Network Virtualization: State of the Art and Research Challenges

N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba, University of Waterloo

ABSTRACT

Recently network virtualization has been pushed forward by its proponents as a long-term solution to the gradual ossification problem faced by the existing Internet and proposed to be an integral part of the next-generation networking paradigm. By allowing multiple heterogeneous network architectures to cohabit on a shared physical substrate, network virtualization provides flexibility, promotes diversity, and promises security and increased manageability. However, many technical issues stand in the way of its successful realization. This article investigates the past and the state of the art in network virtualization along with the future challenges that must be addressed to realize a viable network virtualization environment.

INTRODUCTION

In recent years, the concept of network virtualization has attracted significant attention in the debate on how to model the next-generation networking paradigm that can replace the existing Internet. Architectural purists view network virtualization as a tool for evaluating new architectures, whereas pluralists conceive virtualization as a fundamental diversifying attribute of the next-generation architecture itself [1]. They believe that network virtualization can eradicate the so-called *ossifying forces* of the current Internet by introducing disruptive technologies [1, 2].

Network virtualization is defined by decoupling the roles of the traditional Internet service providers (ISPs) into two independent entities [2, 3]: infrastructure providers (InPs), who manage the physical infrastructure, and service providers (SPs), who create virtual networks (VNs) by aggregating resources from multiple InPs and offer end-to-end services. Such an environment will proliferate deployment of coexisting heterogeneous network architectures free of the inherent limitations of the existing Internet.

In this article we survey the past and the state of the art of network virtualization, and provide a better understanding of the key research challenges. The rest of this article is organized as follows. First, four somewhat similar ideas (virtual local area networks [VLANs], virtual private networks [VPNs], programmable networks, and overlay networks) are briefly reviewed. Next, a reference business model and a conceptual architecture of a network virtualization environment (NVE) are presented, identifying the characteristics and critical design factors to materialize it. Following this, a number of past and present research projects on network virtualization and related concepts are summarized. Finally, a detailed study of the key issues is presented emphasizing open research challenges with an objective to stoke wide interest among researchers in this field.

HISTORICAL PERSPECTIVE

The concept of multiple coexisting logical networks has appeared in the networking literature several times in the past, and can be categorized into four main classes: VLANs, VPNs, active and programmable networks, and overlay networks.

VIRTUAL LOCAL AREA NETWORK

A VLAN is a group of logically networked hosts with a single broadcast domain regardless of their physical connectivity. All frames in a VLAN bear a VLAN ID in the medium access control (MAC) header, and VLAN-enabled switches use both the destination MAC address and VLAN ID to forward frames. Since VLANs are based on logical instead of physical connections, network administration, management, and reconfiguration of VLANs are simpler than in their physical counterparts. In addition, VLANs provide elevated levels of isolation.

VIRTUAL PRIVATE NETWORK

A VPN is a dedicated network connecting multiple sites using private and secured tunnels over shared or public communication networks like the Internet. In most cases, VPNs connect geographically distributed sites of a single corporate enterprise. Each VPN site contains one or more customer edge (CE) devices that are attached to one or more provider edge (PE) routers.

Based on the protocols used in the data plane, VPNs can be classified into the following broad categories.

Layer 1 VPN — The layer 1 VPN (L1VPN) framework emerged in recent years from the need to extend layer 2/3 (L2/L3) packet switching VPN concepts to advanced circuit switching

0163-6804/09/$25.00 © 2009 IEEE

domains. It provides a multiservice backbone where customers can offer their own services, whose payloads can be of any layer (e.g., asynchronous transfer mode [ATM] and IP). This ensures that each service network has an independent address space, an independent L1 resource view, separate policies, and complete isolation from other VPNs.

Layer 2 VPN — Layer 2 VPNs (L2VPNs) transport L2 (typically Ethernet) frames between participating sites. The advantage is that they are agnostic about the higher-level protocols, and consequently more flexible than L3VPN. On the downside, there is no control plane to manage reachability across the VPN.

Layer 3 VPN — A layer 3 VPN (L3VPN) is characterized by its use of L3 protocols in the VPN backbone to carry data between the distributed CEs. There are two types of L3VPNs.

In the *CE-based* VPN approach, the provider network is completely unaware of the existence of a VPN. CE devices create, manage, and tear down the tunnels between themselves. Sender CE devices encapsulate the passenger packets and route them into carrier networks; when these encapsulated packets reach the end of the tunnel (i.e., receiver CE devices), they are extracted, and actual packets are injected into receiver networks.

In the *PE-based* approach, the provider network is responsible for VPN configuration and management. A connected CE device may behave as if it were connected to a private network.

Higher-Layer VPNs — VPNs using higher-layer (e.g., transport, session, or application) protocols also exist. SSL/TLS-based VPNs are popular for their inherent advantages in firewall and NAT traversals from remote locations. Such VPNs are lightweight, easy to install and use, and provide higher granularity of control to their users.

Active and Programmable Networks

Active and programmable networks research was motivated by the need to create, deploy, and manage novel services on the fly in response to user demands. In addition to programmability, they also promote concepts of *isolated environments* to allow multiple parties to run possibly conflicting codes on the same network elements without causing network instability.

Two separate schools of thought emerged on how to actually implement such concepts.

The Open Signaling Approach — Open signaling takes a telecommunication approach with a clear distinction between transport, control, and management planes that constitute programmable networks, and emphasizes quality of service (QoS) guarantees. An abstraction layer is proposed for physical network devices to act as distributed computing environments with well defined open programming interfaces allowing service providers to manipulate network states.

The Active Networks Approach — Active networks promote dynamic deployment of new services at runtime within the confinement of existing networks. Routers or switches in these networks can perform customized computations based on the contents of the *active* packets and can also modify them. Active networks allow the customization of network services at packet transport granularity and offer more flexibility than the open signaling approach at the expense of a more complex programming model.

Overlay Networks

An overlay network is a logical network built on top of one or more existing physical networks. The Internet itself started off as an overlay on top of the telecommunication network. Overlays in the existing Internet are typically implemented in the application layer; however, various implementations at lower layers of the network stack do exist.

Overlays do not require or cause any changes to the underlying network. Consequently, overlays have long been used as relatively easy and inexpensive means to deploy new features and fixes in the Internet. A multitude of application layer overlay designs have been proposed in recent years to address diverse issues, which include ensuring performance and availability of Internet routing, enabling multicasting, providing QoS guarantees, protecting from denial of service attacks, and content distribution and file sharing services. Overlays have also been used as testbeds (e.g., PlanetLab) to design and evaluate new architectures.

The authors in [1] point out that standard overlays falter as a deployment path for radical architectural innovations in at least two ways. First, overlays have largely been used as a means to deploy narrow fixes to specific problems without any holistic view. Second, most overlays have been designed in the application layer on top of IP; hence, they cannot go beyond the inherent limitations of the existing Internet.

Network Virtualization Environment

Unlike the existing all-IP Internet, a virtualized networking environment is a collection of multiple heterogeneous network architectures from different SPs. Each SP leases resources from one or more InPs to create VNs, and deploys customized protocols and services.

Business Model

The main distinction between the participants in the network virtualization model and the traditional model is the presence of two different roles, InPs and SPs, as opposed to the single role of the ISPs [2–4].

InP — InPs deploy and actually manage the underlying physical network resources. They offer their resources through programmable interfaces to different SPs. InPs distinguish themselves through the quality of resources they provide, the freedom they delegate to their customers, and the tools they provide to exploit that freedom.

Unlike the existing all-IP Internet, a virtualized networking environment is a collection of multiple heterogeneous network architectures from different SPs. Each SP leases resources from one or more InPs to create VNs and deploys customized protocols and services.

End users in the network virtualization model are similar to those of the existing Internet, except that the existence of multiple VNs from competing SPs provides them a wider range of choice. Any end user can connect to multiple VNs from different SPs for different services.

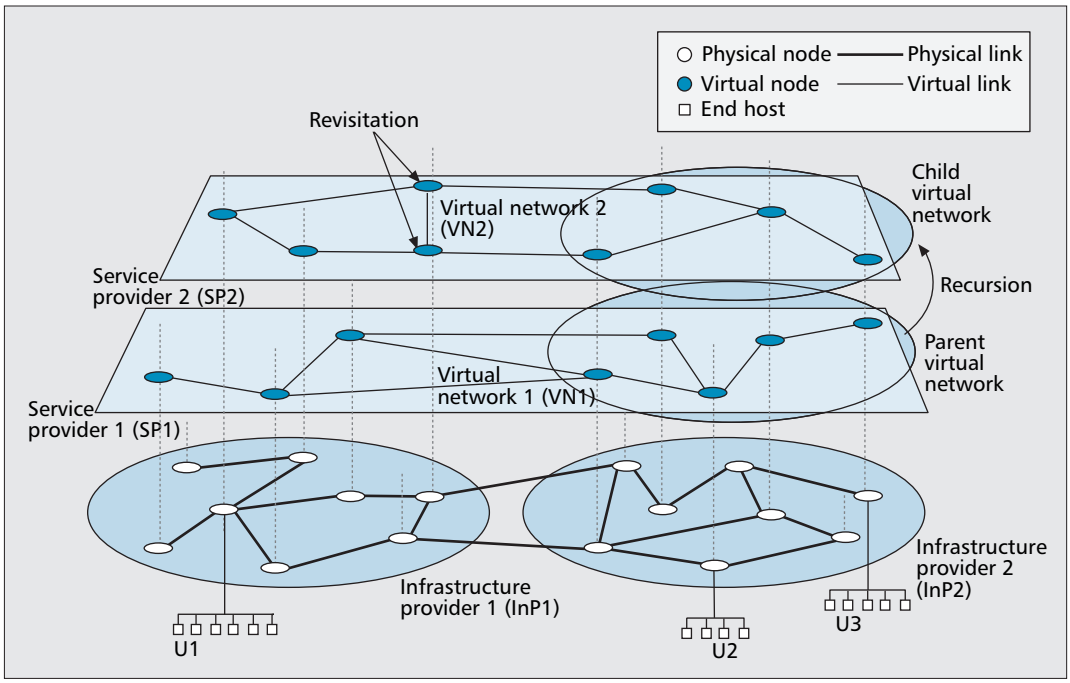

Figure 1. *Network virtualization environment.*

SP — SPs lease resources from multiple InPs to create and deploy VNs by programming allocated network resources to offer end-to-end services to end users. An SP can also provide network services to other SPs. It can also create child VNs by partitioning its resources and act as a virtual InP by leasing those child networks to other SPs (Fig. 1).

End User — End users in the network virtualization model are similar to those of the existing Internet, except that the existence of multiple VNs from competing SPs provides them a wider range of choice. Any end user can connect to multiple VNs from different SPs for different services.

Architecture

In an NVE the basic entity is a VN. A VN is a collection of virtual nodes connected together by a set of virtual links to form a virtual topology, which is essentially a subset of the underlying physical topology. Each virtual node is hosted on a particular physical node, whereas a virtual link spans over a path in the physical network and includes a portion of the network resources along the path.

Each VN is operated and managed by a single SP, even though the underlying physical resources might be aggregated from multiple InPs. Figure 1 depicts two VNs, VN1 and VN2, created by service providers SP1 and SP2, respectively. SP1 composed VN1 on top of the physical resources managed by two different InPs (InP1 and InP2), and provides end-to-end services to end users U2 and U3. SP2, on the other hand, deployed VN2 by combining resources from infrastructure provider InP1 with a child VN from service provider SP1. End users U1 and U3 are connected through VN2.

The owner of a VN is free to implement end-to-end services by deploying custom packet formats, routing protocols, forwarding mechanisms, as well as control and management planes. As mentioned earlier, end users have the choice to opt in to any VN. For example, U3 is subscribed to VN1 and VN2 managed by SP1 and SP2, respectively.

Architectural Principles

Network virtualization propounds the following principles for the next-generation networking paradigm.

Coexistence — Coexistence of multiple VNs is the defining characteristic of an NVE [1–3]. It refers to the fact that multiple VNs from different SPs can coexist together, spanning over part or full of the underlying physical networks provided by one or more InPs. In Fig. 1, VN1 and VN2 are two coexisting VNs.

Recursion — When one or more VNs are spawned from another VN creating a VN hierarchy with *parent-child* relationships, it is known as recursion as well as *nesting* of VNs [5]. Service provider SP1 in Fig. 1 leased away a portion of its allocated resources to SP2, to whom it appears simply as a virtual InP.

Inheritance — Child VNs in an NVE can inherit architectural attributes from their parents, which also means that the constraints on the parent VN automatically translate to similar constraints on its children [5]. For example, constraints imposed by InP2 will automatically be transferred to VN2 from VN1 through inheritance. Inheritance allows an SP to add value to the spawned child VNs before reselling them to other SPs [3].

Revisitation — Revisitation [6] allows a physical node to host multiple virtual nodes of a single VN. Use of multiple logical routers to handle

diverse functionalities in a large complex network allows an SP to logically rearrange its network structure and to simplify the management of a VN. Revisitation can also be useful for creating testbed networks. Figure 1 provides an example of revisitation in VN2.

Design Goals

The design goals for successfully realizing network virtualization have been addressed by different research groups. In order to materialize network virtualization, each of these design criteria should be fulfilled.

Flexibility — Network virtualization must provide freedom in every aspect of networking. Each SP should be free to implement arbitrary network topology, routing and forwarding functionalities, and customized control protocols independent of the underlying physical network and other coexisting VNs. For example, deploying source routing in today's network depends much on consensus among ISPs; in a virtualized environment, the owner of a VN should be able to offer source routing without having to coordinate with any other parties.

Manageability — By separating SPs from InPs, network virtualization will modularize network management tasks and introduce accountability at every layer of networking [3]. It must provide complete end-to-end control of the VNs to the SPs, obviating the requirement of coordination across administrative boundaries seen in the existing Internet.

Scalability — Coexistence of multiple networks is one of the fundamental principles of network virtualization. Scalability is an indispensable part of this equation. InPs in an NVE must scale to support an increasing number of coexisting VNs without affecting their performance.

Isolation — Network virtualization must ensure isolation between coexisting VNs to improve fault tolerance, security, and privacy. Network protocols are prone to misconfigurations and implementation errors. Virtualization must ensure that misconfigurations in one VN are contained within itself and do not affect other coexisting VNs.

Stability and Convergence — Isolation ensures that faults in one VN do not affect other coexisting VNs, but errors and misconfigurations in the underlying physical network can also destabilize an NVE. Moreover, instability in the InPs (e.g., routing oscillation) can lead to instability of all hosted VNs. Virtualization must ensure the stability of an NVE, and in case of any instability the affected VNs must be able to successfully converge to their stable states.

Programmability — To ensure flexibility and manageability, programmability of the network elements is an indispensable requirement. Only through programmability can SPs implement customized protocols and deploy diverse services. Two pressing questions in this respect must have satisfactory answers: "How much programmability should be allowed?" and "How should it be exposed?" A win-win situation must be found where programmability is easy, effective, and secure at the same time.

Heterogeneity — Heterogeneity in the context of network virtualization comes mainly from two fronts: first, heterogeneity of the underlying networking technologies (e.g., optical, wireless, and sensor); second, each end-to-end VN, created on top of that heterogeneous combination of underlying networks, can also be heterogeneous. SPs must be allowed to compose and run cross-domain end-to-end VNs without the need for any technology-specific solutions. Underlying infrastructures must also be capable of supporting heterogeneous protocols and algorithms implemented by different SPs. In addition, heterogeneity of end-user devices must also be taken into account.

Legacy Support — Legacy support or backward compatibility has always been a matter of deep concern while deploying any new technology. Conceptually, network virtualization can easily integrate legacy support by considering the existing Internet as just another VN in its collection of networks; but whether and how it can be done efficiently remains an open challenge.

Network virtualization must provide freedom at every aspect of networking. Each SP should be free to implement arbitrary network topology, routing and forwarding functionalities, and customized control protocols independent of the underlying physical network and other coexisting VNs.

Network Virtualization Projects

Over the years, the term *virtual network* has been used to describe different projects on VPNs, overlay networks, and active or programmable networks. But very few of them actually followed the pluralist view of network virtualization. In Table 1 we summarize the most significant past and on-going projects directly or indirectly related to network virtualization based on the following set of characteristics:

- ***Networking technology***: A handful of network virtualization prototypes have been developed for specific networking technologies with an aim to exploit unique characteristics of those networks to enable virtualization. Such projects include X-Bone for IP networks, Tempest targeting ATM networks, and the very recent GENI initiative that will be agnostic to any specific technology.
- ***Layer of virtualization***: Influenced by the existing Internet, researchers have naturally approached network virtualization in a layered manner. As a result, many projects have attempted to virtualize different layers of the network stack, starting from the physical layer (UCLP) and continuing up to the application layer (VIOLIN).
- ***Architectural domain***: Most projects have focused on particular architectural domains, which dictate the design choices made in the construction of architectures and services that can be offered on those platforms. Examples include network management (VNRMS), virtual active networks (NetScript), and spawning networks (Genesis).
- ***Level of virtualization***: To enable network virtualization, one must virtualize the nodes, links, and every other resource in the network. The level of virtualization refers to

Project	Architectural Domain	Networking Technology	Layer of Virtualization	Level of Virtualization
VNRMS [7]	Virtual network management	ATM/IP		Node/Link
Tempest [8]	Enabling alternate control architectures	ATM	Link	
NetScript [9]	Dynamic composition of services	IP	Network	Node
Genesis [5]	Spawning virtual network architectures		Network	Node/Link
VNET [10]	Virtual machine Grid computing		Link	Node
VIOLIN [11]	Deploying on-demand value-added services on IP overlays	IP	Application	Node
X-Bone [6]	Automating deployment of IP overlays	IP	Network	Node/Link
PlanetLab [12]	Deployment and management of overlay-based testbeds	IP	Application	Node
UCLP	Dynamic provisioning and reconfiguration of lightpaths	SONET	Physical	Link
AGAVE [4]	End-to-end QoS-aware service provisioning	IP	Network	
GENI	Creating customized virtual network testbeds	Heterogeneous		
VINI [13]	Evaluating protocols and services in a realistic environment		Link	
CABO [3]	Deploying value-added end-to-end services on shared infrastructure	Heterogeneous		Full

Table 1. *Characteristics of different network virtualization projects over the years.*

the granularity at which each VN can administer itself. At one end of this spectrum, node virtualization creates VNs by connecting virtual nodes on different physical nodes (e.g., PlanetLab). At the other end, CABO proposes the concept of true plurality where each VN has a semblance of the native network.

It can be noticed from Table 1 that over time, research on network virtualization has shifted focus toward creating a holistic and generalized NVE that features a completely virtualized (virtualization of all network elements), highly customizable (virtualization at lower layers), and technology-agnostic (creation of VNs over a heterogeneous combination of underlying networks) networking facility for the future Internet.

Research Challenges

Most of the existing research work related to network virtualization can at best be described as attempts to fix existing problems, rather than a conscious and focused push to build a complete NVE. As a result, several aspects of network virtualization remain unexplored, and many others require modification and improvement. In this section we summarize the key issues to be resolved for the realization of an NVE.

Interfacing

Every InP must provide an interface, following some standard, so that SPs can communicate with them and express their requirements. In addition, standard interfaces are also required to make programmability of network elements available to the SPs. On a similar note, appropriate interfaces between end users and SPs, as well as among multiple InPs and among SPs must also be identified and standardized.

Signaling and Bootstrapping

Before creating a VN, an SP must already have network connectivity to the InPs in order to issue its requests. This introduces circularity where network connectivity is a prerequisite to itself [3]. There must also be bootstrapping capabilities to allow SPs to customize the virtual nodes and virtual links allocated to them through appropriate interfaces. Both requirements call for at least another network that will always be present to provide connectivity to handle these issues, or an *out-of-band* mechanism to perform signaling and bootstrapping.

Resource and Topology Discovery

In order to allocate resources for requests from different SPs, InPs must be able to determine the topology of the networks they manage as well as the status of the corresponding network elements (physical nodes and interconnections between them, remaining capacities in nodes and links, etc.). Furthermore, two adjacent InPs must also be able to instantiate cross-domain virtual links to enable end-to-end VNs.

From an SP's point of view, a VN should be able to discover the presence and topologies of other coexisting VNs. This will allow VNs to communicate, interact, and collaborate between themselves to provide larger complex services.

Resource Allocation

Efficient allocation and scheduling of physical resources among multiple VN requests is extremely important in order to maximize the number of coexisting VNs, and increase the utilization and revenue of the InPs. The allocation of resources with constraints on virtual nodes and virtual links, also known as the VN embedding problem, can be represented using a mixed-integer program (MIP) [14]. Solving an MIP is known to be $\mathcal{NP}$-hard; so is finding optimal VN embedding.

Existing heuristic-based solutions focus on two major versions of the problem in the single-InP scenario: *offline*, where all the SPs' requests are known in advance, and *online*, the opposite. Even though various constraints and objectives make this problem computationally intractable, the presence of multifarious topologies and possible opportunities to exploit them still leave enough room for research on customized solutions and better approximation algorithms. In addition, VN embedding across multiple InPs is still a virtually untouched problem.

Admission Control and Usage Policing

When establishing a VN, an SP may require specific guarantees for its VNs' attributes as well as its virtual links' characteristics. InPs must perform accurate accounting, and implement admission control and distributed usage policing algorithms to ensure that they can deliver the guaranteed performance, and the hosted VNs do not exceed allocated resources either locally or globally. However, algorithms must be developed for complete VNs, instead of existing admission control or policing algorithms for individual nodes or links.

Virtual Nodes and Virtual Links

Virtual nodes allow multiple SPs to share the same set of physical resources and implement separate customized control protocols on them. Until now, router vendors have promoted virtual nodes as a tool for simplifying core network design, decreasing capital expenditure (CAPEX), and VPN purposes. A similar concept can be extended with programmability to create substrate routers that allow each SP to customize their virtual nodes. Scalability of an NVE is closely tied to scalability of the physical elements used by the InPs. Research in this direction should focus on increasing the number of virtual nodes any single physical router can hold.

To realize network virtualization, links between virtual nodes must also be virtualized. The ability to create tunnels over multiple physical links already exists in the context of VPNs. Similar tunneling mechanisms can also be used in VNs. The speed of transporting packets across a virtual link should be comparable to that of a native link, which translates into minimum encapsulation and multiplexing cost.

Naming and Addressing

Mapping between different address contexts is a well-known problem in the existing literature. But in the presence of different, often incompatible, addressing requirements in different VNs, it becomes even more complicated [15].

Naming and addressing should be decoupled in an NVE so that any end user can move from one SP to another with a single identity. Even though the concept of being simultaneously connected to multiple VNs from different SPs sounds similar to *multihoming*, the problem is exacerbated by the possible heterogeneity among different VNs [15].

Mobility Management

In an NVE, mobility of the devices must be supported congenitally, not using makeshift solutions as in the existing Internet. Mobility in this context does not just refer to its simplest form (i.e., geographic mobility of end user devices); routers in the core network can also move around using migration techniques. As a result, finding the exact location of any device at a particular moment and routing packets accordingly is a complex issue that needs simple solutions. In addition, end users can also move logically from one VN to another in order to access different services, which further complicates the problem.

Monitoring, Configuration, and Failure Handling

To enable individual SPs to configure, monitor, and control their VNs irrespective of others, considerable changes are required from the level of network operations centers (NOCs) to intelligent agents at lower-level network elements. The concept of MIBlets [7] (partitioned management information bases [MIBs]) to gather and process performance statistics for each of the coexisting VNs instead of using a common MIB can be a good starting point. But a full-fledged robust monitoring framework needs more attention and effort.

Failures in the underlying physical network components can give rise to cascading series of failures in all the VNs directly hosted on those components. Detection, propagation, and isolation of such failures, as well as protection and restoration from them are all open research challenges.

Security and Privacy

Isolation between coexisting VNs can only provide a certain level of security and privacy through the use of secured tunnels, encryptions, and so on; but it does not obviate the prevalent threats, intrusions, and attacks to the physical layer and VNs. In addition to that, security and privacy issues specific to network virtualization must also be identified and explored. For example, programmability of the network elements can increase vulnerability if secure programming models and interfaces are unavailable. All these issues require close examination to create a realistic NVE.

Interoperability Issues

End-to-end VNs can span across multiple administrative domains, each using possibly heterogeneous networking technologies and management frameworks. Enabling virtualization in each of these technologies requires specific solutions for provisioning, operation, and maintenance. Interactions between such contrasting underlying infrastructures, while providing a generic and transparent management interface for SPs to

Efficient allocation and scheduling of physical resources among multiple VN requests is extremely important in order to maximize the number of coexisting VNs and to increase the utilization and revenue of the InPs.

Project	Originated	Link
4WARD	Europe	http://www.4ward-project.eu/
AKARI	Japan	http://akari-project.nict.go.jp/
CABO	United States	http://www.cs.princeton.edu/~jrex/virtual.html
Clean Slate	United States	http://cleanslate.stanford.edu/
GENI	United States	http://www.geni.net/
NouVeau	Canada	http://netlab.cs.uwaterloo.ca/virtual/
PlanetLab	United States	http://www.planet-lab.org/
Trilogy	Europe	http://www.trilogy-project.org/
UCLP	Canada	http://www.uclp.ca/
VINI	United States	http://www.vini-veritas.net/
X-Bone	United States	http://www.isi.edu/xbone/

Table 2. *Recent network virtualization related projects.*

easily compose and manage VNs, remains a daunting task. In addition, identification of the necessity, scope, and required interfaces for end-to-end communication across multiple VNs deserves close scrutiny.

Network Virtualization Economics

Unlike traditional networks where bandwidth is the chief commodity, virtual nodes are equally important as virtual links in an NVE. SPs are the buyers in this economy, whereas InPs are the sellers. There can also be brokers who act as mediators between the buyers and sellers. End users also participate as buyers of services from different SPs.

Traditionally, there are two general types of marketplaces: centralized and decentralized. Centralized marketplaces are efficient, but vulnerable and not scalable. On the other hand, fully decentralized marketplaces are extensible and fault-tolerant, but prone to malicious behavior and inefficiency. To find a trade-off between these two options, existing work on peer-to-peer (P2P) marketplaces can be extended to the domain of network virtualization.

Conclusion

Amid current trends of virtualizing practically every aspect of computing (e.g., operating systems, servers, and data centers), network virtualization stands at a unique point in the virtualization design space. On one hand, it is necessary to have a virtualized network to interconnect all other virtualized appliances to give each of the virtual entities a complete semblance of their native counterparts.

On the other hand, after enjoying years of rapid growth, the progress of the Internet and networking in general has come to a standstill. Most researchers now agree that a redesign is a bare necessity, not luxury. Network virtualization can take a leading role in this scenario to promote innovation through disruptive technologies. This realization has given birth to several projects all over the world that are directly or indirectly related to network virtualization (Table 2).

The materialization of an NVE needs to satisfy the requirements set by its characteristics and design goals, but fulfilling these requirements is not so easy. More insights into the challenges outlined in this article are needed for an open, flexible, and heterogeneous networking environment to become a reality.

References

[1] T. Anderson *et al.*, "Overcoming the Internet Impasse through Virtualization," *Computer*, vol. 38, no. 4, 2005, pp. 34–41.
[2] J. Turner and D. Taylor, "Diversifying the Internet," *Proc. GLOBECOM '05*, vol. 2, 2005.
[3] N. Feamster, L. Gao, and J. Rexford, "How to Lease the Internet in your Spare Time," *SIGCOMM Comp. Commun. Revi.*, vol. 37, no. 1, 2007, pp. 61–64.
[4] M. Boucadair *et al.*, "A Framework for End-to-End Service Differentiation: Network Planes and Parallel Internets," *IEEE Commun. Mag.*, vol. 45, no. 9, Sept. 2007, pp. 134–43.
[5] M. Kounavis *et al.*, "The Genesis Kernel: A Programming System for Spawning Network Architectures," *IEEE JSAC*, vol. 19, no. 3, 2001, pp. 511–26.
[6] J. Touch, "Dynamic Internet Overlay Deployment and Management using the X-Bone," *Comp. Networks*, vol. 36, no. 2–3, 2001, pp. 117–35.
[7] W. Ng *et al.*, "MIBlets: A Practical Approach to Virtual Network Management," *Proc. 6th IFIP/IEEE Int'l. Symp. Integrated Net. Mgmt.*, 1999, pp. 201–15.
[8] J. E. van der Merwe *et al.*, "The Tempest: A Practical Framework for Network Programmability," *IEEE Network*, vol. 12, no. 3, 1998, pp. 20–28.
[9] S. da Silva, Y. Yemini, and D. Florissi, "The NetScript Active Network System," *IEEE JSAC*, vol. 19, no. 3, 2001, pp. 538–51.
[10] A. Sundararaj and P. Dinda, "Towards Virtual Networks for Virtual Machine Grid Computing," *Proc. 3rd USENIX Virtual Machine Research Tech. Symp.*, 2004.
[11] P. Ruth *et al.*, "Virtual Distributed Environments in a Shared Infrastructure," *Computer*, vol. 38, no. 5, 2005, pp. 63–69.
[12] L. Peterson *et al.*, "A Blueprint for Introducing Disruptive Technology into the Internet," *SIGCOMM Comp. Commun. Rev.*, vol. 33, no. 1, 2003, pp. 59–64.
[13] A. Bavier *et al.*, "In VINI veritas: Realistic and Controlled Network Experimentation," *Proc. ACM SIGCOMM*, 2006, pp. 3–14.
[14] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual Network Embedding with Coordinated Node and Link Mapping," *IEEE INFOCOM*, 2009.
[15] N. M. M. K. Chowdhury, F. Zaheer, and R. Boutaba, "iMark: An Identity Management Framework for Network Virtualization Environment," *IFIP/IEEE Int'l. Symp. Integrated Net. Mgmt.*, 2009.

Biographies

N. M. Mosharaf Kabir Chowdhury (mosharafkabir@gmail.com) completed his Master's in computer science from the University of Waterloo, Canada, in 2009 and his Bachelor's in computer science and engineering from Bangladesh University of Engineering and Technology in 2006. His research interests include network virtualization, data center networking, and next-generation Internet architectures.

Raouf Boutaba (rboutaba@uwaterloo.ca) is a professor of computer science at the University of Waterloo. His research interests include network and service management. He is the founder and Editor-in-Chief of the *IEEE Transactions on Network and Service Management*. He is a distinguished lecturer of IEEE Communications Society, Chairman of the Technical Committee on Information Infrastructure, and Director of Conference Publications. He has received several best paper awards and other recognitions including the Premier's Research Excellence Award.