Computer Networks 54 (2010) 862–876

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A survey of network virtualization ☆

N.M. Mosharaf Kabir Chowdhury [a,1], Raouf Boutaba [b,c,*]

[a] Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94702, United States
[b] Cheriton School of Computer Science, University of Waterloo, ON, Canada N2L 3G1
[c] Division of IT Convergence Engineering, POSTECH, Pohang KB 790-784, Republic of Korea

ARTICLE INFO

Article history:
Received 3 March 2008
Received in revised form 21 October 2009
Accepted 25 October 2009
Available online 31 October 2009
Responsible Editor: I.F. Akyildiz

Keywords:
Network virtualization
Virtual networks
Next-generation Internet architecture

ABSTRACT

Due to the existence of multiple stakeholders with conflicting goals and policies, alterations to the existing Internet architecture are now limited to simple incremental updates; deployment of any new, radically different technology is next to impossible. To fend off this ossification, *network virtualization* has been propounded as a diversifying attribute of the future inter-networking paradigm. By introducing a plurality of heterogeneous network architectures cohabiting on a shared physical substrate, network virtualization promotes innovations and diversified applications. In this paper, we survey the existing technologies and a wide array of past and state-of-the-art projects on network virtualization followed by a discussion of major challenges in this area.

1. Introduction

The Internet has been stunningly successful over the course of past three decades in supporting multitude of distributed applications and a wide variety of network technologies. However, its popularity has become the biggest impediment to its further growth. Due to its multi-provider nature, adopting a new architecture or modification of the existing one requires consensus among competing stakeholders. As a result, alterations to the Internet architecture have become restricted to simple incremental updates and deployment of new network technologies have become increasingly difficult [1,2].

To fend off this ossification, *network virtualization* has been propounded as a diversifying attribute of the future inter-networking paradigm. Even though architectural purists view network virtualization as a means for evaluating new architectures, the pluralist approach considers virtualization as a fundamental attribute of the architecture itself [1]. They believe that network virtualization can eradicate the *ossifying forces* of the Internet and stimulate innovation [1,2].

1.1. What is network virtualization?

A networking environment supports network virtualization if it allows coexistence of multiple virtual networks on the same physical substrate. Each *virtual network* (VN) in a *network virtualization environment* (NVE) is a collection of virtual nodes and virtual links. Essentially, a virtual network is a subset of the underlying physical network resources.

Network virtualization proposes decoupling of functionalities in a networking environment by separating the role of the traditional Internet Service Providers (ISPs) into two: *infrastructure providers* (InPs), who manage the physical infrastructure, and *service providers* (SPs), who create

☆ This work was jointly supported by the Natural Science and Engineering Council of Canada (NSERC) under its Discovery program, Cisco Systems, and WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0).

* Corresponding author. Address: Cheriton School of Computer Science, University of Waterloo, ON, Canada N2L 3G1. Tel.: +1 519 888 4820; fax: +1 519 885 1208.
E-mail addresses: mosharaf@cs.berkeley.edu (N.M.M.K. Chowdhury), rboutaba@cs.uwaterloo.ca (R. Boutaba).

[1] This work was completed when this author was a Master's student at the University of Waterloo.

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.10.017

virtual networks by aggregating resources from multiple infrastructure providers and offer end-to-end network services [2–4].

Specifically, network virtualization is a networking environment that allows multiple service providers to dynamically compose multiple heterogeneous virtual networks that coexist together in isolation from each other. Service providers can deploy and manage customized end-to-end services on those virtual networks for the end users by effectively sharing and utilizing underlying network resources leased from multiple infrastructure providers [4]. Such a dynamic environment will foster deployment of multiple coexisting heterogeneous network architectures without the inherent limitations found in the existing Internet.

However, as a research area network virtualization is mostly unexplored. Several technical challenges in terms of instantiation, operation, and management of virtual networks are either untouched or require further attention. This presents a wide range of theoretical as well as practical open problems and unique challenges. This paper examines the past and the state of the art in network virtualization and identifies key issues for future exploration.

1.2. Organization

The remainder of this paper is composed as follows: in Section 2, we review four existing technologies – virtual local area networks, virtual private networks, active and programmable networks, and overlay networks – that are closely related to the concept of network virtualization. Later in Section 3, we survey a number of past and present projects on network virtualization and related concepts followed by a summarization of the surveyed projects from different perspectives in Section 4. Section 5 identifies key research issues for further exploration based on a qualitative analysis of the surveyed work. We conclude in Section 6.

2. Technologies

The concept of multiple coexisting networks appeared in the networking literature in different capacities. In this section, we discuss four such incarnations: *Virtual Local Area Networks* (*VLAN*), *Virtual Private Networks* (*VPN*), *active and programmable networks*, and *overlay networks*.

2.1. Virtual local area network

A virtual local area network (VLAN) [5] is a group of hosts with a common interest that are logically brought together under a single broadcast domain regardless of their physical connectivity. Since VLANs are logical entities, i.e., configured in software, they are flexible in terms of network administration, management, and reconfiguration. Moreover, VLANs provide elevated levels of trust, security, and isolation, and they are cost-effective.

Classical VLANs are essentially Layer 2 constructs, even though implementations in different layers do exist. All frames in a VLAN bear a common VLAN ID in their MAC headers, and VLAN-enabled switches use both the destination MAC address and the VLAN ID to forward frames. This process is known as *frame coloring*. Multiple VLANs on multiple switches can be connected together using *trunking*, which allows information from multiple VLANs to be carried over a single link between switches.

2.2. Virtual private network

A virtual private network (VPN) [6–8] is a dedicated communications network of one or more enterprises that are distributed over multiple sites and connected through tunnels over public communication networks (e.g., the Internet).

Each VPN site contains one or more Customer Edge (CE) devices (e.g., hosts or routers), which are attached to one or more Provider Edge (PE) routers. Normally a VPN is managed and provisioned by a VPN service provider (SP) and known as Provider-provisioned VPN (PPVPN) [9]. While VPN implementations exist in several layers of the network stack, the following three are the most prominent ones.

2.2.1. Layer 3 VPN

Layer 3 VPNs (L3VPN) [10,11] are distinguished by their use of layer 3 protocols (e.g., IP or MPLS) in the VPN backbone to carry data between the distributed CEs. L3VPNs can again be classified into two categories: CE-based and PE-based VPNs.

In the *CE-based VPN* approach, CE devices create, manage, and tear up the tunnels without the knowledge of the SP network. *Tunneling* requires three different protocols:

(1) *Carrier protocol* (e.g., IP), used by the SP network to carry the VPN packets.
(2) *Encapsulating protocol*, used to wrap the original data. It can range from very simple wrapper protocols (e.g., GRE [12], PPTP [13], L2TP [14]) to secure protocols (e.g., IPSec [15]).
(3) *Passenger protocol*, which is the original data in customer networks.

Sender CE devices encapsulate the passenger packets and route them into carrier networks. When the encapsulated packets reach the receiver CE devices at the end of the tunnels, they are extracted and actual packets are injected into receiver networks.

In *PE-based L3VPNs*, the SP knows that certain traffic is VPN traffic and process them accordingly. The VPN states are stored in PE devices, and a connected CE device behaves as if it were connected to a private network.

2.2.2. Layer 2 VPN

Layer 2 VPNs (L2VPNs) [16,17] provide end-to-end layer 2 connection between distributed cites by transporting Layer 2 (typically Ethernet but also ATM and Frame Relay) frames between participating sites. The primary advantage of L2VPN is its support of heterogeneous higher-level protocols. But its lack of a control plane takes away its capability of managing reachability across the VPN.

There are two fundamentally different kinds of Layer 2 VPN services that an SP could offer to a customer: point-to-point Virtual Private Wire Service (VPWS) and point-to-multipoint Virtual Private LAN Service (VPLS). There is also the possibility of an IP-only LAN-like Service (IPLS), which is similar to VPLS except that CE devices are hosts or routers instead of switches and only IP packets are carried (either IPv4 or IPv6).

2.2.3. Layer 1 VPN

Accompanied by the rapid advances in next-generation SONET/SDH and optical switching along with GMPLS [18] control, the Layer 1 VPN (L1VPN) [19,20] framework emerged from the need to extend L2/L3 packet-switching VPN concepts to advanced circuit-switching domains. It enables multiple virtual client-provisioned transport networks over a common Layer 1 core infrastructure. The fundamental difference between L1VPNs and L2 or L3 VPNs is that in L1VPNs data plane connectivity does not guarantee control plane connectivity (and vice versa).

The main characteristic of L1VPN is its multi-service backbone where customers can offer their own services with payloads of any layer (e.g., ATM, IP, TDM). This allows each service networks to have independent address space, independent Layer 1 resource view, independent policies, and complete isolation.

L1VPN can be of two types: Virtual Private Wire Services (VPWS) and Virtual Private Line Services (VPLS). VPWS services are point-to-point, while VPLS can be point-to-multipoint.

2.3. Active and programmable networks

While active and programmable networks may not be considered as direct instances of network virtualization, most of the projects in this area pushed forward the concept of coexisting networks through programmability. In order to allow multiple external parties to run possibly conflicting code on the same network elements, active and programmable networks also provide isolated environments to avoid conflicts and network instability.

The programmable networks community discusses how communications hardware can be separated from control software. Two separate schools of thought emerged on how to actually implement such concepts: one from telecommunications community and the other from IP networks community [21].

2.3.1. Open signaling approach

Open signaling takes a telecommunication approach to the problem with a clear distinction between transport, control, and management planes that constitute programmable networks and emphasize QoS guarantees for created services [21]. It argues for modeling communication hardware using a set of open programmable network interfaces to enable controlled access to switches, routers, and eventually network states by external parties.

2.3.2. Active networks approach

The active networks [22] community allow routers and switches to perform customized computations based on packet contents, and they also allow network elements to modify packets. The active networks approach allows customization of network services at packet transport granularity instead of doing so through a programmable control plane. The result is increased flexibility through a more complex programming model with higher security risks.

Different suggestions on levels of programmability exist in active networks literature. At the one end, ANTS [23] offers a Turing-complete machine model at the active router enabling each user to execute any new code. At the other end of the spectrum, DAN [24] only allows the user to call functions already installed at a particular node. However, due to lack of interest from network operators to open up their networks to external parties, none of the proposals are in use.

2.4. Overlay networks

An overlay network is a virtual network that creates a virtual topology on top of the physical topology of another network. Nodes in an overlay network are connected through virtual links which correspond to paths in the underlying network. Overlays are typically implemented in the application layer, though various implementations at lower layers of the network stack do exist.

Overlays are not geographically restricted, and they are flexible and adaptable to changes and easily deployable in comparison to any other network. As a result, overlay networks have long been used to deploy new features and fixes in the Internet. A multitude of overlay designs have been proposed in recent years to address diverse issues, which include: ensuring performance [25] and availability [26] of Internet routing, enabling multicasting [27–29], providing QoS guarantees [30], protecting from denial of service attacks [31,32], and for content distribution [33], file sharing [34] and even in storage systems [35]. Overlays have also been used as testbeds (e.g., PlanetLab [36]) to design and evaluate new architectures. In addition, highly popular and widely used peer-to-peer [34] networks are also overlays in the application layer.

However, in their seminal paper on network virtualization, Anderson et al. [1] point out that existing overlay technologies cannot be considered as a deployment path for disruptive technologies because of two main reasons. First, they are mostly used to deploy narrow fixes to specific problems without any holistic view of the interactions between coexisting overlays. Second, most overlays, being designed and deployed in the application layer on top of IP, are not capable of supporting radically different architectures.

3. Network virtualization projects

Historically "*virtual network*" has been a popular key phrase among networking researchers for describing projects on virtual private networks, overlay networks, and active or programmable networks. In this section, we summarize the key characteristics of a wide range of vir-

tual network architectures and related projects (e.g., overlay, programmable network, or VPN inspired designs).

3.1. Characteristics

In the absence of an established nomenclature for network virtualization, each research group have used its own set of terminologies to describe their work. However, a close observation reveals a set of governing characteristics that regulate the construction of these prototypes. We use the following set of characteristics to better understand the field:

(1) *Networking technology*, which implicitly determines the attributes of the virtual networks deployed on a particular networking platform by its unique set of characteristics. For example, a virtualization architecture based on wired networks (e.g., X-Bone) will obviously be more scalable and flexible in terms of bandwidth than one based on wireless or sensor networks.
(2) *Layer of virtualization*, which refers to the layer in the network stack where virtualization is introduced; the lower it is, the higher the flexibility of a virtual network deployed on that platform. Over the years, researchers have attempted to virtualize different layers of the network stack, starting from the physical layer (e.g., UCLP) and continuing up to the application layer (e.g., PlanetLab).
(3) *Architectural domain*, which indicates the targeted architectural and application domain, and dictates the design choices taken in the construction of architectures and services that can be offered on those platforms. Examples include, network management (e.g., VNRMS), virtual active networks (e.g., NetScript), and spawning networks (e.g., Genesis).
(4) *Granularity of virtualization*, which refers to the granularity at which each virtual network can administer itself. At one end of this spectrum, node virtualization creates virtual networks by connecting virtual machines on different nodes (e.g., PlanetLab). At the other end, CABO and NouVeau propose true plurality where each virtual network has a semblance of the native network.

3.2. Networking technology

3.2.1. IP networks: X-Bone

X-Bone [37,38] was first proposed as a system for rapid and automated deployment and management of overlay networks using encapsulation to enable virtual infrastructure. Later this idea was extended to the concept of *Virtual Internet* (*VI*) [39], which is an IP network composed of tunneled links among a set of virtual routers and hosts, with dynamic resource discovery, deployment, and monitoring support.

A VI virtualizes all the components of the Internet: hosts, routers and links between them. A single network node may participate as a virtual host (VH), a virtual router (VR), or multiple of them simultaneously in a VI. All components participating in the VI must support multihoming, since even a base host with a single VH is necessarily a member of at least two networks: the Internet and the VI overlay. Addresses within each VI is unique and can be reused in another overlay, unless there is no shared common node in the underlying network between the two VIs.

VIs completely decouple the underlying physical network from the overlays and multiple VI can *coexist* together. VIs also support control recursion to allow divide-and-conquer network management and network recursion to stack one VI on another.

Recently, P2P-XBone [40], a peer-to-peer based fusion of X-Bone, was proposed to enable dynamic join/leave of participating nodes from a VI. It also allows creation and release of dynamic IP tunnels, and customized routing table configuration.

3.2.2. ATM networks: Tempest

Tempest [41] is a network control architecture that allows multiple heterogeneous control architectures to run simultaneously over single physical ATM network. It is defined as a set of policies, algorithms, mechanisms, and protocols to control and manage various devices on the network.

Tempest is based on the concept of *switchlets* [42], which allows a single ATM switch to be controlled by multiple controllers by strictly partitioning the resources of that switch between those controllers. The set of switchlets that a controller or group of controllers possess forms its virtual network. Third parties can lease such virtual networks from the Tempest network operator to use them for any purpose as they see fit.

Programmability in Tempest is supported at two levels of granularity: first, switchlets support the introduction of alternative control architectures in the network; and second, services can be refined by dynamically loading programs into the network that customize existing control architectures. This allows the users to have *application-specific* control.

3.3. Layer of virtualization

3.3.1. Physical layer: UCLP

UCLP[2] is a distributed network control and management system for CA*NET 4 network that allows end users to treat network resources as software objects, and lets them provision as well as dynamically reconfigure optical networks (at Layer 1). Users are able to join or divide lightpaths within a single domain, or across multiple independent management domains to create customized logical IP networks.

UCLP takes a modular approach to resource management by introducing three distinct service layers [43,44]. Customers and administrators configure and use end-to-end UCLP resources through the *user access layer*. The *service provisioning layer* manages service logic and data regarding lightpaths. Finally, the *resource management layer* deals with actual physical resources.

[2] http://www.uclp.uwaterloo.ca/.

UCLPv1.4 [45] introduced dynamic topology discovery process and enabled auto-routing through intelligent algorithms alongside already available manual lightpath configuration capabilities. Later, UCLPv2 [46,47] extended UCLP with the use of Service Oriented Architecture (SOA) and workflow technologies with an aim to form the underpinning architectural framework for extending UCLP to allow the interconnection of instruments, time slices, and sensors; and for incorporating virtual routers and switches.

3.3.2. Link layer: VNET

VNET [48] is a Layer 2 overlay network for virtual machines (VMs) that implements a virtual LAN (VLAN) spread over a wide area using Layer 2 tunneling protocol (L2TP). Each physical machine hosting a virtual machine (VM) runs a VNET process that intercepts VM traffic and tunnels it to the appropriate destination. The destination is either another VM that can be contacted directly through VNET or an address external to the overlay. Traffic destined for an external address is routed through the overlay to a VNET proxy node, which is responsible for injecting the packets onto the appropriate network. The overlay thus consists of a set of TCP connections or UDP peers (VNET links) and a set of rules (VNET routes) to control routing on the overlay.

Since VNET operates at Layer 2, it is agnostic to Layer 3. As a result, protocols other than IP can be used. In addition, VNET also supports migration of a VM from one machine to another without any participation from the VM's OS and all connections remain open after migration.

3.3.3. Network layer: AGAVE

The main objective of the AGAVE [49–51] project is to provide end-to-end QoS-aware service provisioning over IP networks following the theme of QoS forwarding mechanisms such as IntServ [52] and DiffServ [53,54]. To achieve this, AGAVE proposes a new inter-domain architecture based on the novel concept of Network Planes (NPs), which allows multiple IP Network Providers (INPs) to build and provide Parallel Internets (PIs) tailored to end-to-end service requirements.

NPs are internal to INPs and are created based on the service requirements described by the SPs. An NP can be engineered for routing, forwarding, or resource management. To enable end-to-end services over multi-provider environment, NPs from different INPs are connected together to form PIs based on inter-INP agreements. One of the interesting feature of AGAVE is that it does not require all the NPs participating in a PI to be homogeneous resulting in greater flexibility.

AGAVE replaces node-centric provisioning/configuration approach in favor of a more centralized network-based configuration, which ensures configuration consistency between participating INPs and reduces misconfiguration errors. Also, it supports an NP emulation function that assesses the status of the network and evaluates the impact of introducing new NPs before accepting new IP-connectivity provisioning requests.

3.3.4. Application layer: VIOLIN

VIOLIN [55,56] is an application-level virtual network architecture, where isolated virtual networks are created in software on top of an overlay infrastructure (e.g., PlanetLab). Capitalizing on the advances in VM technologies, VIOLIN extends the idea of single node isolation in VMs to provide completely isolated virtual networks.

A VIOLIN consists of virtual routers (vRouters), LANs (vLANs) and end hosts (vHosts), all being software entities hosted by overlay hosts. Both vHosts and vRouters are virtual machines running in physical overlay hosts. A vLAN is created by connecting multiple vHosts using virtual switches (vSwitches), while vRouters connect multiple vLANs to form the total network.

VIOLIN provides network isolation with respect to: (i) administration, (ii) address space and protocol, (iii) attack and fault impact, and (iv) resources. The combined effect is a confined, secured, and dedicated environment that can be used to deploy untrusted distributed applications and perform risky network experiments.

3.4. Architectural domain

3.4.1. Network management: VNRMS

VNRMS [57–59] is a flexible and customizable virtual network management architecture, which provides a programmable networking environment to generate multiple levels of virtual networks through nesting from a single physical network (PN). A virtual network is composed of several virtual network resources (VNRs), where each VNR is a subset of a physical network resource (PNR) in the underlying network. VNRMS lets the customers to customize the VNRs through active *resource agents* using a customer-based management system (CNRMS). While the provider VNRMS has access to all the resource agents, a customer can access only those that belong to its virtual network.

In order to allow a CNRMS to manage only a subset of resources in a PNR, the management information base (MIB) of that PNR is logically partitioned into multiple disjoint MIBs, known as MIBlets [60]. MIBlets provide *abstract* and *selective* views of the resources that are allocated to a particular virtual network. An abstract view hides the details of the resource interface that are not relevant to the CNRMS. A selective view restricts the CNRMS to access only the resources allocated to it.

3.4.2. Virtual active networks: NetScript

NetScript [61] is a language system for dynamically programming and deploying protocol software in an active network. It is a strongly typed language that creates universal language abstractions to capture network programmability. Unlike other active network architectures, where packets contain active programs, NetScript packets are passive. These packets are processed by protocol software or hardware when they flow through the network. In this architecture, active packet processing applications and standardized protocols can be composed together, interoperate, and utilize each other's services. Consequently, NetScript can be used to systematically compose, provision, and manage virtual active network abstractions [62].

NetScript supports creation of arbitrary packet formats and dynamic composition of standard and active protocol and it can operate on any type of packet stream. NetScript communication abstractions consider network nodes as collections of Virtual Network Engines (VNEs) interconnected by Virtual Links (VLs) that constitute NetScript Virtual Networks (NVNs) [63].

3.4.3. Spawning networks: Genesis

The Genesis Kernel [64] is a *spawning network* [65,66], a variant of open programmable networks, that automates the life cycle process for the creation, deployment, management, and designing of network architectures. It allows multiple heterogeneous child virtual networks to operate on top of subsets of their parent's resources, and provides isolation among them. The Genesis Kernel also supports nesting of virtual networks and inheritance of architectural components from parent to child networks.

A virtual network in the Genesis Kernel is characterized by a set of routelets interconnected by a set of virtual links. Routelets represent the lowest level of operating system support dedicated to a virtual network, and are designed to operate over a wide variety of networking technologies including IP and ATM technology. They process packets along a programmable data path at the inter-networking layer, while virtual network kernel makes control algorithms support programmability.

3.4.4. Experimental facility: FEDERICA

FEDERICA [67,68] addresses data, control, and management plane challenges in the virtualization capable network infrastructure. It aims to provide an agnostic and transparent infrastructure, which will support isolated coexisting slices with complete user control to the lowest possible layer. Multiple slices can interconnect between themselves and connect to external networks and services to create large federations.

FEDERICA stresses on *reproducibility* of experiments, i.e., given the same initial conditions, the results of an experiment will remain the same. It promotes the use of programmable high-end routers and switches in the core nodes, PC-based virtualization capable non-core nodes, and multi-protocol switches connecting the core nodes to their non-core counterparts. Virtual network assignment is done by a centralized admission control and decision making procedure. A dedicated proxy keeps user slices and FEDERICA resources assigned to different experiments safe from unauthorized accesses.

3.5. Granularity of virtualization

3.5.1. Node virtualization: PlanetLab

PlanetLab [36,69] is an overlay-based testbed that was developed to design, evaluate, and deploy geographically distributed network services with support for researchers and users. Its goal is to create a *service-oriented network architecture* combining the best of both the distributed systems community and the networks community.

PlanetLab is built upon four design principles. First, it supports *sliceability*. That is, each application acquires and runs in a slice of the overlay. Virtual machine monitors (VMMs) running on each node allocate and schedule slices of the nodes' resources to create a distributed virtualized environment. Second, it supports a highly decentralized control structure, enabling nodes to act according to local policies. Third, overlay management is divided into subservices that run on their own slices, instead of a centralized one. Finally, overlay supports an existing and widely adopted programming interface, with internal changes over time keeping the API intact, to promote actual long-term service development instead of just being a temporary testbed.

PlanetLab provides *resource monitor* and *resource broker* services to handle the resource management. To obtain a slice a user first contacts a resource broker, then goes through admission control process in each of the nodes assigned by the broker and finally it launches its service by bootstrapping itself in the resulting slice.

3.5.2. GENI

Based on the experience accumulated from using PlanetLab and other similar testbeds, the Global Environment for Network Innovations (GENI) [70] is a major initiative of the US National Science Foundation (NSF) to build an open, large-scale, realistic experimental facility for evaluating new network architectures, carrying real traffic on behalf of end users, and connecting to the existing Internet to reach external sites. The purpose of GENI is to give researchers the opportunity to create customized virtual network and experiment unfettered by assumptions or requirements of the existing Internet.

Main design goals of GENI [70] include: sliceability to share resources, generality to give an initial flexible platform for the researchers, fidelity, diversity and extensibility, wide deployment and user access for testing and evaluation purposes as well as actual use of deployed services and prototypes, controlled isolation and monitoring facilities.

GENI proposes virtualization in the form of slices of resources in space and time. If resources are partitioned in time, a given resource might not sustain real user workload, thereby limiting its feasibility for deployment studies. On the other hand, if resources are partitioned in space, only a limited number of researchers might be able to include a given resource in their slices. In order to maintain balance, GENI proposes to use both types of virtualization based on resource type. If sufficient capacity is available to support deployment studies, GENI uses time-based slicing; otherwise, it partitions resources in space to support a handful of high priority projects instead of making those resources available to everyone.

3.5.3. VINI

VINI [71] is a virtual network infrastructure allowing network researchers to evaluate their protocols and services in a realistic environment with high degree of control. It can be viewed as an extension to PlanetLab toward GENI, that will be able to provide infrastructure like PlanetLab along with the support for virtual networks as in X-Bone or VIOLIN. However, VINI offers more latitude to researchers than PlanetLab at routing level. It provides the ability to create real complex networks and to inject

exogenous events to create more realistic alternative to simulation and emulation of proposed network architectures.

Initial prototype of VINI (PL-VINI) was implemented on PlanetLab by synthesizing a collection of available software components. It can be considered as a specific instantiation of an overlay network that runs software routers and allows multiple such overlays to exist in parallel. In particular, it used *XORP* for routing [72], *Click* for packet forwarding and network address translation [73], and *OpenVPN*[3] servers to connect with end users.

Recently a software platform for hosting multiple virtual networks on shared physical network infrastructure, Trellis [74], has been developed. Trellis synthesizes container-based virtualization technologies together with a tunneling mechanism into a coherent platform to achieve the following design goals: performance, scalability, flexibility, and isolation. It allows each virtual network to define its custom topology, routing protocols, and forwarding tables.

3.5.4. Full virtualization: CABO

At present, ISPs manage their network infrastructure as well as provide network service to end users. Adopting a new architecture not only requires change in hardware and host software, but it also requires that ISPs jointly agree on any architectural change [1]. CABO [3] promotes separation between infrastructure providers and service providers to end this deadlock. CABO exploits virtualization to allow service providers to simultaneously run multiple end-to-end services over equipment owned by different infrastructure providers.

CABO supports automatic migration of virtual routers from one physical node to another [75], introduces accountability to provide guarantees to service providers [76], proposes a multi-layer routing scheme that is scalable and quick to react to any changes in network conditions [77]. In supporting programmable routers, CABO resembles the theme introduced in active networks research, except that it does not enable users to program the network; rather service providers can customize their networks to provide end-to-end service to the end users.

3.5.5. 4WARD

4WARD [78,79] virtualization framework promises coexistence of multiple networks on a common platform through carrier-grade virtualization of networking resources. It provides means to support on-demand instantiation and dependable inter-operation between heterogeneous virtual networks in a secured and trusted commercial setting. 4WARD also supports virtualization of heterogeneous networking technologies (e.g., wired and wireless), heterogeneous end user devices, and novel networking protocols as part of its core architectural design.

4WARD business model introduces three roles [79]: infrastructure providers, who manage the underlying network resources; virtual network providers (VNP), who create virtual networks; and virtual network operators (VNO), who connect customers to the services provided in different virtual networks. 4WARD is still in its incipient stage with little implementation and significant similarities with other recent proposals. However, the defining characteristic of 4WARD is its promise of bringing network virtualization to the end users as opposed to limiting its scope to experimental networks and testbeds.

3.5.6. NouVeau

NouVeau [80–82] aims for a flexible, manageable, and secure end-to-end network virtualization environment by creating a holistic framework synthesizing the best features from the existing proposals with its own. NouVeau proposes two major roles: infrastructure providers and service providers, but supports a more competitive economic value chain through *recursion* of virtual networks and *inheritance* of parents' properties to child virtual networks.[4] It also supports *revisitation*[5] of virtual nodes to increase manageability. With an aim to provide realistic network virtualization support for end users, NouVeau considers heterogeneity of networking technologies, end user devices, networking protocols, and management paradigms.

Manageability is considered to be the biggest concern for network virtualization and next-generation Internet in general. To this end, it provides algorithms for resource management [82], framework for end-to-end identity management [81], survivable resource allocation mechanisms for fault-tolerance, and mechanisms for creating and managing agreements between service providers and infrastructure providers.

NouVeau argues for secure programming paradigms that are managed by the infrastructure providers with controlled exposure of customization functionalities to the service providers. It supports creation of customized virtual networks in any layer of the networking stack through recursion and inheritance.

4. Discussion

This section presents a qualitative comparison of the surveyed network virtualization projects from three different perspectives: shifting trends in network virtualization research, influence of existing technologies (e.g., VLAN, VPN, etc.), and realization of diverse design goals in the surveyed projects.

4.1. Shifting trends

A summary of the characteristics of the surveyed network virtualization projects is presented in Table 1 (tabularized roughly in chronological order) [4]. One can observe the presence of three unique trends in the progress of network virtualization research over time proceeding

[3] http://www.openvpn.net/.

[4] A service provider with a virtual network in layer N can create a child virtual network – and lease it – to act as a virtual infrastructure provider to another service provider's virtual network in layer $(N+1)$ [80].

[5] Revisitation allows one physical node to host more than one virtual nodes from the same virtual network[38].

Table 1
Characteristics of different network virtualization projects.

Project	Influences of existing concepts	Architectural domain	Networking technology	Layer of virtualization	Granularity of virtualization	References
VNRMS	Programmable networks, VPN	Virtual network management	ATM/IP		Node/link	[57–59]
Tempest	Programmable networks	Enabling alternate control architectures	ATM	Link		[41,42]
NetScript	Active networks	Dynamic composition of services	IP	Network	Node	[61,62]
Genesis	Programmable networks	Spawning virtual network architectures		Network	Node/link	[64–66]
VNET	VLAN, L2VPN	Virtual machine grid computing		Link	Node	[48]
VIOLIN	L2VPN, overlays	Deploying on-demand value-added services on IP overlays	IP	Application	Node	[55,56]
X-Bone	L3VPN, overlays	Automating deployment of IP overlays	IP	Network	Node/link	[37,38]
PlanetLab	Overlays	Deployment and management of overlay-based testbeds	IP	Application	Node	[36]
UCLP	L1VPN, SOA	Dynamic provisioning and reconfiguration of lightpaths	SONET	Physical	Link	[44,46,47]
AGAVE	IntServ, DiffServ, VPN, overlays	End-to-end QoS-aware service provisioning	IP	Network		[49–51]
GENI	VPN, active and programmable networks, overlays	Creating customized virtual network testbeds	Heterogeneous			[70]
VINI	VPN, overlays	Evaluating protocols and services in a realistic environment		Link		[71]
CABO	DiffServ, VPN, active and programmable networks, overlays	Deploying value-added end-to-end services on shared infrastructure	Heterogeneous		Full	[3]
4WARD	Overlays, SOA, autonomic networks	Instantiation,deployment, and management of virtual networks in a commercial setting	Heterogeneous	Network	Full	[78,79]
NouVeau	DiffServ, overlays, active and programmable networks, VPN, autonomic networks	Deploying end-to-end virtual networks on shared infrastructure	Heterogeneous		Full	[80–82]
FEDERICA	SOA, IaaS, VPN	Experimental facility with reproducibility	Heterogeneous	Link	Node/link	[67,68]

toward establishing a holistic and generalized future networking environment.

First of all, unlike the previous projects (e.g., X-Bone, VIOLIN, PlanetLab) that were focused more on connecting virtualized nodes or deploying virtual links between physical nodes, recent proposals (e.g., GENI, 4WARD, NouVeau) strive for achieving not only node or link virtualization but also virtualization of other aspects of networking (e.g., management planes) through effective isolation.

Another important development is pushing virtualization to lower layers of the network stack (e.g., FEDERICA, CABO). The main intuition behind this trend is the observation that the lower layer virtualization would take place, the more agnostic virtual networks would become to higher layer protocols. However, such flexibility comes with a downside; there are no network wide control and management planes. This problem is currently being addressed in ongoing projects using different approaches (e.g., centralized authority) without any definitive answer.

Finally, with the increasing number of mobile and wireless devices and the onset of specialized networking technologies (e.g., sensor networks), network virtualization researchers are trying to accommodate multiple heterogeneous technologies together in an integrated environment (e.g., GENI, 4WARD, FEDERICA, NouVeau). While the previous projects focused on exploiting characteristics of a particular technology (e.g., UCLP is for optical networks only), current research is more about bridging gaps between diverse technologies.

4.2. Influence of existing concepts

Network virtualization literature borrows heavily from past technologies to enable multiple coexisting logical networks (Section 2). On the one hand, tunneling mechanisms in VPNs and overlays set the standard for creating virtual links; on the other hand, programmability in active and programmable networks act as the inspiration behind allowing high level of flexibility in the physical nodes to host multiple virtual nodes.

In addition, concepts like *Quality of Service* (*QoS*), *Differentiated Services* (*DiffServ*) [53,54], *Integrated Services* (*IntServ*) [52], *Service Oriented Architecture* (*SOA*) [83], *Infrastructure as a Service* (*IaaS*), and Cloud Computing [84] also play roles in defining and designing network virtualization projects. Table 1 summarizes the influence of these concepts on the existing projects.

4.3. Design goals

Over time research in network virtualization gradually progressed from focusing on a particular objective to covering multiple ones. While the past projects originating from VPNs or active and programmable networks focused

individually on security, flexibility, or programmability, the more recent ones (e.g., CABO, FEDERICA, NouVeau) take a concerted effort to cover most of them in a single package. A close scrutiny reveals a recurrent set of design goals that provide guidelines to design protocols and algorithms for network virtualization environment.

4.3.1. Flexibility and heterogeneity

All the network virtualization projects provide certain amount of flexibility and heterogeneity, which is determined by the underlying networking technology and the layer at which virtualization is administered. The lower layer virtualization is introduced, the easier it is to introduce higher amount of flexibility and heterogeneity. For example, L2VPNs are agnostic to layer 3 protocols since each of the VPNs is separated at the link layer without having to consider the implications of layer 3 payloads. Same is the case for network virtualization.

Moreover, dependence on specific technologies also reduce the amount of flexibility in a network virtualization environment. For example, virtualization on top of IP substrate already have fixed network layer protocols; hence, virtual networks on those network virtualization environments cannot deploy IP independent mechanisms. A combined effect of these observations can be seen in the shifting trend of network virtualization projects over time toward supporting heterogeneous networking technologies and pushing virtualization at lower and lower layers (Table 1).

4.3.2. Manageability

Manageability in network virtualization has been addressed both at micro level and at macro level in the surveyed projects. At micro level, VNRMS virtualizes the MIBs into MIBlets for easier management of the virtual resources allocated to separate networks. Migration of virtual routers considered in recent projects is another example of micro level management.

CABO propounds network virtualization as a tool for introducing accountability at every strata of networking to improve manageability. In addition, explicit separation of the service providers from the infrastructure providers in CABO and NouVeau also increases manageability at macro level by creating well-defined management responsibilities.

4.3.3. Isolation

Isolation between coexisting virtual networks has been addressed in different capacities in different projects. Most previous projects focused primarily on logical isolation. VIOLIN went the farthest by providing network isolation with respect to administration, address space and protocol, attack and fault impact, and resources. UCLP, on the other hand, provided physical isolation between virtual network lightpaths in the physical (optical) layer.

In recent proposals (e.g., VINI, CABO, NouVeau), isolation has been considered as an indispensable part of the solution with an aim to provide high level of security, privacy, and fault-tolerance between the coexisting heterogeneous virtual networks. Proposed level of isolation in these projects is both logical for ease of administration and management and physical to ensure increased security as well as privacy.

4.3.4. Programmability

Programmability in the earlier projects was understandably expressed in two main forms: defined programmable interfaces (e.g., Tempest, Genesis) and active code (e.g., NetScript). However, in recent projects (e.g., NouVeau) there is no such explicit distinction. The focus is more on enabling a secure programming paradigm with considerable level of flexibility for service providers to be able to deploy customized end-to-end virtual network services without significant compromises from infrastructure providers.

4.3.5. Experimental and deployment facility

Experimentation has always been one of the main motivating factors behind research on network virtualization. In fact, network virtualization originated from the inability of the then-existing testbeds to produce isolated, reliable, and reproducible experimental conditions. Starting from PlanetLab and continuing up to the recent projects (e.g., GENI, VINI, and FEDERICA), providing experimental facility with real traffic, realistic network conditions, and reproducibility of network events has remained a major focus of the existing projects.

While experimentation facilities motivated the academia, commercial institutions have been inspired by the possibility of a fast and reliable deployment path for novel services. CABO, 4WARD, AGAVE, and NouVeau are examples of network virtualization projects that directly address such business concerns.

4.3.6. Legacy support

Network virtualization projects can broadly be categorized into two different classes based on their approaches toward legacy support. On the one hand, recent projects, mostly in the *clean-slate design* camp (e.g., NouVeau, CABO, 4WARD), call for a complete redesign of the networking paradigms by breaking away from the existing practice. While such revolution will give an immense amount of freedom, it might not be practical due to cost and business concerns. On the other hand, *evolutionary approaches* (e.g., FEDERICA, AGAVE) propound a gradual transformation, without completely ignoring the existing solutions, for economic viability.

5. Key research directions

Existent network virtualization related research mostly focus on fixing some of the lingering problems of the current Internet. As a result, several technical challenges in terms of instantiation, operation and management of an overall network virtualization environment remain unexplored till today, and many others require modification and improvement [4]. Examples of instantiation related problems include interfacing, signaling, bootstrapping, and embedding of virtual networks on shared physical infrastructure; implementation of virtual routers and virtual links as well as resource scheduling among coexisting

virtual resources are a few of many operation related challenges; finally, failure handling, mobility management, virtual network configuration and monitoring are some examples of the management problems in the network virtualization environment. In this section, we discuss a wide range of open challenges, both theoretical and practical, under the light of previous work for further exploration.

5.1. Interfacing

Service providers synthesize physical resources from one or more infrastructure providers to create virtual networks. Infrastructure providers must provide well-defined interfaces to allow service providers to communicate and express their requirements. For interoperability, such interfaces should follow a standard that should be able to express virtual network requests in terms of virtual nodes and virtual links along with their corresponding attributes. An XML-based specification language can be a possible candidate in this respect.

Appropriate interfaces between end users and service providers, between infrastructure providers, and between multiple service providers must also be identified and standardized. Examples of such interfaces and agreements between collaborating parties can be found in the AGAVE framework.

5.2. Signaling and bootstrapping

Before creating a virtual network, a service provider must already have network connectivity to one or more infrastructure providers in order to issue its requests. This introduces a circularity where network connectivity is a prerequisite to itself [3]. As long as a network virtualization environment is not mature enough to support itself, signaling must be handled through *out-of-band* communication mechanisms (e.g., the current Internet).

Bootstrapping capabilities are required to allow service providers to customize the virtual resources allocated to them. Standard methods to make programmability of the network elements available to the service providers must also be developed [85]. Both signaling and bootstrapping call for at least another network that will always be present to provide connectivity to handle these issues. Genesis and Tempest follow this approach and provide a separate bootstrapping interface.

5.3. Resource allocation

Resource allocation in a network virtualization environment refers to *static* or *dynamic* allocation of virtual nodes and links on physical nodes and paths, respectively. It is also known as the *virtual network embedding problem* in the existing literature. Embedding of virtual networks, with constraints on nodes and links, can be reduced to the $\mathcal{NP}$-hard *multi-way separator problem* [86] even when all virtual network requests are known in advance.

In order to provide efficient heuristics, existing research has been restricting the problem space in different dimensions, which include: (i) considering offline version of the problem (i.e., all the requests are known in advance) [87–90], (ii) ignoring either node requirements or link requirements [91,87], (iii) assuming infinite capacity of the substrate nodes and links to obviate admission control [91,87,88], and (iv) focusing on specific topologies [87]. Yu et al. [92] addressed these issues by envisioning support from the substrate network through node and link migration as well as multi-path routing. Chowdhury et al. [82] proposed embedding algorithms based on the mathematical formulation of the embedding problem that outperform the previous algorithms in terms of acceptance ratio and total revenue. Unlike others following a centralized approach, Houidi et al. [93] proposed a distributed embedding algorithm but could not achieve competitive performance. All of these algorithms perform static resource allocation.

DaVinci [94] presents a dynamic allocation framework where each substrate link periodically reassigns bandwidth shares between the virtual links, but it does not consider dynamic allocation of virtual nodes. While DaVinci achieves better link utilization, it also gives a hint of best-effort mechanism found in the existing Internet. A careful investigation is required to validate such dynamic allocation schemes.

Finally, all the existing algorithms consider the presence of a single infrastructure provider. An *inter-domain virtual network embedding* is even more complicated. In this case, virtual network requests need to be divided and partially embedded onto different infrastructure provider resources, and then individual embeddings must be connected together based on inter-infrastructure provider policies and agreements. Inter-domain embedding is a mostly untouched area with possibilities for broker-based centralized as well as policy-based decentralized embedding algorithms.

5.4. Resource discovery

In order to allocate resources for requests from different service providers, infrastructure providers must be able to determine the topology of the networks they manage as well as the status of the corresponding network elements (i.e., physical nodes and interconnections between them) [3]. Furthermore, adjacent infrastructure providers must also share reachability information to be able to establish links between their networks to enable inter-domain virtual network instantiation. UCLP promotes a combination of *event-based* and *periodic* topology discovery using an additional topology database [45]. Events update the topology database of an infrastructure provider, and a periodic refresh ensures that even if some events were not notified, the topology database is fresh. CABO argues for the use of a separate discovery plane run by the infrastructure providers as proposed in the 4D network management architecture [95]. Efficiently gathering and dissemination of such information in decision elements could be achieved via discovery techniques discussed in existing distributed computing literature (e.g., Remos [96]).

5.5. Admission control and usage policing

Infrastructure providers must ensure that resources are not over-provisioned to uphold QoS guarantees. Consequently, they have to perform accurate accounting and implement admission control algorithms to ensure that resources allocated to the virtual networks do not exceed the physical capacity of the underlying network. Existing solutions perform admission control while statically embedding virtual networks [92,82]. However, they do not allow dynamic resizing of allocated resources (i.e., adding or removing virtual nodes or links, increasing or decreasing allocated capacities).

In order to avoid constraint violations by globally distributed virtual networks, distributed policing mechanisms must be employed to make sure that service providers cannot overflow the amount of resources allocated to them by direct or indirect means. Raghavan et al. [97] presented such a global rate limiting algorithm coordinated across multiple sites in the context of cloud-based services in the existing Internet. Similar mechanisms need to be developed in the context of network virtualization too.

5.6. Virtual nodes and virtual links

Commercial vendors have been promoting virtual routers and switches as tools for simplifying core network design, decreasing CAPEX, and for VPN purposes [98]. Similar concept can be extended with programmability to create substrate routers that will allow each service provider to customize their virtual routers. A conceptual construct of such substrate routers can be found in [2]. OpenFlow [99] enables programmability on commodity hardware using FPGA-based routers and switches with competitive performance. Examples of extensible and flexible virtual router software architectures include Click Modular Router [73] and VERA [100].

Performance of virtual routers on existing virtual machine systems should also be explored. Design and performance of virtual routers implemented on top of Xen virtual machine systems and the impact of current multi-core processors on their performance has been studied in [101]. RouteBricks [102] achieves up to 35 Gbps speed in software routers using many-core parallelization.

Scalability of a network virtualization environment is closely tied to the scalability of the physical routers. Commercial router vendors have already implemented routers that can hold multiple logical routers [103]. Fu and Rexford [104] present a mechanism that improves scalability by capitalizing on the commonality of address prefixes in multiple FIBs from different virtual routers to decrease memory requirements and lookup times.

To increase network manageability and to handle network failures, migration of virtual routers can be an effective solution [75]. But finding probable destinations for a migrating virtual router is restricted by multiple physical constraints like change of latency, link capacity, platform compatibility issues, and even capabilities of destination physical routers; it is still an open problem.

The ability to create tunnels over multiple physical links in Layers 3, 2, or 1 already exists in the context of L3, L2, and L1 VPNs, respectively. Similar protocols can be used in virtual networks too. The overhead for transporting packets across a virtual link must be minimal compared to that of transporting packets across a native link. This translates into minimum encapsulation and multiplexing cost.

5.7. Resource scheduling

When establishing a virtual network, a service provider requires specific guarantees for the virtual nodes' attributes as well as the virtual links' bandwidth allocated to its network [3]. For virtual routers, a service provider might request guarantees for a minimum packet processing rate of the CPU, specific disk requirements, and a lower bound on the size of the memory. On the other hand, virtual link requests may range from best-effort service to fixed loss and delay characteristics found in dedicated physical links. To provide such guarantees and to create an illusion of an isolated and dedicated network to each service provider, infrastructure providers must employ appropriate scheduling algorithms in all of the network elements.

Existing system virtualization technologies provide efficient scheduling mechanisms for CPU, memory, disk, and network interface in each of the virtual machines running on the host machine [105]. Network virtualization can extend these mechanisms to implement resource scheduling in the physical infrastructure. Previous results from research on packet scheduling algorithms for IP networks can also be useful in the design of schedulers.

5.8. Naming and addressing

Due to potential heterogeneity of naming and addressing schemes in coexisting virtual networks, end-to-end communication and universal connectivity is a major challenge in a network virtualization environment. In addition, end users can simultaneously connect to multiple virtual networks through multiple infrastructure providers using heterogeneous technologies to access different services, which is known as über-homing [81]. Incorporating support for such heterogeneity in multiple dimensions is a fundamental problem in the context of network virtualization.

Recently proposed iMark [81] separates identities of end hosts from their physical and logical locations to add an additional level of indirection and, with the help of a global identifier space, provides universal connectivity without revoking the autonomy of concerned physical and virtual networks. However, while conceptually possible, iMark is not physically implementable due to excessive memory requirements. Therefore, one key research direction in naming and addressing is to find a viable global connectivity enabling framework.

5.9. Dynamism and mobility management

Network virtualization environment is highly dynamic. At macro level, virtual networks with shared interests can be dynamically aggregated together to create *federation* of

virtual networks. Multiple federations and virtual networks can also come together to form *virtual network hierarchies* [81]. Aggregation and dissolution of control and data planes (e.g., naming, addressing, routing, and forwarding information) for macro level dynamism is an unresolved issue.

At micro level, mobility of end users from one physical location to another and migration of virtual routers for operation and management purposes [75] poses the biggest challenge. Finding the exact location of any resource or end user at a particular moment and routing packets accordingly is a complex research challenge that needs efficient solution. In addition, network virtualization allows end users to move logically from one virtual network to another, which further complicates the problem.

5.10. Virtual network operations and management

Network operations and management has always been a great challenge for the network operators. Division of accountability and responsibilities among different participators in a network virtualization environment promises increased manageability and reduced scopes for error [3]. Keller et al. [76] propose proactive and reactive mechanisms to enforce accountability for hosted virtual networks.

Considerable flexibility must be introduced from the level of network operations centers (NOCs) to intelligent agents at network elements, to enable individual service providers configure, monitor, and control their virtual networks irrespective of others. The concept of MIBlets [60] used in VNRMS to gather and process performance statistics for each of the coexisting virtual networks instead of using a common MIB can be a good starting point.

Since a virtual network can span over multiple underlying physical networks, applications must also be developed to aggregate information from diverse, often conflicting, management paradigms followed by participating infrastructure providers. Introducing a common abstraction layer, to be followed by all the management softwares, can be an effective solution [106].

Failures in the underlying physical network components can give rise to cascading failures in the virtual networks directly hosted on those components. For instance, a physical link failure will result in failures of all the virtual links that pass through it. Similarly, any physical node failure might require re-installations of all the service provider's custom softwares. Detection and effective isolation of such failures as well as prevention and recuperation from them to stable states are all open research challenges.

5.11. Security and privacy

Even though network virtualization strives for isolation of faults and attack impacts, it does not necessarily obviate existing threats, intrusions, and attacks to physical and virtual networks. In fact, to some extent, network virtualization gives rise to a new array of security vulnerabilities. For instance, a denial-of-service (DoS) or a distributed DoS (DDoS) attack against the physical network in a virtualized environment will affect all the virtual networks hosted on that network. Programmability of network elements – powerful and expressive in trusted hands – can increase vulnerability if there are security holes in programming models. To avoid such pitfalls, recent proposals (e.g., CABO) argue for controlled programmability by trading off flexibility for security without any definitive answer to permissible levels access to programmable hardware.

A detailed study of possible security vulnerabilities can give insights into developing programming paradigms [107] and virtualization environments that are secure and robust against known attacks. Established secured tunneling and encryption mechanisms (e.g., IPSec [15]) in VPNs can also be used in this context to increase security and enforce privacy.

5.12. Heterogeneity of networking technologies

Each networking technology has its own set of unique characteristics and poses challenges that require specific solutions for provisioning, operation, and maintenance of virtual network on those platforms. For instance, UCLP virtualizes optical networks capitalizing on the property of lightpaths that can be physically sub-divided into smaller lightpaths. Virtual Sensor Networks (VSN) [108], on the other hand, deal with providing protocol support for dynamic formation, usage, adaptation, and maintenance of subsets of sensors under unique power constraints. Similarly, virtualization of wireless networks using different multiplexing techniques creates different complications, e.g., node synchronization and managing device states [109].

End-to-end network virtualization requires framework that handle interactions between such contrasting underlying infrastructures while providing a generic and transparent interface for service providers to easily compose and manage virtual networks.

5.13. Network virtualization economics

In traditional network economics, bandwidth is the chief commodity of interest. But in its network virtualization counterpart, virtual nodes are important as well. In such a marketplace, service providers and infrastructure providers maintain buyer–seller relationships with brokers acting as mediators between these two parties. End users also participate as buyers of services from different service providers [80].

There are two general types of marketplaces: centralized and decentralized. Centralized marketplaces are efficient but vulnerable against attacks and not scalable. On the other hand, fully decentralized marketplaces are extensible and fault-tolerant but prone to malicious behavior and inefficiency. Hausheer and Stiller [110,111] present a semi-decentralized double-auction based marketplace for virtual network environments. However, there work focus mostly on virtual links, leaving incorporating virtual nodes to the economic model as an open challenge.

6. Conclusion

Most researchers agree that the Internet has reached a tipping point where most of their time and effort is spent in putting band aids on its existing flaws rather than in cultivating novel ideas. To fight back this ossification, redesign of the Internet is a bare necessity [112]. Instead of creating yet another *one-size-fits-all* architecture, a versatile networking paradigm must be established that will be flexible enough to support multiple coexisting architectures through network virtualization [1,2]. As a result, major initiatives on next-generation networks (e.g., FIND[6] projects in the US, FIRE[7] projects in the EU, Asia Future Internet (AsiaFI[8]), New Generation Network (NWGN) forum [113] in Japan, and Future Internet Forum (FIF[9]) in South Korea) all around the world are promoting inclusion of network virtualization concepts in their core architectural designs.

Moreover, network virtualization stands at a unique point in the current virtualization landscape as the missing link that will interconnect all other virtualized appliances – ranging from operating systems, storage systems to servers and even large data centers – to create a complete semblance of a virtualized computing environment.

In this paper, we have surveyed the past and the state of the art in network virtualization research. It is evident that even though network virtualization promises an open, flexible, and heterogeneous networking environment, it will also pose a string of challenges in terms of instantiation, operation, and management that will require coordinated attention from researchers working in networking and other related fields for its success and wide acceptance.

Acknowledgements

We would like to thank the anonymous reviewers for their detailed comments and suggestions throughout the reviewing process that have significantly improved the quality of this paper. We also thank IRTF (Internet Research Task Force) Network Virtualization Research Group (NVRG) members for informative discussions on the definition of network virtualization and network virtualization environment.

References

[1] T. Anderson, L. Peterson, S. Shenker, J. Turner, Overcoming the Internet impasse through virtualization, Computer 38 (4) (2005) 34–41.
[2] J. Turner, D. Taylor, Diversifying the internet, in: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'05), vol. 2, 2005.
[3] N. Feamster, L. Gao, J. Rexford, How to lease the internet in your spare time, SIGCOMM Computer Communication Review 37 (1) (2007) 61–64.
[4] N.M.M.K. Chowdhury, R. Boutaba, Network virtualization: state of the art and research challenges, IEEE Communications Magazine 47 (7) (2009) 20–26.
[5] L.S. Committee, IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks, IEEE Std 802.1Q-2005 (May 2006).
[6] P. Ferguson, G. Huston, What is a VPN?, Tech. Rep., Cisco Systems (1998).
[7] E. Rosen, Y. Rekhter, BGP/MPLS VPNs, RFC 2547 (March 1999).
[8] E. Rosen, Y. Rekhter, BGP/MPLS IP Virtual Private Networks (VPNs), RFC 4364 (February 2006).
[9] L. Andersson, T. Madsen, Provider Provisioned Virtual Private Network (VPN) Terminology, RFC 4026 (March 2005).
[10] M. Carugi, D. McDysan, Service Requirements for Layer 3 Provider Provisioned Virtual Private Networks (PPVPNs), RFC 4031 (April 2005).
[11] R. Callon, M. Suzuki, A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs), RFC 4110 (July 2005).
[12] D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, Generic Routing Encapsulation (GRE), RFC 2784 (March 2000).
[13] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, G. Zorn, Point-to-Point Tunneling Protocol (PPTP), RFC 2637 (July 1999).
[14] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter, Layer Two Tunneling Protocol "L2TP", RFC 2661 (August 1999).
[15] S. Kent, K. Seo, Security Architecture for the Internet Protocol, RFC 4301 (December 2005).
[16] L. Andersson, E. Rosen, Framework for Layer 2 Virtual Private Networks (L2VPNs), RFC 4664 (September 2006).
[17] W. Augustyn, Y. Serbest, Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks, RFC 4665 (September 2006).
[18] E. Mannie, Generalized Multi-Protocol Label Switching (GMPLS) Architecture, RFC 3945 (October 2004).
[19] D. Benhaddou, W. Alanqar, Layer 1 virtual private networks in multidomain next-generation networks, IEEE Communications Magazine 45 (4) (2007) 52–58.
[20] T. Takeda, Framework and Requirements for Layer 1 Virtual Private Networks, RFC 4847 (April 2007).
[21] A.T. Campbell, H.G.D. Meer, M.E. Kounavis, K. Miki, J.B. Vicente, D. Villela, A survey of programmable networks, SIGCOMM Computer Communication Review 29 (2) (1999) 7–23.
[22] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, G.J. Minden, A survey of active network research, IEEE Communications Magazine 35 (1) (1997) 80–86.
[23] D. Wetherall, J. Guttag, D. Tennenhouse, ANTS: a toolkit for building and dynamically deploying network protocols, in: IEEE OPENARCH'98, 1998, pp. 117–129.
[24] D. Decasper, B. Plattner, DAN: distributed code caching for active networks, in: Proceedings of the IEEE INFOCOM'98, vol. 2, 1998, pp. 609–616.
[25] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, J. Zahorjan, Detour: a case for informed internet routing and transport, IEEE Internet Computing 19 (1) (1999) 50–59.
[26] D. Andersen, H. Balakrishnan, F. Kaashoek, R. Morris, Resilient overlay networks, SIGOPS Operating Systems Review 35 (5) (2001) 131–145.
[27] H. Eriksson, MBone: the multicast backbone, Communications of the ACM 37 (8) (1994) 54–60.
[28] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek, J. James, W. O'Toole, Overcast: reliable multicasting with an overlay network, in: Proceedings of the Fourth Conference on Symposium on Operating System Design & Implementation (OSDI'00), 2000, pp. 197–212.
[29] Y. Chu, S. Rao, S. Seshan, H. Zhang, Enabling conferencing applications on the internet using an overlay multicast architecture, SIGCOMM Computer Communication Review 31 (4) (2001) 55–67.
[30] L. Subramanian, I. Stoica, H. Balakrishnan, R. Katz, OverQoS: an overlay based architecture for enhancing internet QoS, in: Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI), 2004, pp. 71–84.
[31] A. Keromytis, V. Misra, D. Rubenstein, SOS: secure overlay services, in: Proceedings of the ACM SIGCOMM Conference (SIGCOMM'02), 2002.
[32] D.G. Andersen, Mayday: distributed filtering for internet services, in: Proceedings of the Fourth Conference on USENIX Symposium on Internet Technologies and Systems (USITS'03), USENIX Association, Berkeley, CA, USA, 2003.
[33] B. Krishnamurthy, C. Wills, Y. Zhang, On the use and performance of content distribution networks, in: Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement (IMW'01), ACM, New York, NY, USA, 2001, pp. 169–182.

[6] http://www.nets-find.net/.
[7] http://www.ict-fireworks.eu/.
[8] http://www.asiafi.net/.
[9] http://www.fif.kr/.

[34] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, A survey and comparison of peer-to-peer overlay network schemes, IEEE Communications Surveys & Tutorials 7 (2) (2005) 72–93.
[35] F. Dabek, M.F. Kaashoek, D. Karger, R. Morris, I. Stoica, Wide-area cooperative storage with CFS, in: Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP'01), ACM, New York, NY, USA, 2001, pp. 202–215.
[36] L. Peterson, T. Anderson, D. Culler, T. Roscoe, A blueprint for introducing disruptive technology into the Internet, SIGCOMM Computer Communication Review 33 (1) (2003) 59–64.
[37] J. Touch, S. Hotz, The X-Bone, in: Proceedings of the Third Global Internet Mini-Conference at GLOBECOM'98, 1998, pp. 44–52.
[38] J.D. Touch, Dynamic internet overlay deployment and management using X-Bone, Computer Networks 36 (2-3) (2001) 117–135.
[39] J.D. Touch, Y.-S. Wang, L. Eggert, G. Finn, A Virtual Internet Architecture, Tech. Rep. TR-570, USC/Information Sciences Institute (2003).
[40] N. Fujita, J.D. Touch, V. Pingali, Y.-S. Wang, A dynamic topology and routing management strategy for virtual IP networks, IEICE Transactions on Communications E89-B (9) (2006) 2375–2384.
[41] J.E. van der Merwe, S. Rooney, I. Leslie, S. Crosby, The Tempest—a practical framework for network programmability, IEEE Network Magazine 12 (3) (1998) 20–28.
[42] J.E. van der Merwe, I.M. Leslie, Switchlets and dynamic virtual ATM networks, in: Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management (IM'97), 1997, pp. 355–368.
[43] R. Boutaba, W. Golab, Y. Iraqi, B. St-Arnaud, Grid-controlled lightpaths for high performance grid applications, Journal of Grid Computing 1 (4) (2003) 387–394.
[44] R. Boutaba, W. Golab, Y. Iraqi, B. St-Arnaud, Lightpaths on demand: a web services-based management system, IEEE Communications Magazine 42 (7) (2004) 2–9.
[45] J. Recio, E. Grasa, S. Figuerola, G. Junyent, Evolution of the user controlled lightpath provisioning system, in: Proceedings of the Seventh International Conference on Transparent Optical Networks, vol. 1, 2005, pp. 263–266.
[46] B. Nandy, D. Bennett, I. Ahmad, S. Majumdar, B. St-Arnaud, User Controlled Lightpath Management System based on a Service Oriented Architecture (2006). <http://www.solananetworks.com/UCLP/files/UCLPv2-SOA.pdf>.
[47] E. Grasa, G. Junyent, S. Figuerola, A. Lopez, M. Savoie, Uclpv2: a network virtualization framework built on web services, IEEE Communications Magazine 46 (6) (2008) 126–134.
[48] A. Sundararaj, P. Dinda, Towards virtual networks for virtual machine grid computing, in: Proceedings of the Third USENIX Virtual Machine Research and Technology Symposium (VM'04), 2004, pp. 177–190.
[49] M. Boucadair, B. Decraene, M. Garcia-Osma, A.J. Elizondo, J.R. Sanchez, B. Lemoine, E. Mykoniati, P. Georgatsos, D. Griffin, J. Spencer, J. Griem, N. Wang, M. Howarth, G. Pavlou, S. Georgoulas, B. Quoitin, Parallel Internets Framework, AGAVE Deliverable (2006) (Id: AGAVE/WP1/FTRD/D1.1/public).
[50] M. Boucadair, P. Levis, D. Griffin, N. Wang, M. Howarth, G. Pavlou, E. Mykoniati, P. Georgatsos, B. Quoitin, J.R. Sanchez, M. Garcia-Osma, A framework for end-to-end service differentiation: network planes and parallel Internets, IEEE Communications 45 (9) (2007) 134–143.
[51] N. Wang, D. Griffin, J. Spencer, J. Griem, J.R. Sanchez, M. Boucadair, E. Mykoniati, B. Quoitin, M. Howarth, G. Pavlou, A.J. Elizondo, M.L.G. Osma, P. Georgatsos, A framework for lightweight QoS provisioning: network planes and parallel Internets, in: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM'07), 2007, pp. 797–800.
[52] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: An Overview, RFC 1633 (Informational) (June 1994).
[53] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475 (December 1998).
[54] D. Grossman, New Terminology and Clarifications for Diffserv, RFC 3260 (Informational) (April 2002).
[55] X. Jiang, D. Xu, VIOLIN: Virtual Internetworking on Overlay Infrastructure, Tech. Rep. TR-03-027, Purdue University (2003).
[56] P. Ruth, X. Jiang, D. Xu, S. Goasguen, Virtual distributed environments in a shared infrastructure, Computer 38 (5) (2005) 63–69.
[57] A. Jun, A. Leon-Garcia, A virtual network approach to network resources management, in: Proceedings of the Canadian Conference on Broadband Research (CCBR'98), 1998.
[58] A. Jun, A. Leon-Garcia, Virtual network resources management: a divide-and-conquer approach for the control of future networks, in: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'98), vol. 2, 1998, pp. 1065–1070.
[59] W. Ng, R. Boutaba, A. Leon-Garcia, Provision and customization of ATM virtual networks for supporting IP services, in: Proceedings of the IEEE ATM Workshop'1999, 1999, pp. 205–210.
[60] W. Ng, D. Jun, H. Chow, R. Boutaba, A. Leon-Garcia, Miblets: a practical approach to virtual network management, in: Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99), 1999, pp. 201–215.
[61] S. da Silva, Y. Yemini, D. Florissi, The NetScript active network system, IEEE Journal on Selected Areas in Communication 19 (3) (2001) 538–551.
[62] S. daSilva, D. Florissi, Y. Yemini, NetScript: A Language-based Approach to Active Networks, Tech. Rep., Columbia University (January 1998).
[63] Y. Yemini, S. daSilva, Towards programmable networks, in: IFIP/IEEE International Symposium on Distributed Systems: Operations and Management, 1997.
[64] M. Kounavis, A. Campbell, S. Chou, F. Modoux, J. Vicente, H. Zhuang, The Genesis Kernel: a programming system for spawning network architectures, IEEE Journal on Selected Areas in Communications 19 (3) (2001) 511–526.
[65] A.A. Lazar, A.T. Campbell, Spawning Networking Architectures (White Paper), Tech. Rep., Columbia University (1998).
[66] A.T. Campbell, M.E. Kounavis, D.A. Villela, J. Vicente, K. Miki, H.G.D. Meer, K.S. Kalaichelvan, Spawning networks, IEEE Network Magazine 13 (4) (1999) 16–30.
[67] P. Szegedi, S. Figuerola, M. Campanella, V. Maglaris, C. Cervell-Pastor, With evolution for revolution: the FEDERICA approach, IEEE Communications Magazine 47 (7) (2009) 34–39.
[68] P. Kauffman, M. Roesler, U. Monaco, A. Sevasti, S. Figuerola, A. Berna, J. Pons, D. Kagoleras, J.-M. Uze, P. Sjödin, M. Hidell, L.D. Cristina Cervelló-Pastor, R. Machado, Evaluation of current network control and management plane for multi-domain network infrastructure, FEDERICA Deliverable (2008) (Id: DJRA1.1).
[69] N. Spring, L. Peterson, A. Bavier, V. Pai, Using PlanetLab for network research: myths, realities, and best practices, SIGOPS Operating Systems Review 40 (1) (2006) 17–24.
[70] G.P. Group, GENIdesign principles, Computer 39 (9) (2006) 102–105.
[71] A. Bavier, N. Feamster, M. Huang, L. Peterson, J. Rexford, In VINI veritas: realistic and controlled network experimentation, in: Proceedings of the SIGCOMM'06, ACM, New York, NY, USA, 2006, pp. 3–14.
[72] M. Handley, E. Kohler, A. Ghosh, O. Hodson, P. Radoslavov, Designing extensible IP router software, in: Proceedings of the Second Conference on Symposium on Networked Systems Design & Implementation (NSDI'05), USENIX Association, Berkeley, CA, USA, 2005, pp. 189–202.
[73] E. Kohler, R. Morris, B. Chen, J. Jannotti, M.F. Kaashoek, The Click modular router, ACM Transactions on Computer Systems 18 (3) (2000) 263–297.
[74] S. Bhatia, M. Motiwala, W. Mühlbauer, Y. Mundada, V. Valancius, A. Bavier, N. Feamster, L. Peterson, J. Rexford, Trellis: a platform for building flexible, fast virtual networks on commodity hardware, in: Proceedings of Workshop on Real Overlays and Distributed Systems (ROADS), 2008.
[75] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, J. Rexford, Virtual routers on the move: live router migration as a network-management primitive, in: Proceedings of the ACM SIGCOMM'08, 2008, pp. 231–242.
[76] E. Keller, R. Lee, J. Rexford, Accountability in hosted virtual networks, in: Proceedings of ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA), 2009.
[77] Y. Zhu, J. Rexford, A. Bavier, N. Feamster, UFO: A Resilient Layered Routing Architecture, Tech. Rep. TR-780-07, Princeton University (2007).
[78] P. Aranda, A.-M. Biraghi, M.-A. Callejo, J.-M. Cabero, J. Carapinha, F. Cardoso, L. Correia, M. Dianati, I.E. Khayat, M. Johnsson, Y. Lemieux, M.P. deLeon, J. Salo, G. Schultz, D. Sebastiao, M. Soellner, Y. Zaki, L. Zhao, M. Zitterbart, D 2.1 Technical Requirements (August 2008) (Id: FP7-ICT-2007-1-216041-4WARD/D2.1).
[79] T.-R. Banniza, A.-M. Biraghi, L. Correia, T. Monath, M. Kind, J. Salo, D. Sebastiao, K. Wuenstel, D 1.1 First Project-wide Assessment on Non-technical Drivers (January 2009) (Id: FP7-ICT-2007-1-216041-4WARD/D-1.1).

[80] N.M.M.K. Chowdhury, Identity Management and Resource Allocation in the Network Virtualization Environment, Master's Thesis, Cheriton School of Computer Science, University of Waterloo (January 2009).
[81] N.M.M.K. Chowdhury, F. Zaheer, R. Boutaba, iMark: an identity management framework for network virtualization environment, in: Proceedings of the 11th IFIP/IEEE International Symposium on Integrated Network Management (IM), 2009.
[82] N.M.M.K. Chowdhury, M.R. Rahman, R. Boutaba, Virtual network embedding with coordinated node and link mapping, in: Proceedings of the 28th Conference on Computer Communications (INFOCOM), 2009.
[83] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, 2005.
[84] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, Above the Clouds: A Berkeley View of Cloud Computing, Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley (February 2009).
[85] J. Mogul, P. Yalagandula, J. Tourrilhes, R. McGeer, S. Banerjee, T. Connors, P. Sharma, Orphal: API Design Challenges for Open Router Platforms on Proprietary Hardware, Tech. Rep. HPL-2008-108, HP Labs (2008).
[86] D. Andersen, Theoretical Approaches to Node Assignment, Unpublished Manuscript (2002). <http://www.cs.cmu.edu/dga/papers/andersen-assign.ps>.
[87] J. Lu, J. Turner, Efficient Mapping of Virtual Networks onto a Shared Substrate, Tech. Rep. WUCSE-2006-35, Washington University (2006).
[88] Y. Zhu, M. Ammar, Algorithms for assigning substrate network resources to virtual network components, in: Proceedings of the IEEE INFOCOM'06, 2006.
[89] W. Szeto, Y. Iraqi, R. Boutaba, A multi-commodity flow based approach to virtual network resource allocation, in: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'03), 2003, pp. 3004–3008.
[90] A. Gupta, J.M. Kleinberg, A. Kumar, R. Rastogi, B. Yener, Provisioning a virtual private network: a network design problem for multicommodity flow, in: ACM Symposium on Theory of Computing, 2001, pp. 389–398.
[91] J. Fan, M. Ammar, Dynamic topology configuration in service overlay networks – a study of reconfiguration policies, in: Proceedings of the IEEE INFOCOM'06, 2006.
[92] M. Yu, Y. Yi, J. Rexford, M. Chiang, Rethinking virtual network embedding: substrate support for path splitting and migration, ACM SIGCOMM Computer Communication Review 38 (2) (2008) 17–29.
[93] I. Houidi, W. Louati, D. Zeghlache, A distributed virtual network mapping algorithm, in: Proceedings of IEEE ICC, 2008, pp. 5634–5640.
[94] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, M. Chiang, Davinci: dynamically adaptive virtual networks for a customized internet, in: ACM CoNEXT, 2008.
[95] A. Greenberg, G. Hjalmtysson, D.A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, H. Zhang, A clean slate 4D approach to network control and management, SIGCOMM Computer Communication Review 35 (5) (2005) 41–54.
[96] P.A. Dinda, T. Gross, R. Karrer, B. Lowekamp, N. Miller, P. Steenkiste, D. Sutherland, The architecture of the remos system, in: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC'01), 2001, p. 252.
[97] B. Raghavan, K. Vishwanath, S. Ramabhadran, K. Yocum, A.C. Snoeren, Cloud control with distributed rate limiting, in: Proceedings of the SIGCOMM'07, 2007, pp. 337–348.
[98] D. McPherson et al., Core network design and vendor prophecies, in: NANOG 25, 2003.
[99] J. Naous, D. Erickson, G.A. Covington, G. Appenzeller, N. McKeown, Implementing an openflow switch on the netfpga platform, in: Proceedings of the Fourth ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS'08), 2008, pp. 1–9.
[100] S. Karlin, L. Peterson, VERA: an extensible router architecture, Computer Networks 38 (3) (2002) 277–293.
[101] L. Mathy, N. Egi, M. Hoerdt, A. Greenhalgh, M. Handley, (Some) implementation issues for virtual routers, in: Workshop on Management of Network Virtualisation, 2007.
[102] M. Dobrescu, N. Egi, K. Argyraki, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, S. Ratnasamy, Routebricks: exploiting parallelism to scale software routers, in: Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP), 2009.
[103] M. Kolon, Intelligent Logical Router Service, Tech. Rep. 200097-001, Juniper Networks (October 2004).
[104] J. Fu, J. Rexford, Efficient IP-address lookup with a shared forwarding table for multiple virtual routers, in: ACM CoNEXT, 2008.
[105] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the art of virtualization, in: Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03), ACM, New York, NY, USA, 2003, pp. 164–177.
[106] M. Feridan, M. Moser, A. Tanner, Building an abstraction layer for management systems integration, in: Proceedings of the First IEEE/IFIP International Workshop on End-to-End Virtualization and Grid Management (EVGM'2007), 2007, pp. 57–60.
[107] A. Milanova, S. Fahmy, D. Musser, B. Yener, A secure programming paradigm for network virtualization (invited paper), in: Proceedings of the Third International Conference on Broadband Communications, Networks, and Systems (BROADNETS'2006), 2006.
[108] A.P. Jayasumana, Q. Han, T.H. Illangasekare, Virtual sensor networks – a resource efficient approach for concurrent applications, in: Proceedings of the International Conference on Information Technology (ITNG'07), IEEE Computer Society, Washington, DC, USA, 2007, pp. 111–115.
[109] G. Smith, A. Chaturvedi, A. Mishra, S. Banerjee, Wireless virtualization on commodity 802.11 hardware, in: Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTECH'07), ACM, New York, NY, USA, 2007, pp. 75–82.
[110] D. Hausheer, B. Stiller, Auctions for virtual network environments, in: Workshop on Management of Network Virtualisation, 2007.
[111] D. Hausheer, B. Stiller, PeerMart: decentralized auctions for bandwidth trading on demand, ERCIM News (68) (2007) 42–43.
[112] A. Feldmann, Internet clean-slate design: what and why?, SIGCOMM Computer Communication Review 37 (3) (2007) 59–64
[113] New Generation Network Architecture: AKARI Conceptual Design (ver1.1) (June 2008). <http://www.akari-project.nict.go.jp/eng/concept-design/AKARI_fulltext_e_translated_version_1_1.pdf>.

N.M. Mosharaf Kabir Chowdhury is a Ph.D. student in Computer Science at the University of California, Berkeley. He received his Master's in Computer Science focusing on network virtualization from the University of Waterloo in 2009. He has published several papers in journals, magazines, conferences, and workshops and served as a reviewer for similar venues. His research interests include clean-slate designs for the future Internet architecture, network virtualization, and data-center networking.

Raouf Boutaba is a Professor of Computer Science, a Cheriton Faculty Fellow, at the University of Waterloo (Canada) and a Visiting Distinguished Professor at POSTECH (Korea). His research interests include network, resource, and service management in wired and wireless networks. He is the founding Editor-in-Chief of the IEEE Transactions on Network and Service Management and is on the editorial boards of other journals. He is currently a distinguished lecturer of the IEEE Communications Society, the chairman of the IEEE Technical Committee on Information Infrastructure, and is serving as the Director of ComSoc Conference Publications. He has received several best paper awards as well as other recognitions such as the Premier's research excellence award and the Don Stokesburry award.