved to pave the way for deployment of services based on network virtualization.

The scalable & adaptive Internet solutions (SAIL) project is another Framework Programme 7 project. Cloud networking (CloNet) [66], a key focus of SAIL, aims to expand the concept of network virtualization to accommodate Cloud providers as VNOs.

C. Testbeds for Next Generation Internet

Many in the networking research community have argued that the current Internet is ossified [67] and needs to be improved. Network virtualization offers a natural solution to overcoming the Internet impasse [68]–[70], namely, the deployment of testbeds as virtualized infrastructure on which to experiment with newly designed architectures. The GENI and 4WARD projects in the United States and the EU, respectively, are currently two major Next Generation Internet initiatives. Both have adopted network virtualization as a core strategy.

GENI is a collection of research projects in the United States, most of which are clustered around four control frameworks: PlanetLab, ProtoGENI (a descendent of Emulab), ORCA (open resource control architecture) [71] and ORBIT [72]. Broadly speaking, GENI does not aim to be the Next Generation Internet; it hopes to provide researchers ways to explore all the possibilities and addresses most of the challenges in the network virtualization area. GENI aims to design a streamlined process to instantiate virtual networks, to allow on-demand addition of resources to these networks, and to manage the virtual networks.

G-Lab and OneLab, which we discussed in Section III.A.5, provide testbeds enabled by virtualization to researchers to explore the Future Internet. Another recent effort, Future Internet core platform (FI-WARE) [73], also a EU 7th Framework Programme project, proposes to leverage network virtualization in its testbed architecture design.

D. Wireless Network Virtualization

While most network virtualization technologies and concepts have their origins in wired networks, researchers are also exploring the implication of virtulization to wireless networks, including mobile cellular networks and sensor networks. In this context, wireless spectrum represents a resource to be virtualized, consistent with the link virtualization concept we discussed in Section II.B. Various wireless virtualization techniques are discussed in [74], along with new challenges introduced due to mobility and unique properties of wireless nodes.

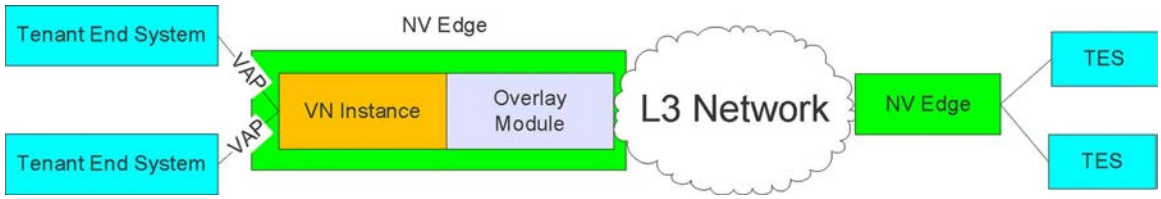

Fig. 13. NVO3 Reference Module.

1) Mobile Cellular Network Virtualization: The mobile virtual network operator (MVNO) concept is well known and understood in the mobile industry. Notably, most Virgin Mobile brands around the world have been established as MVNOs. An MVNO gets access to mobile network owned by other providers through contracts and agreements, and runs its owb billing and customer servicess. Network virtualization technologies and the 4WARD VNet Model in Section V.B will provide seamless support for MVNOs. Several use cases are discussed in [75].

In [76], the network virtualization substrate (NVS) concept is proposed to virtualize a WiMAX network. A WiMAX base station uses orthogonal frequency division multiple access (OFDMA) frame structure for transmission between the base station and its subscribers. NVS virtualizes the wireless spectrum into distinct slices by using a slice ID to group a slice's download link subFrame and upload link subframe, and to provide flow-level isolation. OS platform virtualization technologies are also employed to virtualize other components in a WiMAX network including the access services network and the connectivity services network. Similarly, it is proposed in [77] to schedule a physical resource block, the smallest unit that the long term evolution (LTE) MAC scheduler handles, among different users to achieve spectrum virtualization. LTE enhanced Node B (eNodeB) may be also virtualized as virtual eNodeB using OS flatform virtualization technologies.

2) Sensor Network Virtualization: The impact of network virtualization on sensor networks may not be as profound as on cellular networks. Most sensor nodes have limited resources, are dedicated to a specific application, and are sensitive for latency. As long as these limitations persist, virtualization solutions may remain a challenge.

Current research activities for sensor network virtualization mostly focuses on virtualizing the sensor software platform to enable multiple applications sharing the same sensor [78], [79]. A virtual sensor network, as proposed in [80], consists of a dynamic subset of sensors that can form a virtual network and work together on a particular application. Multiple virtual sensor networks may be formed on the same physical sensor network. Readers are referred to [81] for a detailed survey on virtualization of wireless sensor networks.

E. Standardization Effort on Virtual Networks in Data Center

With the rapid adaptation and development of virtual networks in data centers, IETF has chartered the Network Virtualization Overlays (NVO3) working group to standardize the NVO3 framework to solve new challenges.

A virtual network in a data center is defined as "a virtual L2 or L3 domain that belongs to a tenant [82]". One distinct characteristic of this virtual network is the Tenant End System (TES) attached to it. These TESs could be a non-virtualized server, a physical appliance or most likely a VM instance in today's data center. The NVO3 framework uses the overlay network technologies we have discussed in Section II.C.1, and its aim is to (1) dynamically provision a tenant virtual network; (2) provide isolation between tenant virtual networks; (3) enable connectivity between tenant virtual networks and other networks, such as a customer site network; and (4) gracefully handle VM mobility in a tenant virtual network.

Fig. 13 gives an overview of the NVO3 reference module. TESs could be attached to a Virtual Network Instance (VNI) through a Virtual Access Point (VAP), and the VNI could connect to other VNIs or other part of customer network through an overlay module. The overlay module tunnels traffic and provides L2 or L3 connectivity between VNIs upon the underlay L3 network. The collection of VAP, VNI and the overlay module is called Network Virtualization Edge (NVE). For more details of NVO3, readers are referred to the NVO3 problem statement [83], framework [82] and use cases [84].

There are several approaches that can be used to implement the overlay module in the NVO3 framework. Virtual eXtensible Local Area Network (VXLAN) [85] is proposed in the wake of the insufficient VLAN ranges in data center multi-tenant environments. A new VXLAN header is added to a L2 packet that is carried in L3 UDP packets. Instead of introducing new headers, Network Virtualization using Generic Routing Encapsulation (NVGRE) [86] leverages the current GRE header to carry a 24 bit Virtual Subnet Identifier, and a L2 packet is encapsulated as a GRE packet upon a L3 IP packet. Stateless Transport Tunnelling Protocol (STT) [87] is another flavor of encapsulation. It encapsulateis data into a TCP-header-like SST header so that the NIC, which supports TCP segmentation offload, could segment the data for better performance. STT also allows more flexibility of defining virtual network meta-data.

F. Research Challenges

1) Control Frameworks: Control frameworks are key to enable network virtualization. The four control frameworks under GENI collaborate and compete with each other. We envision that multiple control frameworks will succeed and federate (work together) with each other to compose a future global control framework.

One aspect of such frameworks is resource control, which includes polling resources, leasing resources to requesters, interacting with other resource control frameworks and acting as a resource broker. This is crucial to network leasing, as discussed in Section V.A. Another aspect has to do with enabling resource initialization, deployment, monitoring, etc.

If a control framework is not open to the outside world, it makes the controlling aspects easy while sacrificing the ability to federate with other frameworks. On the other hand, if a framework is too open, it is flexible yet runs a greater risk of being abused. Each of the existing control frameworks employs a different philosophy in balancing these tradeoffs.

2) Security: In a virtualized network, security issues can be broadly classified into three categories. First, some issues are related directly to similar considerations in virtualized operating systems. For instance, in operating systems, the hypervisor should be well secured because it controls, and can potentially affect, all the VMs. Similarly, in a virtual network environment, the substrate that controls the different virtual networks should be protected. Multiple users are accessing the same physical network resources, such as routers. Isolation between the different virtual networks is essential to maintain the illusion of separation: if a user in one virtual network is somehow able to detect the presence of other virtual networks, then the illusion of separation is broken. On the other hand, isolation can only provide limited security and privacy with the use of current encryption techniques. For instance, since there may be a single physical router, isolation does not eliminate the risks involved when the router itself is attached.

Another class of security issues include well-known goals such as confidentiality, integrity and availability. Several solutions, such as authentication and intrusion detection, have been designed to address such goals and to prevent attacks related to privacy, non-repudiation and "man-in-the-middle."

The third type of security vulnerabilities is specific to virtual networks. These arise when networks are programmable. In the absence of well-structured policies and rules, programmability may significantly increase the vulnerability of the network. So far, security issues that are specific to virtual networks are relatively unaddressed in the field. Specifically, the community has neither shown that virtual networks are as secure as traditional networks, nor provided enough security measures to defend them. Hence, we expect this field to become increasingly important as network virtualization technologies proliferate.

VI. Summary

In this paper, we have conducted a thorough review of network virtualization efforts by both the industry and academic research groups. By focusing on the main feature of virtualization, i.e., resource abstraction, and identifying relevant network resources our aim has been to provide a unifying perspective that brings to light the commonalities among a seemingly diverse set of approaches. We also discussed a set of research directions and challenges that will arise as network virtualization becomes the enabler for future Internet architecture research, deployment, and experimentation.

Acknowledgment

The authors would like to thank T. Biswas for her comments on an early version of this paper and for in-depth discussion of some topics. We also thank the anonymous reviewers for their constructive comments that helped us improve the article.

References

[1] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, pp. 862–876, Apr. 2010.
[2] M. Chowdhury and R. Boutaba, "Network virtualization: State of the art and research challenges," *IEEE Commun. Mag.*, vol. 47, pp. 20–26, Jul. 2009.
[3] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *SOSP'03: Proc. the Nineteenth ACM Symp. Operating Systems Principles*, New York, NY, USA, 2003, pp. 164–177, ACM.
[4] VMware, Virtualization Overview [Online]. Available: http://www.vmware.com/pdf/virtualization.pdf
[5] R. McDougall and J. Mauro, "Zones," in *Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, Jul. 20, 2006, ch. 6.
[6] VMWare, VMware Virtual Networking Concepts Jul. 2007 [Online]. Available: http://www.vmware.com/files/pdf/virtual_networking_concepts.pdf
[7] S. Tripathi, N. Droux, T. Srinivasan, and K. Belgaied, "Crossbow: From hardware virtualized NICs to virtualized networks," in *Proc. VISA'09*, New York, 2009, pp. 53–62.
[8] 802.1qbg—Edge Virtual Bridging [Online]. Available: http://www.ieee802.org/1/pages/802.1bg.html
[9] 802.1qbh—Bridge Port Extension [Online]. Available: http://www.ieee802.org/1/pages/802.1bh.html
[10] Cisco, Cisco Nexus 1000 v Series Switches [Online]. Available: http://www.cisco.com/en/US/products/ps9902/index.html
[11] PCI-SIG, Single Root I/O Virtualization and Sharing 1.1 Specification [Online]. Available: http://www.pcisig.com/members/downloads/specifications/iov/sr-iov1_1_20%Jan10_cb.pdf
[12] Y. Dong, X. Yang, X. Li, J. Li, K. Tian, and H. Guan, "High performance network virtualization with SR-IOV," in *Proc. High Performance Computer Architecture (HPCA)*, 2010, pp. 1–10.
[13] J. Sonderegger, O. Blomberg, K. Milne, and S. Palislamovic, "Virtualization for high availability," in *Junos High Availability: Best Practices for High Network Uptime (Animal Guide)*, 1st ed. New York: O'Reilly Media, Aug. 2009, ch. 5, p. 119.
[14] M. Victor and R. Kumar, "A virtualization technologies primer: Theory," in *Network Virtualization*, 1st ed. New York: Cisco Press, Jul. 2006, ch. 4.
[15] Infinera, Bandwidth Virtualization Enables a Programmable Optical Network [Online]. Available: http://www.infinera.com/pdfs/whitepapers/White_Paper_Bandwidth_Virtuali%zation.pdf 2007
[16] K. Savetz, N. Randall, and Y. Lepage, *MBONE: Multicasting Tomorrow's Internet*. New York: Wiley, 1996, (Computers).
[17] R. Rockell and R. Fink, "Bone Backbone Routing Guidelines," RFC 2772 (Informational), Feb. 2000, Updated by RFC 3152.
[18] T. Takeda, "Framework and Requirements for Layer 1 Virtual Private Networks," RFC 4847 (Informational), Apr. 2007.
[19] H. Ould-Brahim, D. Fedyk, and Y. Rekhter, "BGP-Based Auto-Discovery for Layer-1 VPNs," RFC 5195 (Proposed Standard), Jun. 2008.
[20] D. Fedyk, Y. Rekhter, D. Papadimitriou, R. Rabbat, and L. Berger, "Layer 1 VPN Basic Mode," RFC 5251 (Proposed Standard), Jul. 2008.
[21] I. Bryskin and L. Berger, "OSPF-based Layer 1 VPN Auto-Discovery," RFC 5252 (Proposed Standard), Jul. 2008.
[22] T. Takeda, "Applicability Statement for Layer 1 Virtual Private Network (L1VPN) Basic Mode," RFC 5253 (Informational), Jul. 2008.
[23] L. Berger, "OSPFv3-Based Layer 1 VPN Auto-Discovery," RFC 5523 (Experimental), Apr. 2009.
[24] M. Victor and R. Kumar, "Transport virtualization—VNs," in *Network Virtualization*, 1st ed. : Cisco Press, Jul. 2006, ch. 3, p. 41.
[25] P. Ferguson and G. Huston, "What is a VPN?—Part I," *Internet Protocol J.*, vol. 1, no. 1, 1998.
[26] T. Nam-Kee, "VPN-in-brief," in *Building VPNs: With IPSec and MPLS*. New York: McGraw-Hill Professional, Jul. 2003, ch. 1, pp. 9–11.
[27] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A blueprint for introducing disruptive technology into the Internet," in *Proc. SIGCOMM Comput. Commun. Rev.*, 2003, vol. 33, no. 1, pp. 59–64.
[28] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak, "Operating system support for planetary-scale network services," in *Proc. NSDI*, Berkeley, CA, 2004, p. 19.
[29] P. Brett, M. Bowman, J. Sedayao, R. Adams, R. Knauerhause, and A. Klingaman, "Securing the PlanetLab distributed testbed: How to manage security in an environment with no firewalls, with all users having root, and no direct physical control of any system," in *Proc. LISA*, Berkeley, CA, 2004, pp. 195–202.

[30] S. Soltesz, H. Pötzl, M. E. Fiuczynski, A. Bavier, and L. Peterson, "Container-based operating system virtualization: A scalable, high-performance alternative to hypervisors," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 3, pp. 275–287, 2007.

[31] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford, "In VINI veritas: Realistic and controlled network experimentation," in *Proc. ACM SIGCOMM Computer Commun. Rev.*, Pisa, Italy, Oct. 2006.

[32] S. Bhatia, M. Motiwala, W. Muhlbauer, Y. Mundada, V. Valancius, A. Bavier, N. Feamster, L. Peterson, and J. Rexford, "Trellis: A platform for building flexible, fast virtual networks on commodity hardware," in *Proc. CONEXT*, New York, 2008, pp. 1–6.

[33] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," in *Proc. 5th Symp. Operating Systems Design Implement.*, Boston, MA, Dec. 2002, pp. 255–270.

[34] M. Hibler, R. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, K. Webb, and J. Lepreau, "Large-scale virtualization in the Emulab network testbed," in *Proc. ATC'08: USENIX 2008 Annual Technical Conf. Annual Technical*, Berkeley, CA, 2008, pp. 113–128.

[35] L. Rizzo, "Dummynet: A simple approach to the evaluation of network protocols," in *Proc. SIGCOMM Comput. Commun. Rev.*, 1997, vol. 27, no. 1, pp. 31–41.

[36] L. Rizzo, "Dummynet and forward error correction," in *Proc. ATEC'98: Proc. Annual Conf. USENIX Annual Technical Conf.*, Berkeley, CA, USA, 1998, pp. 31–31.

[37] K. Fall, "Network emulation in the Vint/NS simulator," in *Proc. ISCC'99 4th IEEE Symp. Computers Communications*, Washington, DC, 1999, p. 244.

[38] L. Peterson *et al.*, "GENI design principles," *Computer*, vol. 39, pp. 102–105, Sep. 2006.

[39] X. Jiang and D. Xu, "VIOLIN: Virtual Internetworking on Overlay Infrastructure," Purdue University, , Tech. Rep., 2003.

[40] P. Ruth, X. Jiang, D. Xu, and S. Goasguen, "Virtual distributed environments in a shared infrastructure," *Computer*, vol. 38, pp. 63–69, May 2005.

[41] D. Schwerdel, D. Hock, D. Günther, B. Reuther, P. Müller, and P. Tran-Gia, "ToMaTo—A network experimentation tool," in *Proc. 7th Int. ICST Conf. Testbeds Research Infrastructures for the Development of Networks and Communities (TridentCom 2011)*, Shanghai, China, Apr. 2011.

[42] J. Touch and S. Hotz, "The X-Bone," in *Proc. 3rd Global Internet Mini-Conf. Globecom*, Nov. 1998, pp. 59–68.

[43] J. Touch, "Dynamic Internet overlay deployment and management using the X-Bone," in *Proc. Int. Conf. Network Protocols*, 2000, pp. 59–68.

[44] E. Grasa, G. Junyent, S. Figuerola, A. Lopez, and M. Savoie, "UCLPv2: A network virtualization framework built on Web services [Web services in telecommunications, Part II]," *IEEE Commun. Mag.*, vol. 46, pp. 126–134, Mar. 2008.

[45] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," in *Proc. SIGCOMM Computer Commun. Rev.*, Mar. 2008, vol. 38, pp. 69–74.

[46] R. Sherwood, G. Gibby, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "FlowVisor: A Network Virtualization Layer," Deutsche Telekom Inc. R&D Lab., Stanford University, Nicira Networks, Tech. Rep. OPENFLOW-TR-2009-1, 2009.

[47] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T.-Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K.-K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar, "Carving research slices out of your production networks with OpenFlow," in *Proc. ACM SIGCOMM Computer Communication Review*, Jan. 2010, vol. 40, pp. 129–130.

[48] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, "OpenRoads: Empowering research in mobile networks," in *Proc. SIGCOMM Comput. Commun. Rev.*, 2010, vol. 40, no. 1, pp. 125–126.

[49] K.-K. Yap, R. Sherwood, M. Kobayashi, T.-Y. Huang, M. Chan, N. Handigol, N. McKeown, and G. Parulkar, "Blueprint for introducing innovation into wireless mobile networks," in *Proc. 2nd ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures, VISA'10*, New York, NY, USA, 2010, pp. 25–32, ACM.

[50] G. Kontesidou and K. Zarifis, "OpenFlow Virtual Networking: A Flow-Based Network Virtualization Architecture," Master's Thesis, KTH Royal Institute of Technology, , 2009, TRITA-ICT-EX-2009:205.

[51] NEC, NEC ProgrammableFlow: Redefining Cloud Network Virtualization With OpenFlow. [Online]. Available: http://www.nec.com/en/global/prod/pflow/images_documents/NEC_Programmab%leFlow_Redefining_Cloud_Network_Virtualization_with_OpenFlow.pdf

[52] J. D. Touch, Y.-S. Wang, L. Eggert, and G. G. Finn, "A Virtual Internet Architecture," USC/Information Sciences Institute, , Tech. Rep. ISI Technical Report ISI-TR-2003-570, Mar. 2003.

[53] Cisco, Network Virtualization-Path Isolation Design Guide. [Online]. Available: http://www.cisco.com/en/US/docs/solutions/Enterprise/Network_Virtualiza%tion/PathIsol.html

[54] W. von Hagen, "Professional Xen Virtualization," 2008, Wrox.

[55] S. Baucke *et al.*, "Virtualization Approach: Concept," , Tech. Rep. FP7-ICT-2007-1-216041-4WARD/D-3.1.1, The 4WARD Project, 2009.

[56] S. Jeong and H. Otsuki, "Framework of Network Virtualization," 2009, FG-FN OD-17.

[57] R. Mangharam, "Mixed Reality, Now a Reality: Network Virtualization for Real-Time Automotive-CPS Networks," Department of Electrical & Systems Engineering, Univ. Pennsylvania, Philadelphia, Tech. Rep., Mar. 2008.

[58] X. Chang, "Network simulations with OPNET," in *Proc. 31st Conf. Winter Simulation: Simulation—A Bridge to the Future—Volume 1, WSC'99*, New York, 1999, pp. 307–314.

[59] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 project goals," in *Proc. 2006 Workshop on NS-2: The IP Network Simulator, WNS2'06*, New York, 2006.

[60] S. Hemminger, "Network emulation with NetEm," in *Proc. 6th Australia's National Linux Conference (LCA'05)*, 2005.

[61] M. Carson and D. Santay, "NIST net: A linux-based network emulation tool," in *Proc. SIGCOMM Comput. Commun. Rev.*, 2003, vol. 33, no. 3, pp. 111–126.

[62] N. Feamster, L. Gao, and J. Rexford, "How to lease the Internet in your spare time," in *Proc. SIGCOMM Comput. Commun. Rev.*, 2007, vol. 37, no. 1, pp. 61–64.

[63] N. Niebert, S. Baucke, I. E. Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. M. Correia, "The way 4WARD to the creation of a future internet," in *Proc. PIMRC*, 2008, pp. 1–5.

[64] D. Schlosser, M. Jarschel, M. Duelli, T. Hoßfeld, K. Hoffmann, M. Hoffmann, H. J. Morper, D. Jurca, and A. Khan, "A use case driven approach to network virtualization," in *IEEE Kaleidoscope 2010*, Würzburg, Dec. 2010, accepted at, published via OPUS Würzburg under OpenAccess.

[65] J. Carapinha and J. Jiménez, "Network virtualization—A view from the bottom," in *VISA'09: Proc. 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, New York, 2009, pp. 73–80, ACM.

[66] P. A. Aranda-Gutirrez and J. Carapinha, "Cloud networking: Implications of agile virtualization on provider relationships," in *Proc. ECEASST*, 2011, vol. 37.

[67] R. Dutta, G. N. Rouskas, I. Baldine, A. Bragg, and D. Stevenson, "The SILO architecture for services integration, control, and optimization for the future internet," in *Proc. IEEE ICC 2007*, Glasgow, U.K., Jun. 2007, pp. 1899–1944.

[68] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[69] N. Niebert, I. E. Khayat, S. Baucke, R. Keller, R. Rembarz, and J. Sachs, "Network virtualization: A viable path towards the future Internet," *Wireless Pers. Commun.*, vol. 45, pp. 511–520, Mar. 2008.

[70] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network virtualization: Implementation steps towards the future internet," in *Proc. KiVS 2009*, Kassel, Mar. 2009.

[71] J. Chase, L. Grit, D. Irwin, V. Marupadi, P. Shivam, and A. Yumerefendi, "Beyond virtual data centers: Toward an open resource control architecture," in *Proc. Sel. Papers Int. Conf. Virtual Computing Initiative (ACM Digital Library)*, May 2007, ACM.

[72] M. Ott, I. Seskar, R. Siraccusa, and M. Singh, "ORBIT testbed software architecture: Supporting experiments as a service," in *Proc. 1st Int. Conf. Testbeds Research Infrastructures for the Development Netw. Communities, TRIDENTCOM'05*, Washington, DC, 2005, pp. 136–145.

[73] J. Hierro, Overall FI-WARE Vision. [Online]. Available: http://forge.fiware.eu/plugins/mediawiki/wiki/fiware/index.php/Overall%_FI-WARE_Vision

[74] S. Paul and S. Seshan, "Technical Document on Wireless Virtualization," GENI: Global Environment for Network Innovations, , Tech. Rep., Sep. 2006, GDD-06-17.

[75] M. Hoffmann and M. Staufer, "Network virtualization for future mobile networks: General architecture and applications," in *Proc. 2011 IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2011, pp. 1–5, ACM.
[76] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "NVS: A virtualization substrate for WiMAX networks," in *Proc. 16th Annual Int. Conf. Mobile Computing Netw. MobiCom'10*, New York, 2010, pp. 233–244, ACM.
[77] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE mobile network virtualization," *Mobile Networks and Applications*, vol. 16, pp. 424–432, Aug. 2011.
[78] H. B. Lim, M. Iqbal, and T. J. Ng, "A virtualization framework for heterogeneous sensor network platforms," in *Proc. 7th ACM Conf. Embedded Networked Sensor Systems, SenSys'09*, New York, 2009, pp. 319–320.
[79] D. E. Irwin, N. Sharma, P. J. Shenoy, and M. Zink, "Towards a virtualized sensing environment," in *TRIDENTCOM*, T. Magedanz, A. Gavras, H.-T. Nguyen, and J. S. Chase, Eds. New York: Springer, 2010, vol. 46, pp. 133–142.
[80] A. P. Jayasumana, Q. Han, and T. H. Illangasekare, "Virtual sensor networks—A resource efficient approach for concurrent applications," in *Proc. Int. Conf. Inf. Technol. ITNG'07*, Washington, DC, 2007, pp. 111–115.
[81] M. M. Islam, M. M. Hassan, G.-W. Lee, and E.-N. Huh, "A survey on virtualization of wireless sensor networks," *Sensors*, vol. 12, pp. 2175–2207, 2012.
[82] F. Balus, T. Morin, N. Bitar, and Y. Rekhter, "Framework for DC network virtualization," Jul. 2012, Internet Draft draft-lasserre-nvo3-framework-03.
[83] T. Narten, M. Sridharan, D. Dutt, D. Black, and L. Kreeger, "Problem statement: Overlays for network virtualization," Jul. 2012, Internet Draft draft-narten-nvo3-overlay-problem-statement-03.
[84] L. Yong, M. Toy, A. Isaac, V. Manral, and L. Dunbar, "Use cases for DC network virtualization overlays," Jul. 2012, Internet Draft draft-mity-nvo3-use-case-01.
[85] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "VXLAN: A framework for overlaying virtualized layer 2 networks over layer 3 networks," Feb. 2012, Internet Draft draft-mahalingam-dutt-dcops-vxlan-01.
[86] M. Sridharan, A. Greenberg, N. Venkataramiah, Y. Wang, K. Duda, I. Ganga, G. Lin, M. Pearson, P. Thaler, and C. Tumuluri, "NVGRE: Network virtualization using generic routing encapsulation," Jul. 2012, Internet Draft draft-sridharan-virtualization-nvgre-01.
[87] B. Davie and J. Gross, "A stateless transport tunneling protocol for network virtualization (STT)," Mar. 2012, Internet Draft draft-davie-stt-01.

Anjing Wang received the B.Eng. and M.Eng. degrees from the Department of Information and Communication, Xi'an Jiaotong University, China, and the Ph.D. degree in computer science from North Carolina State University, Raleigh.

He is a Senior Software Engineer at Ericsson, Silicon Valley, CA. His research interests include network architecture, routing protocols and network system design. He is the author of several patents in the wireless LAN area.

Dr. Wang is a member of the IEEE, a member of the ACM, and a member of the Optical Society of America.

Mohan Iyer received the B.E. degree from the University of Pune, Pune, India, and the Ph.D. degree from North Carolina State University, Raleigh.

He is a Senior Software Engineer at Oracle Corporation, working in the network virtualization team within the Solaris Core OS division. He has worked for two years as a Member of the Technical Staff in the Telecom Business Unit of Persistent Systems, Pune. His research interests include network design and optimization, network architectures and protocols.

Rudra Dutta was born in Kolkata, India, in 1968. He received the B.E. degree in electrical engineering from Jadavpur University, Kolkata, India, in 1991, the M.E. degree in systems science and automation from Indian Institute of Science, Bangalore, India in 1993, and the Ph.D. in computer science from North Carolina State University, Raleigh, in 2001.

From 1993 to 1997 he worked for IBM as a software developer and programmer in various networking related projects. He has been employed from 2001–2007 as Assistant Professor, and since 2007 as an Associate Professor, in the department of Computer Science at the North Carolina State University, Raleigh. During the summer of 2005, he was a visiting researcher at the IBM WebSphere Technology Institute in RTP, NC. His current research interests focus on design and performance optimization of large networking systems. His research is supported currently by grants from the National Science Foundation and industry, including a recent GENI grant from NSF.

Dr. Dutta has served as a reviewer for many premium journals, on NSF and DoE review panels, as part of the organizing committee of many premium conferences, including Program Co-chair for the Second International Workshop on Traffic Grooming. Most recently, he has served as Program Chair for the Optical Networking Symposium at IEEE Globecom 2008, Program Chair for IEEE ANTS 2009, and General Chair of IEEE ANTS 2010. He is currently serving on the Steering Committee of IEEE ANTS 2012, and on the editorial board of the Elsevier Journal of Optical Switching and Networking.

George N. Rouskas (F'12) received the Diploma in computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, and the M.S. and Ph.D. degrees in computer science from the College of Computing, Georgia Institute of Technology, Atlanta, GA.

He is a Professor of Computer Science at North Carolina State University. His research interests include network architectures and protocols, optical networks, network design and optimization, and performance evaluation. He is co-editor of the book *Next-Generation Internet Architectures and Protocols* (Cambridge University Press, 2011), author of the book *Internet Tiered Services* (Springer, 2009), and co-editor of the book *Traffic Grooming for Optical Networks* (Springer 2008).

Dr. Rouskas is founding co-editor-in-chief of the *Optical Switching and Networking Journal* and he has served on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE/OSA JOURNAL OF OPTICAL NETWORKING, *Computer Networks*, and *Optical Networks*. He is the General Chair for ICCCN 2013, and he has served as TPC or general chair for several conferences, including ICCCN 2011, the IEEE GLOBECOM 2010 ONS Symposium, BROADNETS 2007, IEEE LANMAN 2004 and 2005, and IFIP NETWORKING 2004. He is the recipient of a 1997 NSF CAREER Award, the 2004 ALCOA Foundation Engineering Research Achievement Award, and the 2003 NCSU Alumni Outstanding Research Award, and he was inducted in the NCSU Academy of Outstanding Teachers in 2004. He is the Secretary of the IEEE Optical Networking Technical Committee (ONTC), and he served as a Distinguished Lecturer for the IEEE Communications Society in 2010–2011.

Ilia Baldine (M'98) received the B.S. degree in computer science and mathematics from the Illinois Institute of Technology, Chicago, in 1993, and the M.S. and Ph.D. degrees in computer science from North Carolina State University, Raleigh, in 1995 and 1998, respectively.

He leads RENCI (Renaissance Computing Institute, The University of North Carolina at Chapel Hill) network research and infrastructure programs. He is a networking researcher with a wide range of interests, including high-speed optical network architectures, cross-layer protocol interactions, novel signaling schemes and network security. Before joining RENCI, Baldine was the principal scientist at the Center for Advanced Network Research at the Research Triangle Institute, and a network research engineer at the Advanced Network Research group at MCNC, where he was a team member and a leader of a number of federally funded research efforts.